MEASUREMENT AND MODELING OF LARGE SCALE NETWORKS

By

Muhammad Zubair Shafiq

A DISSERTATION

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

Computer Science and Engineering—Doctor of Philosophy

2014

ABSTRACT

MEASUREMENT AND MODELING OF LARGE SCALE NETWORKS

By

Muhammad Zubair Shafiq

The goal of this thesis is to identify measurement, modeling, and optimization opportunities for large scale networks – with specific focus on cellular networks and online social networks. These networks are facing unprecedented operational challenges due to their very large scale.

Cellular networks are experiencing an explosive increase in the volume of traffic for the last few years. This unprecedented increase in the volume of mobile traffic is attributed to the increase in the subscriber base, improving network connection speeds, and improving hardware and software capabilities of modern smartphones. In contrast to the traditional fixed IP networks, mobile network operators are faced with the constraint of limited radio frequency spectrum at their disposal. As the communication technologies evolve beyond 3G to Long Term Evolution (LTE), the competition for the limited radio frequency spectrum is becoming even more intense. Therefore, mobile network operators increasingly focus on optimizing different aspects of the network by customized design and management to improve key performance indicators (KPIs).

Online social networks are increasing at a very rapid pace, while trying to provide more content-rich and interactive services to their users. For instance, Facebook currently has more than 1.2 billion monthly active users and offers news feed, graph search, groups, photo sharing, and messaging services. The information for such a large user base cannot be efficiently and securely managed by traditional database systems. Social network service providers are deploying novel large scale infrastructure to cope with these scaling challenges.

In this thesis, I present novel approaches to tackle these challenges by revisiting the cur-

rent practices for the design, deployment, and management of large scale network systems using a combination of theoretical and empirical methods. I take a data-driven approach in which the theoretical and empirical analyses are intertwined. First, I measure and analyze the trends in data and then model the identified trends using suitable parametric models. Finally, I rigorously evaluate the developed models and the resulting system design prototypes using extensive simulations, realistic testbed environments, or real-world deployment. This methodology is to used to address several problems related to cellular networks and online social networks.

ACKNOWLEDGEMENTS

Working towards a Ph.D. has been a deeply enriching experience; at times it has been exciting, at times depressing (particularly after crushing paper rejections), but it has always been very rewarding. 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. I do not know where to start. My work would not have been possible without his constant guidance – like when he nudged me into considering measurement research more seriously – 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. Eric Torng, Guoliang Xing, and Habib Salehi for their encouragement and insightful comments during my qualifier and comprehensive exams.

- I would also like to thank Drs. Jeffrey Pang, Jia Wang, and Lusheng Ji. I learned a lot from them during my summer internship at AT&T Labs - Research and during our collaboration throughout my Ph.D. My collaboration with them has been one of the most fruitful and fun engagements I have experienced. I have learned a lot writing papers with them.

- I would also like to thank Drs. Franck Le and Mudhakar Srivatsa. I really enjoyed working with them during my summer internship at IBM T.J. Watson Research Center and during our collaboration after the internship.

- 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 and WAVES lab at Michigan State University. In particular, I would like to thank Ahmed Majeed Khan, Amir Khakpour, Chad Meiners, Eric Norige, Faraz Ahmed, Fei Chen, Hassan Aqeel Khan, Muhammad Shahzad, and Muhammad Usman Ilyas for numerous insightful discussions and collaborations on various projects.

- 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. Syed Ali Khayam and Muddassar Farooq, who advised my undergraduate thesis and encouraged me to pursue Ph.D.

- I am also deeply indebted to Mr. Saleem, my high school physics teacher in Bahawalpur, Pakistan. His passion for science and scholarly pursuit has been an inspiration to me (and surely, many of his other students) and helped set me on the path on which I find myself today.

- Finally, I do not know how I can thank my family enough: my wife and parents, from whom I realized that kindness and devotion is endless, my brother, Omair, who always supports me no matter what, and the rest of the family members who were always supportive of my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Background

Big data has become the cornerstone of the revolution in computing systems that is transforming how network infrastructure, services, and applications are designed, deployed, and managed. Many conventional wisdoms on which we have based the designs of existing network systems are either not well understood or they are simply outdated due to the rapidly changing nature of workloads, performance metrics, and user requirements. Therefore, existing systems face unprecedented challenges in terms of efficiency, performance, reliability, adaptability, and scalability due to their inferior and outdated designs.

This thesis aims to tackle these challenges by revisiting the current practices for the design, deployment, and management of network systems using a combination of theoretical and empirical methods. To model dynamic workload behaviors, my work takes a data-driven approach in which the theoretical and empirical analyses are intertwined. First, I measure and analyze the trends in data and then model the identified trends using suitable parametric models. Finally, I rigorously evaluate the developed models and the resulting system design prototypes using extensive simulations, realistic testbed environments, or real-world deployment. I have successfully applied this methodology to address several problems related to mobile and cellular networks and online social networks – especially for the issues that arise from dynamic workload behaviors.

1.2 Contributions

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

1.2.1 Quality-of-Experience for Mobile Video

Mobile network operators have a significant interest in the performance of streaming video on their networks because network factors directly influence the quality-of-experience. However, unlike video service providers, network operators are not privy to the client- or server-side logs typically used to measure user engagement. To address this limitation, I analyze the impact of cellular network performance on mobile video abandonment (a measure of user engagement) from the perspective of a network operator. I develop predictive models to enable mobile network operators to monitor mobile video user engagement using only standard radio network statistics and/or TCP/IP flow records, a necessity for continuous monitoring at scale [88].

1.2.2 Mobile Network Performance during Crowded Events

Cellular network usage is at an all-time high even under normal operating conditions and projections show that traffic volume will further increase 26x by 2015 as compared to 2010. During crowded events, cellular networks face voice and data traffic volumes that are often orders of magnitude higher than what they face during routine days. I conduct a large scale investigation of cellular network performance during crowded events to identify the root cause of performance degradation. I design and evaluate practical mitigation schemes, such as radio resource allocation tuning and opportunistic connection sharing, which do not require making significant changes to the current cellular network infrastructure [90].

1.2.3 Breaching Privacy in Encrypted IM Networks

The proliferation of online social networks has attracted the interest of computer scientists to mine the available social network data for developing behavior profiles of people. IM services – such as Yahoo! Messenger, Skype, IRC, and ICQ – are popular tools to privately communicate with friends and family over the Internet. I demonstrate how to de-anonymizing friendship links in encrypted instant messaging (IM) networks using timing analysis on the collected network traffic logs [48]. This work is also relevant to related problems such as mix network de-anonymization.

1.2.4 Information Hub Identification in Social Networks

Identifying top-k information hubs is crucial for many applications such as advertising in social networks where advertisers are interested in identifying hubs to whom free samples can be given. Centralized computation of top-k information hubs is mostly unrealistic for parties such as advertisers because online social networking companies are reluctant to share their interaction or friendship graphs due to privacy concerns and regulations. I develop distributed and privacy-preserving algorithms for computing top-k information hubs in online social networks [46, 47].

1.3 Published Material

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

- M. Zubair Shafiq, Jeffrey Erman, Lusheng Ji, Alex X. Liu, Jeffrey Pang, Jia Wang. Understanding the Impact of Network Dynamics on Mobile Video User Engagement. ACM SIGMETRICS, 2014.
- M. Zubair Shafiq, Alex X. Liu, Amir Khakpour. Revisiting Caching in Content Delivery Networks: Measurement, Design, and Evaluation (Extended Abstract). ACM

SIGMETRICS, 2014.

- M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, Shobha Venkataraman, Jia Wang. A First Look at Cellular Network Performance during Crowded Events. ACM SIGMETRICS, 2013.

- M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, Jia Wang. A First Look at Cellular Machine-to-Machine Traffic - Large Scale Measurement and Characterization. ACM SIGMETRICS, 2012.

- M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jia Wang. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. ACM SIGMETRICS, 2011.

- Muhammad U. Ilyas, M. Zubair Shafiq, Alex X. Liu, Hayder Radha. Who are You Talking to? Breaching Privacy in Encrypted IM Networks. IEEE ICNP, 2013.

- Yipeng Wang, Xiaochun Yun, M. Zubair Shafiq, Liyan Wang, Alex X. Liu, Zhibin Zhang, Danfeng(Daphne) Yao, Yongzheng Zhang, Li Guo. A Semantics Aware Approach to Automated Reverse Engineering Unknown Protocols. IEEE ICNP, 2012.

- M. Zubair Shafiq, Muhammad U. Ilyas, Alex X. Liu, Hayder Radha. Identifying Leaders and Followers in Online Social Networks. IEEE Journal on Selected Areas in Communications (JSAC), 2013.

- Muhammad U. Ilyas, M. Zubair Shafiq, Alex X. Liu, Hayder Radha. A Distributed Algorithm for Identifying Information Hubs in Social Networks. IEEE Journal on Selected Areas in Communications (JSAC), 2013.

- M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, Jia Wang. Characterizing Geospatial Dynamics of Application Usage in a 3G Cellular Data Network. IEEE INFOCOM, 2012.

2 Quality of Experience for Mobile Video

2.1 Introduction

Online video services such as YouTube, Netflix, and Hulu are very popular on mobile networks. It has been estimated that video currently makes up more than half of all mobile data traffic and will grow by a factor of 16 by 2017 [13]. Therefore, it is crucial for mobile network operators to monitor the user experience, or Quality of Experience (QoE), of video streaming and understand how network characteristics and performance influence it.

Unfortunately, prior approaches for monitoring and understanding the user experience of video streaming are insufficient for mobile network operators. Recent seminal work [15, 16, 31, 55] investigated how video streaming quality influences important user engagement metrics, such as video abandonment rate. However, these studies rely on client-side instrumentation to measure video quality metrics such as buffering, startup delay, and bitrate. This instrumentation is not available to network operators, so the ability to measure user engagement using only network-side measurements is crucial from their perspective. Other work used network traffic analysis to study video streaming volume and abandonment rates in wired [36,39] and wireless networks [33]. However, these techniques use deep-packet-inspection to extract information beyond TCP/IP headers, which requires significant computational resources to employ at the scale of network carriers and can pose privacy problems

in practice. Moreover, these studies did not provide insight into how network characteristics and performance influence abandonment rates.

To redress these limitations, this work presents the first large-scale study to characterize video streaming performance in cellular networks and its impact on user engagement. Our study is based on month-long anonymized data sets collected from the core network and radio access network of a tier-1 cellular network in the United States. We analyze 27 terabytes of video streaming traffic from nearly half a million users in this data set. Our analysis makes two main contributions.

First, to the best of our knowledge, our analysis is the first to quantify the impact that network characteristics have on mobile video user engagement in the wild. We quantify the effect that 31 different cellular network factors have on video abandonment rate and video skip (*e.g.*, fast forward) rate. In particular, we quantify user engagement by labeling video streaming sessions in our data set as `completed`/`abandoned` and `skipped`/`non-skipped`, and then evaluate the extent to which core network and radio network factors correlate with abandonment rate and skip rate. These factors include TCP flow throughput, flow duration, handover rate, signal strength, and the physical location's land cover type. Our results provide network operators insights and direct guidance on how to improve user engagement. For example, improving mean signal-to-interference ratio by 1 dB reduces the likelihood of video abandonment by 2%. Moreover, reducing the load in a cell sector by 10 active users reduces the likelihood of video abandonment by 7%. Through these insights, network operators can identify and prioritize network factors that have the most impact on user engagement.

Second, we are the first to show how a network operator can monitor mobile video user engagement using only standard radio network statistics and/or TCP/IP flow records, a necessity for continuous monitoring at scale and for mitigating privacy concerns. Moreover, we show that our approach can predict video abandonment very early in a video session, which can help future networks decide which users to optimize performance for (*e.g.*, using LTE

self-organizing networks [1]). Specifically, we model the complex relationships between network factors and video abandonment. We find that the C4.5/M5P algorithm with bootstrap aggregation can build decision/regression tree models that accurately predict video abandonment. Our results show that it can predict whether a video streaming session is `abandoned` or `skipped` with more than 87% accuracy by observing only the initial 10 seconds. Our model achieves significantly better accuracy than prior models that require video service provider logs [15, 16], while only using standard radio network statistics and/or TCP/IP headers readily available to network operators.

The rest of this chapter is organized as follows. In Section 2.2, we present a brief background and details of the data collection process. Section 2.3 presents the characterization of video mobile video streaming performance and its impact on user engagement. We develop a machine learning model for user engagement and present the results in Section 2.4. Section 3.6 reviews related work and the chapter is concluded in Section 4.7.

2.2 Data

To study mobile video streaming performance, we collected anonymized flow-level logs from a tier-1 cellular network in the United States. Next, we first provide a brief background of video streaming in cellular networks, description of our data collection methodology, and some high-level statistics of the collected data set.

2.2.1 Cellular Network Background

A typical UMTS cellular network, shown in Figure 2.1, can be visualized as consisting of two major components: Radio Access Network (RAN) and Core Network (CN). RAN consists of NodeBs and Radio Network Controllers (RNCs). Each NodeB has multiple antennas, where each antenna corresponds to a different cell sector. A user via user equipment (UE) connects to an *active set* of one or more cell sectors in the RAN. The UE periodically selects a primary

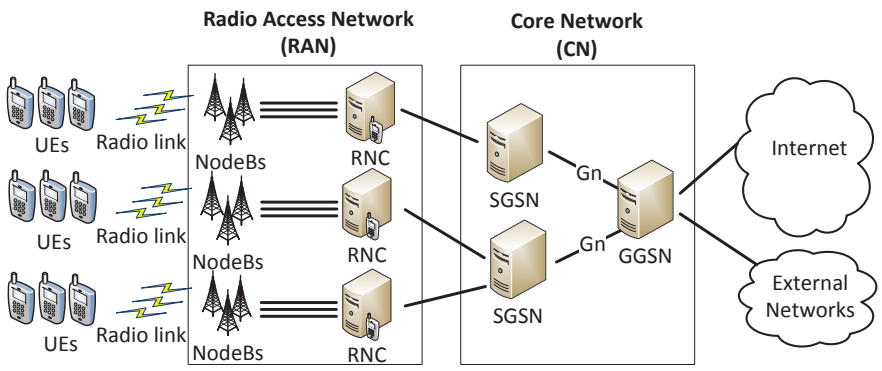


Figure 2.1. Cellular network architecture. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

or serving cell among the active set based on their signal strength information. From the active set, only the primary cell actually transmits downlink data to the UE. The traffic generated by a UE is sent to the corresponding NodeB by cell sectors. Each RNC controls and exchanges traffic with multiple NodeBs, each of which serves many users in its coverage area. RNCs manage control signaling such as Radio Access Bearer (RAB) assignments, transmission scheduling, and handovers. Each UE negotiates allocation of radio resources with the RAN based on a wide range of factors, such as available radio resources and signal strength [14].

CN consists of Serving GPRS Support Nodes (SGSNs) facing the user and Gateway GPRS Support Nodes (GGSNs) facing the Internet and other external networks. RNCs send traffic from NodeBs to SGSNs, which then send it to GGSNs. GGSNs eventually send traffic to external networks, such as the Internet. For data connections, the IP layer of a UE is peered with the IP layer of GGSNs in the form of tunnels known as Packet Data Protocol (PDP) contexts. These tunnels, implemented as GPRS Tunneling Protocol (GTP) tunnels, carry IP packets between the UEs and their peering GGSNs. From the perspective of an external network such as the Internet, a GGSN connecting CN to the Internet appears just like an IP router and the UEs that connect through the GGSN appear as IP hosts behind the router.

2.2.2 Data Collection and Pre-processing

For our study, we simultaneously collected two anonymized data sets from the RAN and CN of a tier-1 cellular network in the United States. Our data collection covers a major metropolitan area in the Western United States over the duration of one month in 2012. The RAN data set is collected at the RNCs and contains event-driven signaling information such as current active set, RAB state, handovers, bitrate, signal strength, and RRC requests from users and corresponding responses from the network. The CN data set is collected from the Gn interfaces between SGSNs and GGSNs, and contains flow-level information of video streaming traffic such as server IP and port, client IP and port, flow duration, TCP flags, anonymized user identifier (IMSI), and anonymized device identifier (IMEI). These fields require only TCP/IP or GTP level information, which is efficiently collected.

In order to determine the ground-truth of video abandonment, we also collected the following HTTP-level information: URL, host, user agent, content type, content length, and byte-range request from clients and response from servers. Large scale monitoring tools often do not collect HTTP information because it requires processing hundreds of bytes of text beyond the 40-byte TCP/IP header. Thus, it is important that day-to-day monitoring does not require its collection at scale. All device and user identifiers (*e.g.*, IMSI, IMEI) in our data sets are anonymized to protect privacy without affecting the usefulness of our analysis. The data sets do not permit the reversal of the anonymization or re-identification of users.

To minimize the confounding factors that different content providers (live vs. video-on-demand), connectivity (cellular vs. cable), and device type (mobile vs. desktop) could have on our network-centric analysis, we chose to focus on the most popular video service provider in our cellular network data set. This provider (anonymized for business confidentiality) serves user generated content on demand, and according to a recent study [33], it serves over 37% of all video objects. This provider streams videos using progressive download with byte-range requests, which is the most common protocol currently in use. We believe the conclusions we draw in this work apply to 9 of the 14 most popular mobile video content

providers as they use the same protocol [33]. Previous work found the top providers that use this protocol behave similarly in wired networks [83].

Since our collected data contains traffic records for all types of content, we first need to separate video streaming traffic from the rest. Towards this end, we use the HTTP host and content-type headers to separate the video streaming traffic from other TCP/IP traffic. We can also separate video traffic based only on the server IP and port, since all video streaming traffic is served by a known block of CDN cache servers.

A video is progressively downloaded in one or multiple HTTP byte-range requests, which represent different portions of the video [33]. Figure 2.2 illustrates a video streaming session that involves multiple HTTP byte-range server response flows. The x-axis represents time, which starts with the first HTTP byte-range server response flow. The y-axis represents byte-range of the video file with maximum value same as the video file size, which is highlighted by the horizontal broken line. Consequently, each gray rectangle represents a distinct HTTP byte-range server response flow. Note that flows may have different byte-range lengths and durations, they may be overlapping, and there may be time gaps between consecutive flows.

For the purpose of our analysis, we group HTTP flows into video sessions based on a unique ID that is the same in the URLs of each video session. In practice, we found that we can group flows into sessions without any HTTP information. In particular, by looking for a change in the server IP to determine when a new video session for a user starts, we can detect session starts correctly with 98% accuracy. This is because videos are served from a CDN and different videos are likely served from different cache servers. Even if all videos were served from a single server, we found that we can still detect session starts with 97% accuracy using a simple decision tree classifier trained on the size of and inter-arrival time gap between HTTP flows. (We omit details due to space constraints.) Thus, we conclude that TCP/IP information would be sufficient to detect and group HTTP flows into video sessions.

2.2.3 Video Traffic Statistics

Next we present some details of our collected data set such as aggregate statistics, container types, encoding bitrates, and video player types. Overall, our data set consists of more than 27 terabytes worth of video streaming traffic, from more than 37 million flows, from almost half a million users over the course of one month.

Our data set mostly contains standard definition video streaming streaming traffic. Figure 2.3(a) shows the distribution of video streaming traffic with respect to container types. The most common container types [5] are: (1) 3GP (3GPP file format), (2) MPEG-4, (3) FLV (Flash), (4) WebM, and (5) MP2T (MPEG transport stream). We observe that a vast majority, almost 70%, of video streaming traffic uses the 3GPP container type – followed by MPEG-4 and Flash container types as distant 2nd and 3rd most popular, respectively. Only a small fraction, less than 2%, of the video streaming traffic belongs to containers types used for live content. We exclude these from our analysis since our focus is on video-on-demand streaming. Further analysis of video encoding bitrate showed that a majority of video streaming traffic belongs to lower bitrates, which correspond to 144/240p video resolution. 240p is the most commonly used video resolution. Only a small fraction of video streaming traffic belongs to higher video resolutions. For example, less than 5% video streaming traffic belongs to high definition (HD) 720p content.

Short duration videos account for most of the streaming traffic in our data set. In Figure 2.3(b), we plot the cumulative distribution function (CDF) of video duration. We observe that more than 70% videos are less than 5 minutes long and only 10% videos are longer than 15 minutes. This type of skewed distribution is expected for content providers that serve user generated content [33].

2.2.4 Quantifying User Engagement

As a first step towards analyzing user engagement, we discuss two ways to quantify it: discrete and continuous.

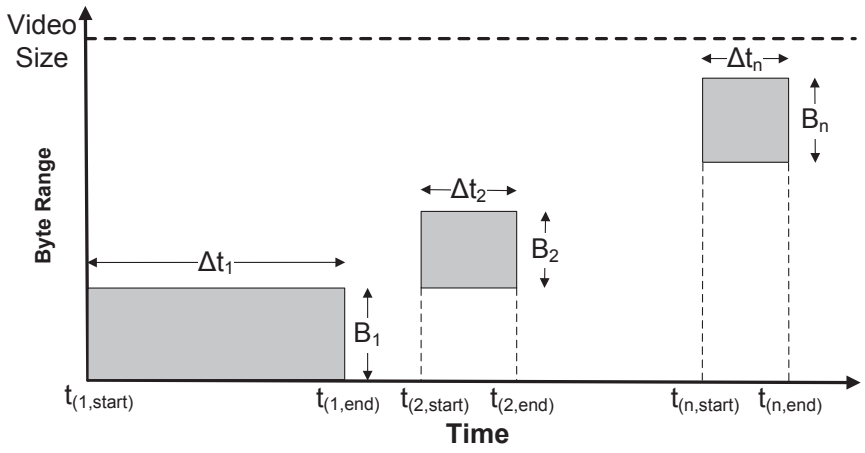
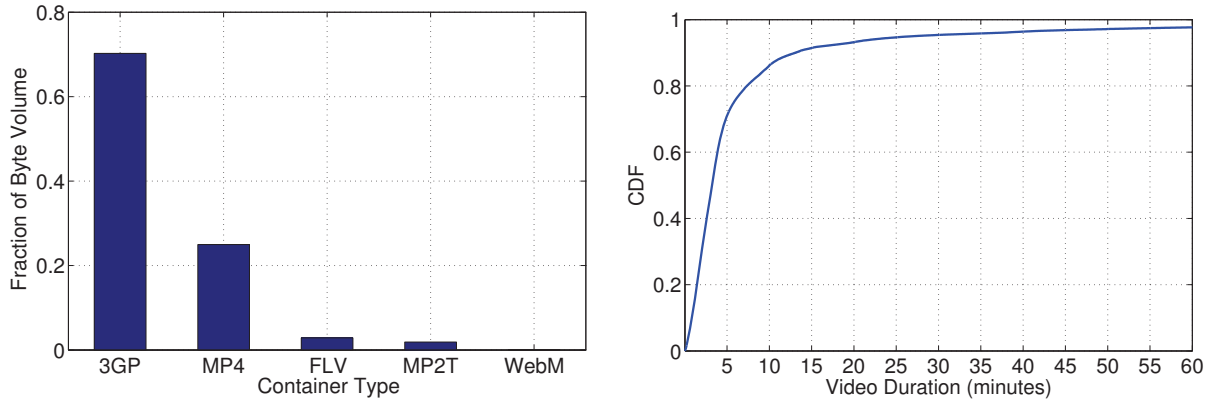


Figure 2.2. Illustration of a video streaming session. Gray rectangles represent distinct flows in a session.

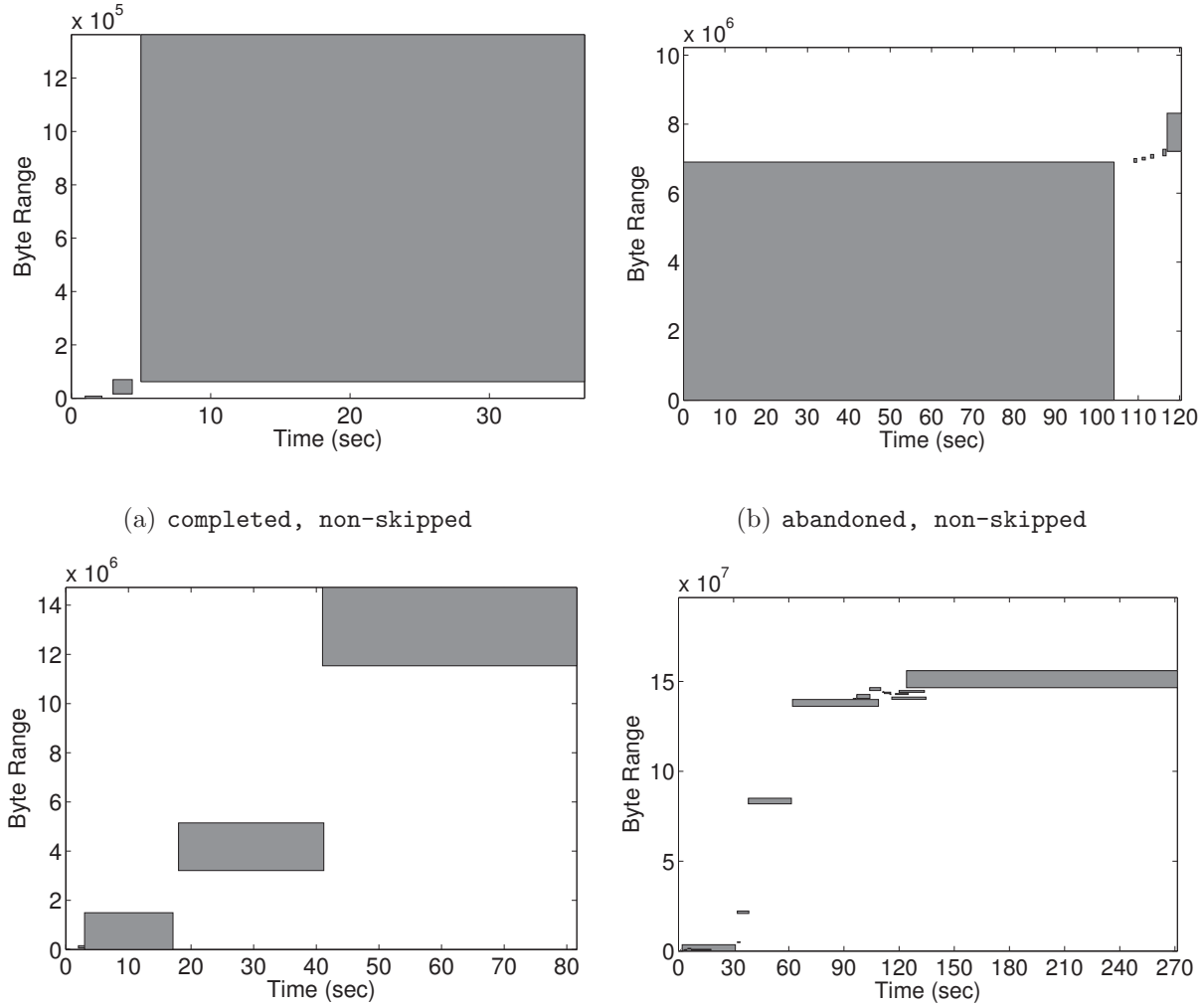


(a) Container type

(b) Video duration

Figure 2.3. Distributions of container type, video duration, and video player type

For discrete quantification, we first use a nominal variable that represents the following classes: `completed` and `abandoned`. The `completed` class represents video streaming sessions in which the download process reaches the end-point. The `abandoned` class represents video streaming sessions in which the download process is abandoned before reaching the end-point. In our data set, 21.2% sessions belong to the `completed` class and 78.8% sessions belong to the `abandoned` class. Since users tend to skip videos when streaming gets stuck, we also use a nominal variable that represents the following classes: `skipped` and `non-skipped`.



(a) `completed, non-skipped`

(b) `abandoned, non-skipped`

(c) `completed, skipped`

(d) `abandoned, skipped`

Figure 2.4. Examples of video streaming session classes. Y-axis limits are set to the video sizes.

The `skipped` class represents video streaming sessions in which the download process includes at least one seek-forward between the start-point and the last byte downloaded. The `non-skipped` class represents video streaming sessions in which the download process does not include seek-forward between the start-point and the last byte downloaded. In our data set, 33.9% sessions belong to the `skipped` class and 66.1% sessions belong to the `non-skipped` class. Combining the aforementioned user engagement classification schemes, we can define the following four non-overlapping classes: (1) `completed, non-skipped`, (2) `abandoned, non-skipped`, (3) `completed, skipped`, and (4) `abandoned, skipped`. In our data set,

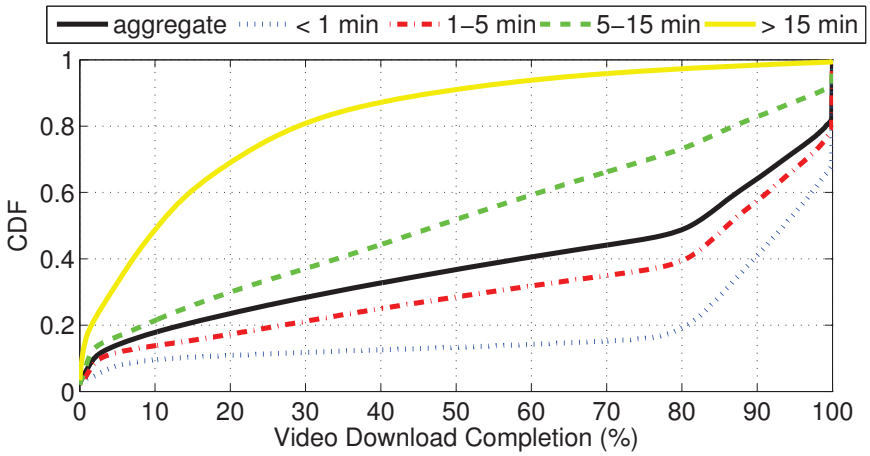


Figure 2.5. Distribution of video download completion

17.6% sessions belong to the `completed, non-skipped` class, 48.5% sessions belong to the `abandoned, non-skipped` class, 3.6% sessions belong to the `completed, skipped` class, and 30.3% sessions belong to the `abandoned, skipped` class. Figure 2.4 illustrates examples of video streaming sessions for the four user engagement classes. As mentioned earlier and observable in Figure 2.4, sessions generally consist of more than one flow. On average, a video streaming session in our data set consists of 11 flows, where earlier flows tend to be larger than the following flows. This trend is because video players tend to aggressively download larger chunks to fill up the available buffer during the initial buffering phase of a video streaming session [83]. The download rate in this initial phase is limited by the end-to-end available bandwidth. Afterwards in the steady state phase, the remaining video is generally downloaded in multiple smaller flows. The download rate in this phase depends on the video encoding rate and the playback progress.

For continuous quantification, we use a continuous variable ($\in [0,1]$) representing the fraction of video download completion. Figure 2.5 shows the CDF of video download completion. Comparing videos of different durations, we observe that shorter videos achieve higher download completion than longer videos. For aggregate distribution, almost 15% of video streaming sessions are abandoned with less than 5% download completion. However, after

Table 2.1. Core network features. i denotes the flow index of a session with N flows.

Feature	Description
Flow volume	(B_i) The number of bytes transferred during the i^{th} flow. (Summary stats)
Flow duration	(t_i) The duration (in seconds) from the SYN packet to the last packet in the i^{th} flow. (Summary stats)
Flow TCP throughput	(T_i) The ratio of flow volume to flow duration in the i^{th} flow, in KB/s. (Summary stats)
Flow inter-arrival time	(I_i) Time (in seconds) between the end of the i^{th} flow and the start of the $i+1^{\text{th}}$ flow. (Summary stats)
Flow flags	FIN_i and RST_i respectively denote the number of packets with TCP-Finish (no more data from sender indicating completion) and TCP-Reset (reset the connection indicating some unexpected error) flags set in the i^{th} flow. Based on the direction of packet transfer, we distinguish between client-to-server (c→s) and server-to-client (s→c) flags. (Summary stats)
Largest flow volume	(B_j) The largest flow volume among all flow volumes, where j denotes the index of this flow.
Largest flow duration	(t_j) The duration of the j^{th} flow.
Largest flow TCP throughput	(T_j) The throughput of the j^{th} flow.
Largest flow flags	FIN_j and RST_j respectively denote the number of packets with TCP-Finish and TCP-Reset flags set in the j^{th} flow. We distinguish between c→s and s→c flags.
Number of flows	(N) The total number of flows in a session.
Session volume	(**B**) The sum of all flow volumes in a session.
Session duration	(**t**) The sum of all flow durations in a session. $\mathbf{t} = \sum_{i=1}^{N} t_i$.
Session TCP throughput	(**T**) The average throughput of a session. $\mathbf{T} = \sum_{i=1}^{N} B_i \ / \ \sum_{i=1}^{N} t_i$.
Session inter-arrival time	(**I**) The sum of all flow inter-arrival times (in seconds) in a session
Session flags	(**FIN** and **RST**) respectively denote the number of packets with TCP-Finish and TCP-Reset flags set in a session. We distinguish between c→s and s→c flags.

the 5% completion mark, the distribution is fairly uniform until the 80% completion mark. The initial modality in the distribution indicates abandonment that is likely either because users tend to sample videos [15] or due to longer join times [31]. The later modality in the distribution (excluding the 100% completion mark) indicates abandonment that is likely either because users lose interest in the content (*e.g.*, due to video closing credits) or because shorter videos achieve higher download completion due to aggressive initial buffering.

We note that our definitions of user engagement detect abandonment and skips only during the download phase of a video. We cannot detect a video abandonment or skip if these events occur after a video has downloaded completely (*e.g.*, due to lack of user interest). However, network operators are typically not interested in those events because they are unlikely to be influenced by network factors.

Table 2.2. Radio access network features.

Feature	Description
# soft handovers	(H_S) This handover occurs when a cell is added or removed from the active set [105]. (Session- and cell-level)
# inter-frequency handovers	(H_{IF}) This type of handover occurs when a UE switches to cell sector of the same or different NodeB with different operating frequency [105]. (Session- and cell-level)
# IRAT handovers	(H_{RAT}) This type of handover occurs when a UE switches between different radio access technologies (*e.g.*, UMTS and GPRS) [105]. (Session- and cell-level)
# RRC failure events	A RRC failure event is logged when a request by a user to allocate more radio resources is denied by the respective RNC due to network overload or other issues [90]. (Session- and cell-level)
# admission control failures	These events occur when a user cannot finish the admission control procedure often due to lack of available capacity. (Session- and cell-level)
Received signal code power	RSCP is the RF energy of the downlink signal obtained after the correlation and descrambling process [105]. It is usually measured in dBm. (Summary stats)
Signal energy to interference	This ratio (E_c/I_o) denotes the ratio of the received energy to the interference level of the downlink common pilot channel [105]. It is usually measured in dB. (Summary stats)
Received signal strength	RSSI takes into account both RSCP and E_c/I_o [105]. It is usually measured in dBm. It is defined as: RSSI = RSCP - E_c/I_o. (Summary stats)
Size of active set	(S_{AS}) The number of unique cell sectors in the active set. (Summary stats)
Uplink RLC throughput	(T_U) The uplink data rate for UE in the DCH state (in kbps). (Session- and cell-level summary stats)
Downlink RLC throughput	(T_D) The downlink data rate for UE in the DCH state (in kbps). (Session- and cell-level summary stats)
# Users in DCH state	(U_{DCH}) Users served by the representative cell over a window of 1 hour.
Landcover	A nominal variable that defines the geographical terrain of a cell. 2006 National Land Cover Database contains the 16-class geographical terrain categorization of the United States at a spatial resolution of 30 meters [3,37]. The categories include developed-open space, developed-high intensity, perennial ice/snow, deciduous forest, open water, *etc.* We extract the top-3 most common landcover categories in terms of spatial area within 1 km of the representative cell.
Elevation	Elevation of a cell is extracted from the National Elevation Dataset (NED) at a spatial resolution of 30 meters [6]. We use average elevation of the representative cell as a feature.

2.3 Analysis of Network Factors

Our main goal is to understand the influence of network factors on user engagement. Towards this end, this section presents an in-depth analysis of the relationships between network factors and video abandonment.

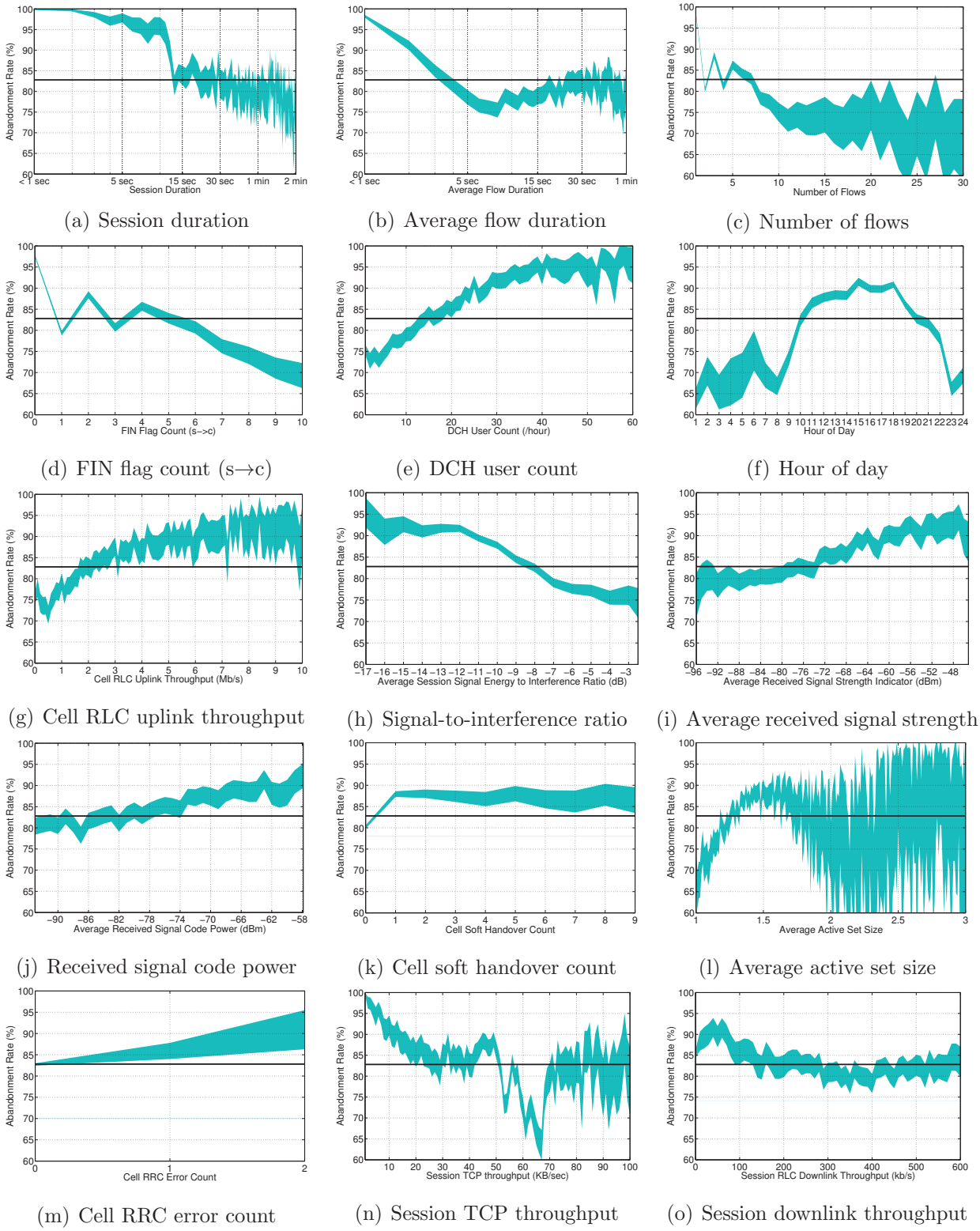
We first itemize a wide range of factors that can potentially impact or be influenced by mobile video user engagement. We compile a comprehensive list of features from the information available in both CN and RAN data sets. It is noteworthy that while features

extracted from the RAN data set are only applicable for cellular networks, features extracted from the CN data set are applicable for other kinds of wired and wireless networks as well.

For each video streaming session, we can extract CN features for individual flows and the whole session (labeled as *Flow* and *Session* features in Table 2.1, respectively). Since sessions may have different number of flows, we compute the following statistical measures to summarize the flow-level features for whole sessions: mean, standard deviation, minimum, maximum, 25^{th} percentile, median (50^{th} percentile), and 75^{th} percentile. Hence each flow-level feature listed in Table 2.1 (labeled with "Summary stats") represents 7 summary values. We also extract these features for the largest flow (in terms of byte volume) of a video streaming session, as a single flow typically dominates each video session.

For each video streaming session, we also extract RAN features for the user and the cell sectors that service the user during the session. The RAN features are described in Table 2.2. For session-level features, the RAN data set records belonging to a user can be identified using the combination of IMSI and session start and end timestamp information. For cell-level features, however, the selection criterion of the representative cell for a session is not obvious because the active set and the primary cell may change between the start and end of a session. Towards this end, we select the most common cell sector (in terms of number of records) to be the representative cell for a session. For each session, cell-level features are computed for all users served by the representative cell in the time window at the session start. Features in Table 2.2 labeled with "Session- and cell-level" indicate features that we compute both a session-level value and cell-level value, as defined above. For example, for # soft handovers, we compute one feature as the number of soft handovers for the user during the session, and another as the number of soft handovers for all users in the representative cell of that session. For features that can hold multiple values during a session (*e.g.*, RSSI), we compute the same 7 summary statistic values listed above for flow features. These features are labeled with "Summary stats" in Table 2.2.

To better understand the relationship between features and user engagement, we plot the



(a) Session duration (b) Average flow duration (c) Number of flows

(d) FIN flag count (s→c) (e) DCH user count (f) Hour of day

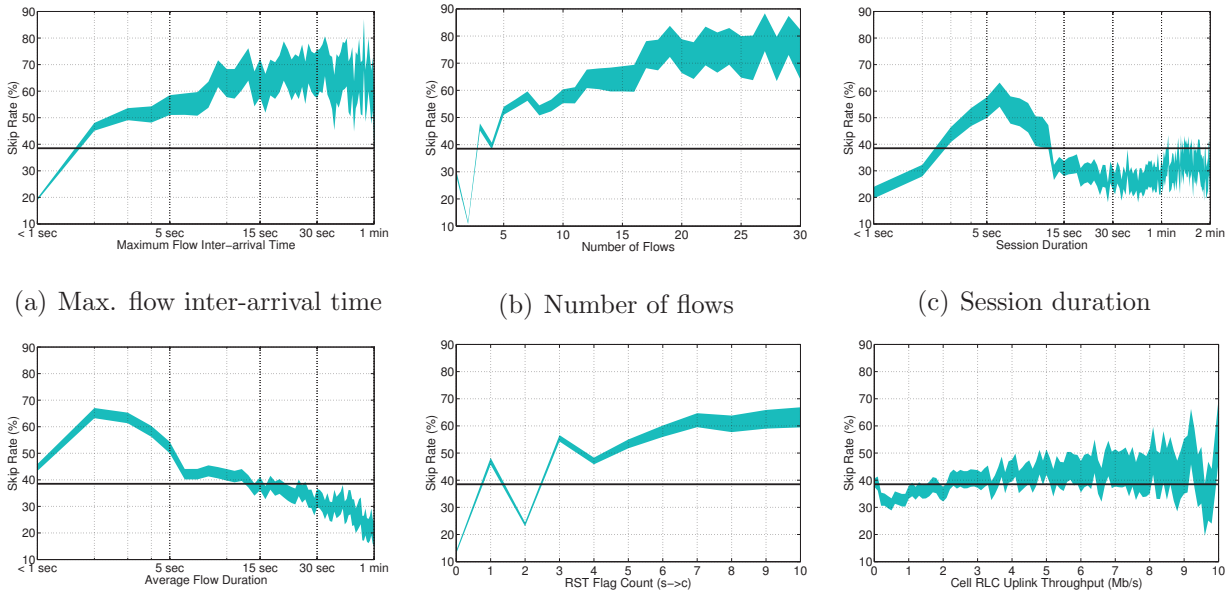
(g) Cell RLC uplink throughput (h) Signal-to-interference ratio (i) Average received signal strength

(j) Received signal code power (k) Cell soft handover count (l) Average active set size

(m) Cell RRC error count (n) Session TCP throughput (o) Session downlink throughput

Figure 2.6. Abandonment rate distributions. Shaded areas represent confidence intervals.

abandonment rate distributions of prominent features in Figure 2.6. The abandonment rate is defined as the fraction of sessions in the data set that are abandoned. The shaded areas



(a) Max. flow inter-arrival time (b) Number of flows (c) Session duration

(d) Average flow duration (e) RST flag count (s→c) (f) Cell RLC uplink throughput

Figure 2.7. Skip rate distributions. Shaded areas represent the 95% confidence interval.

represent the 95% confidence interval [112]. The horizontal line in each plot denotes the average abandonment rate. Figure 2.6 suggests the following implications:

Abandoned sessions are shorter. Although this result is expected, we find that each measure of session length provides unique information. Figure 2.6(a) shows sessions shorter than 15 seconds are significantly more likely to be abandoned. The sharp inflection point may be due to automated failure of sessions that do not complete the initial buffering phase. Similarly, Figure 2.6(b) shows a sharp drop in abandonment rate for sessions with average flow duration longer than 1-3 seconds. Figures 2.6(c) and 2.6(d), both measures of flow count, show that sessions with more flows are less likely to be abandoned. Thus, each of these features provides information useful for detecting abandonment.

Network load increases the abandonment rate. Despite the low bitrate of video streams relative to the capacity of a cell (∼500 kbps vs. 3-14 Mbps), we find there is a nearly linear relationship between various measures of RAN load and abandonment rate. For example, Figure 2.6(e) shows that the abandonment rate goes up by roughly 7% for each 10 active users in a sector, even though these resources are scheduled in a proportional fair manner

every 2 ms [14]. This load relationship can also be seen in Figure 2.6(f), which shows that abandonment rate is highest during the peak load hours of the day and much lower during the off-peak hours. This effect can be explained by Figure 2.6(g), which shows that the abandonment rate begins to grow when aggregate cell uplink throughput is just 50 kbps, significantly less than the cell capacity. This is likely because even small amounts of uplink traffic can cause interference, and Figure 2.6(h) shows that abandonment rate decreases by 2% for each dB increase in the signal-to-interference ratio (E_c/I_o). Furthermore, Figures 2.6(i) and 2.6(j) show that the abandonment rate increases as RSSI and RSCP increase, contrary to the general belief that higher received power means a better user experience. These E_c/I_o, RSSI, and RSCP results strongly suggest that users with higher received power also experience more interference. Hence, user engagement in our data set is more limited by interference rather than poor coverage. In summary, these results suggest that measures a cellular operator takes to reduce per-sector load and interference will improve user engagement in a roughly linear manner.

Handovers increase the abandonment rate. Another important question for operators is whether cell handovers disrupt the user experience. Our results suggest that all handover types are correlated with a decrease in user engagement. Figure 2.6(k) shows that cells with soft handovers, which are "make-before-break" handovers, have significantly higher abandonment rates. This result is supported by Figure 2.6(l), which shows increase abandonment rates for non-integral mean active set values (i.e., sessions that incurred active set additions or deletions during soft handovers). These effects may be partially due to the RRC signalling required for handover. Figure 2.6(m) shows that when RRC signalling errors occur, abandonment rate increases as well.

Higher throughput does not always mean lower abandonment. Although measured throughput is often used as a proxy for network quality (*e.g.*, [39]), our results suggest higher average throughput does not always indicate lower abandonment. Figures 2.6(n) and 2.6(o) show that abandonment rate decreases as average TCP and RLC throughput increases up to a

point. However, the abandonment rate is lowest at TCP throughput equal to the steady state streaming rate, and it grows for higher throughput values. This pattern is because early abandonment, while the video is still in the non-rate-limited buffering phase, actually results in higher average throughput than watching a video through the rate-limited steady state phase.

In Figure 2.7, we plot the skip rate distributions of prominent features. The skip rate is defined as the fraction of sessions in the data set that are skipped. Due to space constraints, we only plot the skip rate curves for features that have different trends than the respective abandonment rate curves. The shaded areas represent the 95% confidence interval [112]. The horizontal line in these plots denotes the average skip rate.

We note that skip rate has a direct relationship with maximum flow inter-arrival time (Figure 2.7(a)) and number of flows (Figure 2.7(b)). This is likely because skips result in more flows and larger gaps between them. Skip rate peaks at session and flow durations of just a few seconds (Figures 2.7(c) and 2.7(d)), suggesting that users chose to skip early in a session, either due to network issues or lack of interest. Figure 2.7(e) shows larger RST flag count correlated with higher skip rate likely because skips cause connection resets. These contrasting patterns imply that it is more challenging to measure both skips and abandonment than a single engagement metric.

2.4 Modeling User Engagement

In this section, we develop models to accurately predict user engagement using only standard radio network statistics and/or TCP/IP header information.

2.4.1 Background and Problem Statement

Network operators would like to predict user engagement for three main applications. First, directly estimating these metrics from network traffic requires cost-prohibitive collection of

sensitive data (requiring deep-packet-inspection) beyond TCP/IP headers. Thus, cost and privacy concerns would be alleviated with a model that accurately predicts these engagement metrics using only standard radio network statistics and/or TCP/IP header information that is already collected. A simple and efficient model would be able to monitor video engagement metrics over an entire network in real-time to facilitate trending and alarming applications. Second, self-organizing networks [1] (SON) enable mobile networks to adapt resource allocation dynamically. Thus, the ability to accurately predict video abandonment early in a video session can help guide SONs to provide more resources to the most vulnerable sessions. Third, an interpretable model that relates network factors to user engagement metrics can help network operators in prioritizing infrastructure upgrades by identifying the combination of factors that need to be adjusted for improving engagement.

Our goal is to jointly use the available features to accurately model both nominal and continuous measures of user engagement (defined in Section 2.2). Moreover, we want our models to make the prediction decisions as early as possible in a video session. Therefore, we define the modeling problem as follows: *given the feature set computed over the initial τ seconds ($\tau \leq \mathbf{t}$) of a video session, predict the user engagement metric.*

2.4.2 Proposed Approach

As we observed in Section 2.3, many network features are not independent of each other and the relationships among them can be non-linear. Therefore, modeling user engagement given all available features is a non-trivial prediction task. Furthermore, our modeling approach should answer pertinent questions such as: Which features are more useful for prediction? How many features do we need to reap a substantial accuracy gain?

To address these challenges, we use a machine learning approach for modeling the complex relationships between network features and user engagement metrics. The choice of learning algorithm is crucial to successfully modeling feature interdependence and non-linearity. After some pilot experiments, we found that decision tree algorithms with bootstrap aggregation

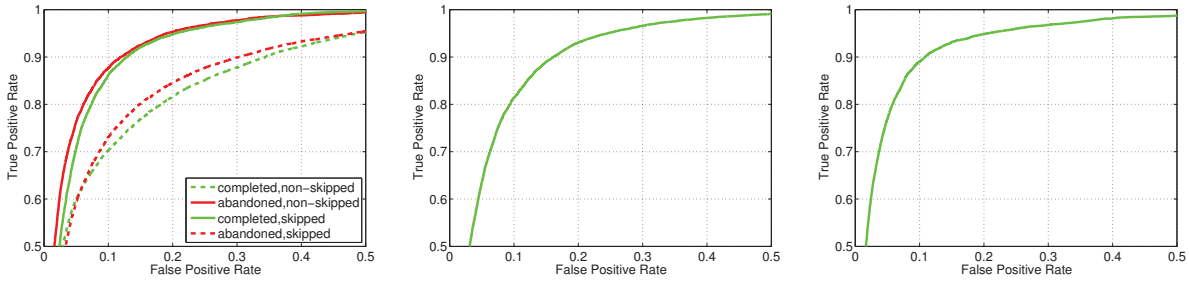
(or bagging) [113] work well for both nominal (classification) and continuous (regression) user engagement metrics. Other commonly used Bayes and linear regression algorithms were outperformed by the decision tree algorithms in our pilot experiments. Decision trees do not require feature independence assumption and can handle non-linearities by employing multiple splits/breaks for each feature. Furthermore, decision tree models comprise of simple if-then-else branches, which can process data efficiently. For our experiments, we used C4.5 decision tree algorithm [81] and M5P regression tree algorithm [82].

2.4.3 Experimental Setup

We evaluate the effectiveness of classification models in terms of the following standard Receiver Operating Characteristic (ROC) metrics [35]: (1) True Positives (TP), (2) True Negatives (TN), (3) False Positives (FP), and (4) False Negatives (FN). We summarize the classification results in terms of the following ROC metrics: True positive rate $= \frac{|\text{TP}|}{|\text{TP}|+|\text{FN}|}$, False positive rate $= \frac{|\text{FP}|}{|\text{FP}|+|\text{TN}|}$, and Accuracy $= \frac{|\text{TP}|+|\text{TN}|}{|\text{TP}|+|\text{TN}|+|\text{FP}|+|\text{FN}|}$. We also plot the standard ROC threshold curves in our evaluation. An ideal ROC threshold curve approaches the top-left corner corresponding to 100% true positive rate and 0% false alarm rate. The Area Under Curve (AUC $\in [0, 1]$) metric summarizes the classification effectiveness of an ROC threshold curve, where the AUC values approaching 1 indicate better accuracy. Besides, we evaluate the effectiveness of regression models in terms of the standard root-mean-square error (RMSE $\in [0, 1]$) metric.

To avoid class imbalance and over-fitting during model training, we use k-fold cross-validation with class resampling [113]. In our pilot experiments, different values of k yielded very similar results. All experimental results reported in this work are presented for $k = 10$. Furthermore, we evaluate the feature sets on varying initial time window sizes: $\tau = \mathbf{t}$ (*i.e.*, use all available data), $\tau \leq 60$ seconds, and $\tau \leq 10$ seconds. We expect the classification accuracy to degrade for smaller initial time windows.

We separately evaluate the core network feature set (abbreviated as **CN**), the radio net-



(a) 4-way (b) `completed` vs. `abandoned` (c) `completed, non-skip.` vs. `rest`

Figure 2.8. ROC threshold plots for various class pairs

work feature set (abbreviated as **RAN**), and the combined feature set (abbreviated as **All**). The radio network and core network features are separately grouped because they require different types of instrumentation. Recall from Section 2.2, measuring radio network features requires instrumentation at RNCs and measuring core network features requires instrumentation at Gn interfaces in a cellular network.

2.4.4 Evaluation

Our experimental results demonstrate that the proposed machine learning model can predict both video abandonment and skips with high accuracy using only the initial 10 seconds of a session, while meeting the constraints of network operators. We find that although some features are more useful than the rest for prediction, using all available features results in significant accuracy gain as compared to using only a few top features. Moreover, our decision and regression tree models are interpretable; they inform us about the relative usefulness of features by ordering them at different tree levels and we can understand the specific set of network conditions that impact user engagement by following each path in the tree. Many of these conditions can be influenced by a network operator, and thus provide guidance on how to improve user engagement in different situations.

Next we present detailed results for both classification and regression. For classification, we build decision tree models to predict both individual classes and their combinations. For

Table 2.3. Accuracy of 4-way classification

Feature Set	`completed non-ski.` (%)	`abandoned non-ski.` (%)	`completed skipped` (%)	`abandoned skipped` (%)	Avg. (%)
$\tau = \mathbf{t}$					
CN	72.0	78.4	76.2	73.4	75.0
RAN	64.1	53.7	73.2	55.7	61.7
All	73.1	77.8	77.4	74.4	**75.7**
$\tau \leq 60$ seconds					
CN	69.5	62.7	63.8	64.6	65.2
RAN	62.6	47.8	58.5	57.0	56.5
All	70.4	63.7	65.7	65.4	**66.3**
$\tau \leq 10$ seconds					
CN	69.5	59.6	63.3	65.3	64.4
RAN	60.5	46.6	59.0	57.4	55.9
All	69.6	60.7	64.9	65.5	**65.2**

individual classes, we train the decision tree algorithm for 4-way classification. By combining classes, we change the granularity at which the model predicts user engagement. We use the following two class pairs: `completed` vs. `abandoned` and `completed, non-skipped` vs. `rest`. For combined classes, we train the decision tree algorithm for 2-way classification. Naturally, we expect better accuracy for 2-way classification than 4-way classification because the model is trained at a coarser granularity.

For 4-way classification, we observe that the core network feature set outperforms the radio network feature set. Combining the core and radio network feature sets improves the average accuracy. In Table 2.3, we observe the best average accuracy of 75.7% for the combined feature set at $\tau = \mathbf{t}$ seconds. In practice, improvement in accuracy means that fewer sessions need to be measured before the network operator can be confident that a real change in user engagement has occurred and an alarm can be raised. For a cell sector serving only a handful of users simultaneously, this can mean a significant reduction in time to detection of issues since video sessions may not be frequent. For the combined feature set, ROC threshold curves are plotted in Figure 2.8(a). The ordering of ROC curves conforms with the class-wise accuracy results in Table 2.3. The best operating accuracy of 77.8% is observed for `abandoned, non-skipped` class, which corresponds to 95.5% AUC. As expected, in Table 2.3 we observe that the average accuracy degrades for smaller values of

Table 2.4. Accuracy of `completed` vs. `abandoned` and `completed, non-skipped` vs. `rest` classification

Feature Set	`comp.` (%)	`abandoned` (%)	Avg. (%)	`completed, non-ski.` (%)	`rest` (%)	Avg. (%)
$\tau = \mathbf{t}$						
CN	80.5	85.9	83.2	77.2	92.3	**88.5**
RAN	73.9	77.9	75.9	71.9	88.8	84.5
All	80.5	86.5	**83.5**	76.9	92.4	**88.5**
$\tau \leq 60$ seconds						
CN	79.5	82.1	**80.8**	78.0	91.5	**88.1**
RAN	74.1	78.4	76.3	72.7	88.4	84.4
All	78.8	82.6	80.7	77.6	91.4	88.0
$\tau \leq 10$ seconds						
CN	79.6	80.7	**80.1**	77.1	90.7	**87.3**
RAN	74.2	78.9	76.5	73.2	89.2	85.1
All	77.8	82.1	79.9	76.6	90.7	87.2

τ. We observe the best average accuracy of 65.2% for the combined feature set at $\tau \leq 10$ seconds, representing more than 10% accuracy reduction as compared to $\tau = \mathbf{t}$ seconds.

Since operators may only be interested in predicting video abandonment, it is also important to build models to accurately predict `completed` vs. `abandoned` and `completed, non-skipped` vs. `rest` class pairs (instead of all four classes). These class pairs compare the scenarios when users either abandon or skip the video streaming session. Table 2.4 presents the classification results for these two class pairs. As expected, we observe significant improvement in accuracy for both class pairs as compared to 4-way classification due to reduced number of classes. Moreover, we observe that the average accuracy suffers only minor degradation (less than 5%) as τ is reduced. For `completed` vs. `abandoned` class pair, we observe the best average accuracy of 83.5% for the combined feature set at $\tau = \mathbf{t}$ seconds. For the combined feature set, the ROC threshold curve is plotted in Figure 2.8(b), which corresponds to 93.4% AUC. For `completed, non-skipped` vs. `rest` class pair, we observe the best average accuracy of 88.5% for the combined feature set at $\tau = \mathbf{t}$ seconds. For the combined feature set, the ROC threshold curve is plotted in Figure 2.8(c), which corresponds to 95.1% AUC.

For regression, we build regression tree models to predict video download completion.

Table 2.5. Root-mean-square error of regression

Feature Set	Linear Regression	M5P Regression Tree
$\tau = \mathbf{t}$		
CN	0.25	0.15
RAN	0.30	0.27
All	0.23	**0.14**
$\tau \leq 60$ seconds		
CN	0.27	0.18
RAN	0.36	0.34
All	0.24	**0.17**
$\tau \leq 10$ seconds		
CN	0.29	0.22
RAN	0.37	0.34
All	0.28	**0.21**

Overall, we observe similar patterns across feature sets and varying initial window sizes for regression results as observed for classification results earlier. Table 2.5 presents the results of M5P regression tree algorithm and a simple linear regression algorithm. We note that M5P regression tree algorithm consistently outperforms the simple linear regression algorithm, indicating that M5P can successfully capture the non-linear dependencies between features and video download completion that are not modeled by the simple linear regression algorithm. RMSE is lower for larger τ values, and **All** feature set has the lowest RMSE as compared to individual **CN** and **RAN** feature sets. We observe the best RMSE of 0.14 for $\tau = \mathbf{t}$ and **All** feature set.

2.4.5 Discussion

Our evaluation highlighted that using all features together results in better classification/regression accuracy than using their subsets. To systematically analyze the utility of adding features to the classification/regression model, we plot accuracy versus feature set size for `completed` vs. `abandoned` classification in Figure 2.9. Towards this end, we iteratively rank the feature set using the following greedy approach: for k^{th} iteration, we evaluate the accuracy gain of the model by separately adding candidate features and selecting the $k+1^{th}$ feature which provides the best accuracy. The top features are related to session size and TCP throughput which we believe are correlated with user engagement, as sufficient

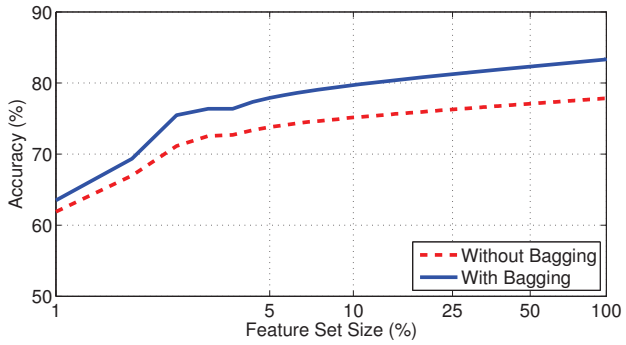
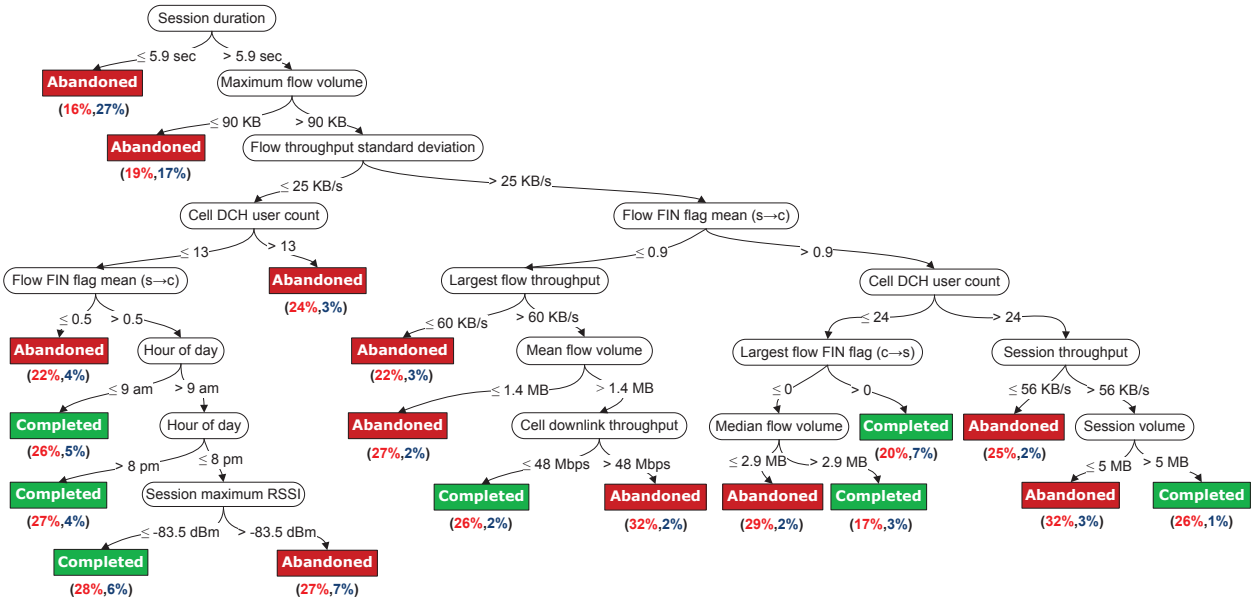


Figure 2.9. Accuracy vs. feature set size for `completed` vs. `abandoned` classification

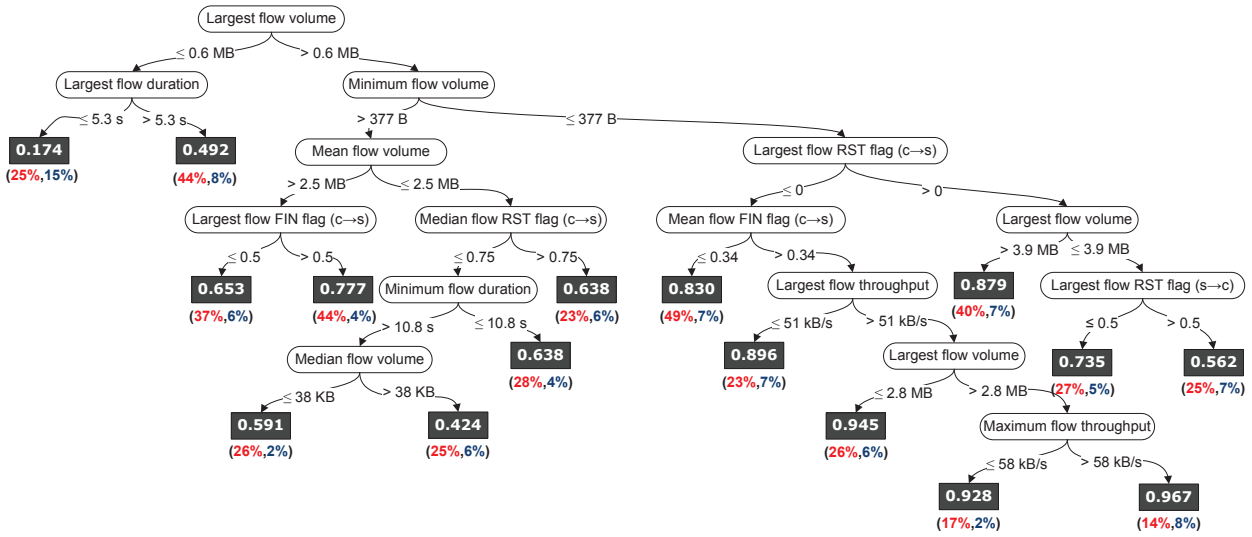
throughput is required for video streaming and abandonment results in low volume sessions. The plot shows that a few top features provide most of the accuracy gain. However, the gains in accuracy we achieve from including 5% to 100% of features are not diminishing. Thus, it makes sense to use all available features because the computational overheads of feature extraction and testing for additional features is low (in order of milliseconds).

The decision/regression tree models also provide actionable insights. The pruned versions of the decision tree model for `completed` vs. `abandoned` class pair and the regression tree model for video download completion are plotted in Figures 2.10(a) and (b), respectively. The tuples below the rectangular leaf nodes represent their error and population size. Due to space constraints, we only plot the tree models for $\tau \leq 10$ seconds which are useful for network operators to predict user engagement by observing only the initial 10 seconds of video streaming sessions.

From the model, network operators can identify network factors that may have the most impact on user engagement and make decisions to prioritize certain infrastructure upgrades. The features at the higher levels of a tree tend to have more distinguishing power and account for more population size than lower level features. The root node is session duration for Figure 2.10(a) and largest flow volume in Figure 2.10(b). However, it is noteworthy that the ordering of network factors in Figure 2.10 does not strictly determine their importance. First, trees shown in Figure 2.10 represent one of many candidate tree models generated during



(a) C4.5 decision tree model for `completed` vs. `abandoned` class pair



(b) M5P regression tree model for video download completion

Figure 2.10. Pruned decision and regression tree models for $\tau \leq 10$ seconds using **All** feature set. The tuples below leaf nodes represent (error%,population size%).

bagging – other candidate trees have different feature ordering. Second, these features are not independent – session duration and maximum flow volume (top two features in Figure 2.10(a)) jointly account for session throughput and largest flow throughput to some extent.

The paths from the root node to leaves represent the equivalent rule sets, which inform

network operators of the interdependence among multiple features. For the regression tree in Figure 2.10(b), if largest flow volume is ≤ 0.6 MB and largest flow duration is ≤ 5.3 seconds then video download completion prediction is 17.4%. In contrast, if largest flow volume is ≤ 0.6 MB and largest flow duration is > 5.3 seconds then video download completion prediction is increased to 49.2%. Moreover, for the decision tree in Figure 2.10(a), if session duration > 5.9 seconds and maximum flow volume is ≤ 90 KB after the first 10 seconds then our model predicts that the video session will be abandoned with 19% error probability.

A network operator can influence many network factors to improve user engagement. The feature splits in Figure 2.10 provide network operators actionable insights for this purpose. For example, the decision tree predicts a session to be abandoned if cell DCH user count is larger than a threshold under certain conditions. Most cellular network users are covered by multiple cell sectors, and handover algorithms use signal quality and sector load to determine which sector each user should receive data from. The thresholds used for handover are typically fixed at a single global value. However, the feature splits in Figure 2.10(a) suggest that the network operator can tolerate a higher cell sector load threshold for sessions with higher throughput variance than sessions with lower throughput variance (24 vs. 13 users occupying DCH channels).

Below, we discuss limitations of our analysis and results. First, our results are based on traces from a single video service provider that uses progressive download with byte-range requests. Therefore, our findings may not be representative of video service providers that use other streaming methods. Second, our user engagement model cannot differentiate between video abandonment due to network-related issues and due to lack of user interest. Distinguishing between these two cases requires either client- or server-side information, which is not available to network operators.

2.5 Related Work

Prior studies can be categorized based on whether they use network-side or user-side instrumentation.

2.5.1 Network-side Instrumentation

Our study builds upon previous work by Gill *et al.* [39], Finamore *et al.* [36], and Erman *et al.* [33]. Each of these studies characterized the abandonment rate of video streaming sessions by collecting passive network traces at a campus edge network, 5 different wired edge locations, and a cellular network, respectively. To estimate video quality, these studies use deep-packet-inspection techniques (*e.g.*, [86]) to understand the video provider protocol. Our finding that 77% of video sessions are not completely streamed is closest to Finamore's result (80%), whereas Gill and Erman found lower abandonment rates (50% and 60%, respectively). All these results indicate that abandonment rates are high. Based on the ratio of download rate to encoding bitrate of video, Gill *et al.* concluded that approximately 20% of the video streaming sessions were interrupted due to poor performance. However, we find that average throughput is not always a good indicator of abandonment rate.

Our work makes two significant contributions on top of these studies. First, in order to measure abandonment, previous studies relied on deep-packet-inspection to extract information beyond TCP/IP headers, which requires prohibitive computational resources to employ at the scale of network carriers and can pose privacy problems in practice. Our work demonstrates that we can accurately measure abandonment without such information. Second, these studies did not provide insight into how network characteristics and performance impact abandonment rates. Our study is the first to examine the relationship between mobile network factors and user engagement and the first to provide guidance on how operators can reduce video abandonment.

2.5.2 Client-side Instrumentation

In [83], Rao *et al.* conducted an active measurement study of video streaming traffic from YouTube and Netflix. They proposed models to express various properties of completed and interrupted video streaming traffic as a function of the video parameters. However, the authors did not study user engagement because this work is based on active measurement data.

Dobrian *et al.* conducted a large scale, passive, user-side study to understand the impact of video streaming quality on user engagement [31]. They used video player instrumentation to quantify video streaming quality metrics such as join time, buffering ratio, average bitrate, rendering quality, and rate of buffering. Their analysis showed that buffering ratio has the largest impact on user engagement for non-live content and average bitrate significantly impacts user engagement for live content. Krishnan and Sitaraman also conducted a large scale, passive, user-side study to understand the impact of video streaming quality on user engagement [55]. They quantified the impact of video streaming quality metrics on user engagement using quasi-experimental designs. In [15, 16], Balachandran *et al.* developed a QoE model using various user-side video quality metrics. Specifically, they developed a decision tree based machine learning model to predict the extent of the video watched by users. For the two-class problem (completed vs. interrupted/abandoned), their trained model achieved up to 70% accuracy which progressively decreases as the number of classes is increased.

While these studies analyzed user engagement using data collected via video player instrumentation, our work focuses on characterizing and modeling user engagement using network-side measurements. Modeling user engagement using network-side data is particularly important for network operators because they do not have access to video player instrumentation data. Interestingly, our model based on network-side data can predict whether a user completely downloads a video with more than 87% accuracy, which is significantly better than the client-side model developed by Balachandran *et al.* Furthermore, since our

model does not require client-side instrumentation, it can be used by any network operator, not just the video content provider.

2.6 Conclusions

This work represents the first characterization of mobile video streaming performance and models its impact on user engagement from the perspective of network operators. We observed that many network features exhibit strong correlation with abandonment rate and skip rate. Our proposed model achieved more than 87% accuracy by observing only the initial 10 seconds of video streaming sessions. Overall, we conclude that the proposed model based on standard radio network statistics and/or TCP/IP header information can be successfully used by network operators to predict video abandonment. Our model is useful for network operators to continuously monitor at scale to proactively mitigate the factors that can adversely impact user engagement.

3 Mobile Network Performance during Crowded Events

3.1 Introduction

Crowded events, such as football games, public demonstrations, and political protests, put an extremely high demand for communication capacity on cellular networks around the duration of the events [11]. Cellular networks are facing unprecedent challenges in dealing with such spiky demand. First, cellular network utilization has already been rapidly approaching its full capacity throughout the world due to the increasing prevalence of cellular devices such as smartphones, tablets, and Machine-to-Machine (M2M) devices. Even in the United States, cellular network usage is at an all-time high even under normal operating conditions and projections show that traffic volume will further increase by 26 times by 2015 as compared to 2010 [10,58]. Second, the spiky demand caused by crowded events is often extremely high because there maybe a large number (often tens of thousands) of users gathered in a small region (such as a football stadium) that is covered by only a small number of cell towers. Even worse, people tend to use their cellular devices more than usual during the events to either talk with their friends or access the Internet (such as uploading a photo to Facebook or a video clip to YouTube during a football game). Third, it is critical for cellular networks to cope with such high demand during crowded events because poor performance will affect a large number of people and cause widespread user dissatisfaction. Although cellular network

operators have deployed remediation solutions, such as portable base stations called Cells on Wheels (COWs) for temporarily increasing communication capacity and free Wi-Fi access points for offloading Internet traffic from cellular base stations, crowded events still remain a major challenge for cellular network operators.

To the best of our knowledge, this work presents the first thorough investigation of cellular network performance during crowded events. Based on the real-world voice and data traces that we collected from a tier-1 cellular network in the United States during two high-profile crowded events in 2012, we aim to answer the following three key questions.

How does cellular network performance degrade during crowded events as compared to routine days? To answer this question, we characterize cellular network performance during both the pre- and post-connection phases, which helps us to understand user experience before and after acquiring radio resources. For pre-connection phase, we find that pre-connection failures dramatically increase during the crowded events by 100-5000 times as compared to their average on routine days. These failures occur because when too many users attempt to acquire radio resources at the same time, they exhaust the limited bandwidth of the signaling channel resulting in connection timeouts and failures. We find that this resource exhaustion occurs not only at the event venue, but also as far as 10 miles around the event as users arrive and depart. Moreover, some failures, such as dropped and blocked voice calls, are most likely to occur in bursts just before, after, and during event intermissions. For post-connection phase, we find that voice network performance in terms of dropped and blocked calls degrades during crowded events by 7-30 times, and data network performance in terms of packet loss ratio and round trip time (RTT) degrades during crowded events by 1.5-7 times, compared to their average on routine days.

What causes the performance degradation? To answer this question, we analyze user traffic patterns in terms of both aggregate network load and user-level session characteristics. For aggregate network load, we find that uplink traffic volume increases by 4-8 times, and both downlink traffic volume and the number of users increase by 3 times, during the crowded

events as compared to their average on routine days. We conclude that the large number of users trying to access radio resources at the same time is a major cause of the observed excessive pre-connection failures. For user-level session characteristics, we find that the average byte volume per session decreases by 0.5 times during the events even though the average session length increases. Our investigation suggests that this change in workload is due to a change in application usage during these events, such as the increased use of online social networks. We conclude that lower byte volume per session, despite an increase in average session length, is a major cause of the waste of radio resources in the post-connection phase.

How would practical mitigation schemes perform in real-life? To answer this question, we investigate two practical mitigation schemes that do not require making significant changes to the cellular infrastructure: radio resource allocation tuning and opportunistic connection sharing. Radio resource allocation tuning addresses the issue of inefficient radio resource allocation in the post-connection phase by adjusting cellular network resource allocation parameters. Cellular networks allocate resources to each user using a Radio Resource Control (RRC) state machine, which is synchronously maintained by the network and devices. Different states of the RRC state machine correspond to different amount of radio resources allocated by the network and energy consumption by cellular devices. Since a large number of users contend for limited radio resources during crowded events, we show that more aggressive release of radio resources via 1-2 seconds shorter RRC timeouts helps to achieve a better tradeoff between wasted radio resources, energy consumption, and delay during crowded events. Note that cellular network operators often know the time and location of large crowded events beforehand; thus, it is practical for them to adjust cellular network parameters before events and restore them after events. Opportunistic connection sharing addresses increased pre-connection failures by aggregating traffic from multiple devices into a single cellular connection. That is, by having some devices share their cellular connection with nearby devices over their Wi-Fi or Bluetooth interface (*i.e.*, "tethering"), opportunistic

connection sharing should reduce the number of overall cellular connection requests, thereby reducing request congestion and connection failures. Using trace-driven simulations, we show that connection sharing can reduce connection failures by more than 95% when employed by a small number of devices in each cell sector. Although much work has been done on opportunistic connection sharing to address issues such as mobility, energy use, and incentives [2, 42, 67], no prior work has demonstrated the significant benefit that such connection sharing can achieve based on real-life cellular network data.

The rest of this chapter is organized as follows. In Section 3.2, we present details of the data collection process. Section 3.3 presents the characterization of performance issues during the crowded events and Section 3.4 presents various aspects of user traffic patterns to study the underlying causes of performance issues. We conduct trace-driven simulations to evaluate radio network parameter tuning and opportunistic connection sharing in Section 3.5. Section 3.6 reviews related work and the work is concluded in Section 3.7.

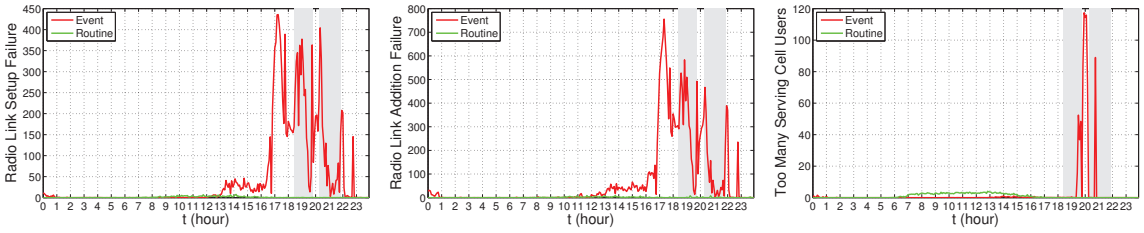
3.2 Data Set

The data set used in this study contains anonymized logs collected from RAN and CN of a tier-1 cellular network in the United States serving over 100 million customers. Our data set consists of two separate collections, each covering a metropolitan area during a high-profile event in 2012. The collections include information from hundreds of thousands of users and thousands of cell locations over multiple days including the event days. The first event, referred to as Event A hereafter, is a sporting event that consists of two segments of activities separated by an intermission. The second event, referred to as Event B hereafter, is a conference event that consists of multiple segments of activities separated by intermissions of varying lengths. In terms of publicly available attendance statistics, event A is roughly twice the size of event B. The activity segments in both events are illustrated by gray bars in all timeseries figures presented in this work. Furthermore, it is noteworthy that free Wi-Fi

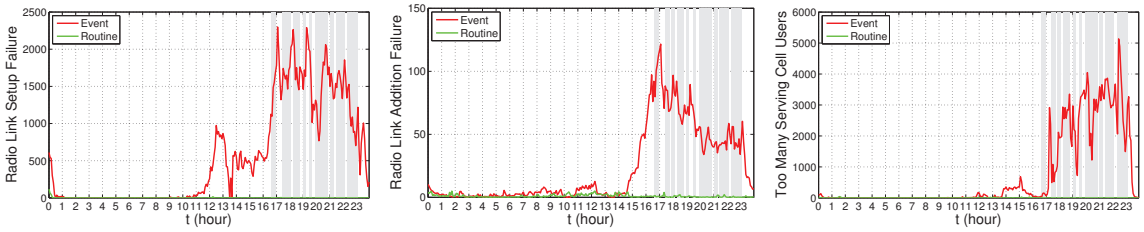
service was provided to all users during both of the events to offload as much cellular network traffic as possible. However, we do not have measurements on the network traffic that was offloaded to these Wi-Fi services; thus, we acknowledge that our results may be biased by this offloading.

The anonymized logs collected at an RNC in RAN contain throughput and RRC protocol request/response information. Using RRC requests from UEs and responses from the RNC, the RAB status of all UEs can be monitored. The anonymized logs collected from the CN contain TCP header information of IP flows carried in PDP context tunnels. They are collected from the Gn interfaces between SGSNs and GGSNs in the core network. They contain timestamp, per-flow traffic volume, content publisher, RTT computed during TCP handshake [49], and estimated packet loss ratio for each TCP flow aggregated in 5 minute bins. All device and user identifiers (*e.g.*, IMSI, IMEI) are anonymized to protect privacy without affecting the usefulness of our analysis. The data set does not permit the reversal of the anonymization or re-identification of users. We note that logs collected at RNCs encompass both voice and data traffic, whereas logs collected from the CN contain only data traffic information.

Next, we characterize performance issues during the aforementioned two high-profile events in Section 3.3. To study the underlying causes of the identified performance issues, we then correlate network performance with various aspects of user traffic patterns in Section 3.4. Throughout, we present results of the event day in relation to a routine day for baseline comparison. We normalize the actual measurement values by their mean values on the routine day (unless stated otherwise); our results thus effectively represent how the event differs from routine conditions. We omit absolute numbers from some non-normalized plots due to proprietary reasons.

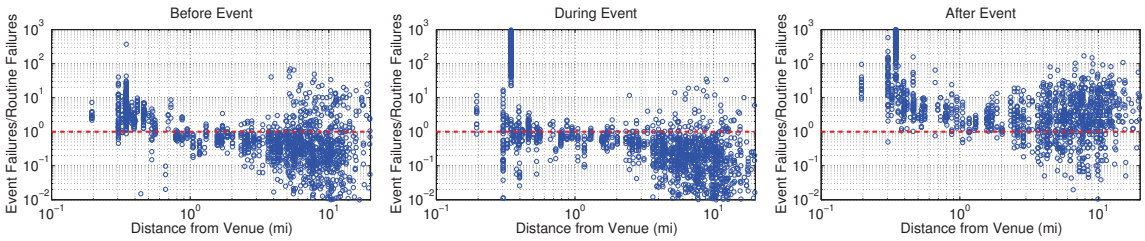


(a) Event A

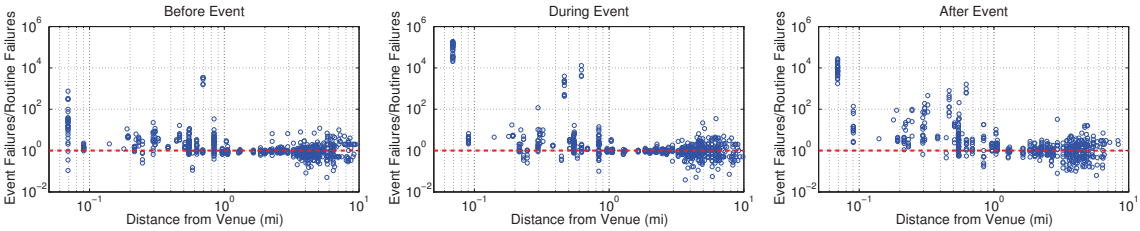


(b) Event B

Figure 3.1. (Normalized) Timeseries of common types of RRC failures



(a) Event A



(b) Event B

Figure 3.2. RRC failure ratios plotted as a function of distance to the venue and for time intervals before, during, and after the event

3.3 Characterizing Performance Issues

Generally speaking, a user's experience about network performance can be divided into two phases. The *pre-connection phase* is characterized by the UE attempting to establish a

connection with the cellular network, or in other words establishing a RAB. In this phase, the user waits for connection establishment, while not being able to exchange traffic at this time. The *post-connection phase* starts after a RAB is assigned. In this phase, user experience is related to more traditional voice call performance metrics such as call drop and block rate or end-to-end TCP performance metrics, such as delay and packet loss. Below, we separately discuss both pre- and post-connection network performance experienced by users during both events.

3.3.1 Pre-connection Network Performance

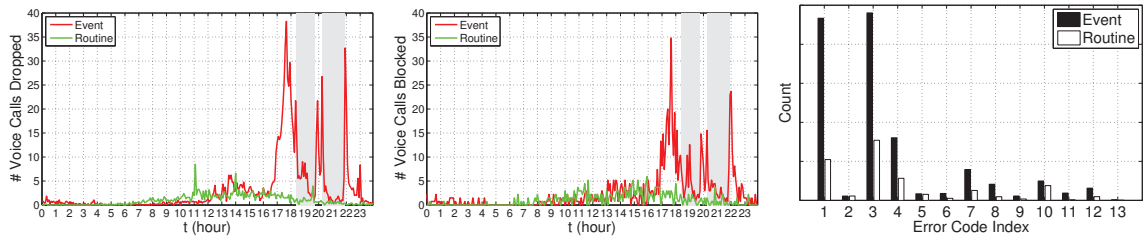
Users may experience difficulty in establishing RABs in the pre-connection phase due to a wide variety of reasons. Every time a request to allocate more radio resources is denied by the RNC, a RRC failure and its underlying reason is logged by our measurement apparatus. In our analysis, we study the logs of various types of RRC failures that are collected at the RNC. Each type of RRC failure corresponds to a specific problem in the cellular network operation. The 3 most common types of failures observed in our data set are the following.

1. *Radio link setup failures* occur when a user's request to setup a radio link is not served due to poor RF channel quality, which is often caused by increased interference.

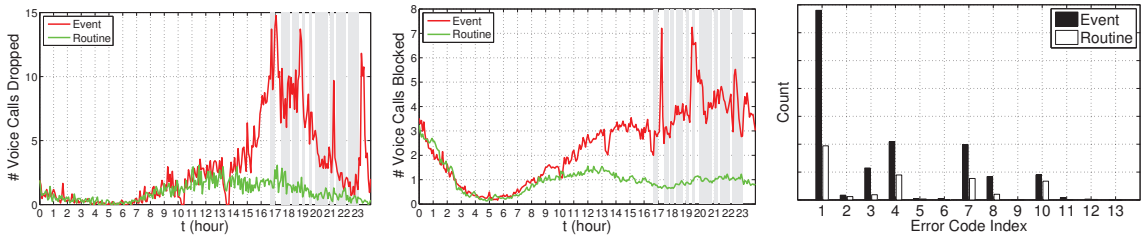
2. *Radio link addition failures* occur when a user's request to add a radio link to an existing radio connection for soft handovers is denied.

3. *Too many serving cell users* indicates blocking for new users which results when all available RABs are occupied by existing users.

Figure 3.1 plots the timeseries of the most common types of RRC failures on the event and routine days for both events. We observe that RRC failures increase sharply on the event days, whereas they are negligible (and steady) on the routine days for both events. For both events, RRC failures start occurring around noon and generally reach their peak either just before or during the event. Specifically, *radio link addition failures* peak at more than 700x



(a) Event A



(b) Event B

Figure 3.3. (Normalized) Voice performance measurements

their average on the routine day for event A and *too many serving cell users* peak at more than 5000x their average on the routine day for event B.

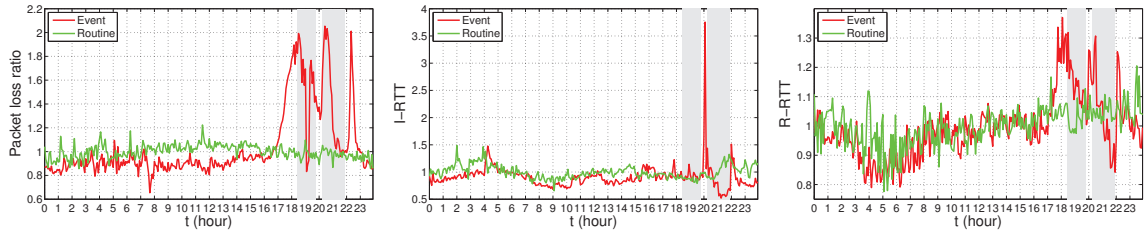
The nature of RRC failures for both events indicates that their potential root cause is high network load and congestion due to a large number of competing users at cell sector level. Therefore, we next analyze RRC failures at cell sector level before, during, and after the events as a function of distance from the venue. Figure 3.2 shows the scatter plots between the distance of cell sectors from the venue (in miles) and the ratio of the number of RRC failures on the event day to that on the routine day. The horizontal dashed line at $y = 1$ is a reference for the data points where RRC failures on the event and routine days are equal. So the data points above the reference line represent cell sectors that have more RRC failures on the event day than the routine day. Likewise, the data points below the reference line represent cell sectors that have less RRC failures on the event day than the routine day. Both x- and y-axes are converted to logarithmic scale for the sake of clarity. Note that there are many cell sectors equidistant from the venue, especially those cell sectors that are close to the venue. These cell sectors are mounted at the same cell tower but face different directions, and have different tilt angles and frequencies.

Overall, we observe that cell sectors closer to the venue have 2-3 orders of magnitude more RRC failures on the event day than the routine day. The RRC failure ratios progressively decrease as the distance of cell sectors to the venue increases. For both events, we observe interesting dynamics across the scatter plots for time intervals before, during, and after the event. For event A, we observe that the failure ratios generally increase by 2-3 orders of magnitude for cell sectors less than half a mile from the venue throughout the event day. In contrast, for the cell sectors that are far from the venue, their failure ratios drop during the event and jump by 1-2 orders of magnitude after the event finishes. The aforementioned observations can be linked to the sporting nature of event A, where people swarm the venue before and during the event, creating a void in surrounding areas. The post-event jump in the failure ratio is likely correlated with most people leaving the venue and using their devices to share their experience with others via voice calls or social network posts (we show later in this section that the observed user activity supports this hypothesis). We observe similar trends for event B as well; however, the post-event jump in the failure ratio is clearly visible only for cells within 1 mile of the venue. For these reasons, while characterizing user network traffic in the next section, we focus our attention on the cell sectors that are within 1 mile radius of the venues for both events.

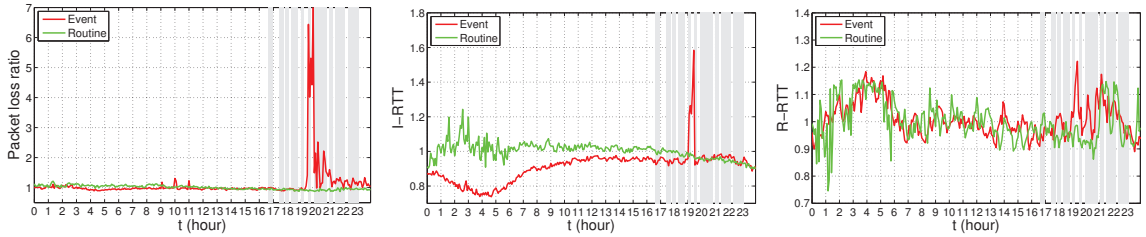
Pre-connection failures (especially those pertaining radio link addition and indicating too many serving cell users) peak by a factor of 700 (for event A) and 5000 (for event B) relative to their average on the routine days. These failures increase by 2-3 orders of magnitude in cell sectors very close to the venues before and during the events, but only increase in cell sectors further away after the venues.

3.3.2 Post-connection Network Performance

As discussed in Section 3.2, during the RAB setup phase, the RNC verifies that the needed radio resource for the request actually exists before it assigns a RAB. In other words, if a device has successfully acquired a RAB for communication, its performance should theoreti-



(a) Event A



(b) Event B

Figure 3.4. (Normalized) Data performance measurements

cally remain acceptable per operator's configuration even if the overall network demand level exceeds network capacity. This is because excessive demand requests will get blocked off by the RNC from acquiring any RAB. However, network conditions can quickly change even for UEs that have already acquired a RAB because of factors such as interference, mobility, *etc.* Such dynamic network conditions can force UEs to request a change in current RAB status, initiating a series of RRC failures which could in turn result in degraded voice and data performance. Below, we separately discuss voice and data performance.

To quantify voice performance, we study voice call drop and block rates for both events in Figure 3.3. Similar to our observations about pre-connection network performance, the number of voice call drops and blocks increase substantially on the event days as compared to the routine days for both events. Specifically, we observe peak increases of more than 30x and 7x relative to their average on the routine days for events A and B, respectively. It is noteworthy that voice call drop and block rates peak just before the start of the events, during the intermissions, and at the end of the events. This observation is consistent with our expectation that users are less likely to make voice calls during event activities and more likely to make voice calls either before the start of events, after the end of events, or during

Table 3.1. Description of voice call error codes

Index	Category	Description
1	Unspecified	All cases which do not map to the ones described below
2	Radio Connection Supervision	Radio Link Control (RLC) unrecoverable
3	Radio Connection Supervision	Maximum number of RLC retransmissions
4	Radio Connection Supervision	Expiry of timer
5	Radio Connection Supervision	Radio link failure indication
6	Operations & Management	Cell lock indication
7	Soft Handover	No active set addition update
8	Soft Handover	No active set deletion update
9	Soft Handover	No active set replacement update
10	Soft Handover	Cell not in the neighbor set
11	Soft Handover	High speed-downlink shared channel cell change failure
12	Inter-Frequency Handover	Inter-frequency handover failure
13	Channel Switching	Transition to DCH state not completed

intermissions between event activities. To further investigate the root causes of voice call blocks and drops, we also plot the histogram of their error codes in Figure 3.3. The error code descriptions in Table 3.1 indicate that the two most common categories of error codes for both events are related to radio connection supervision and soft handovers, which in turn point to interference and mobility as the root cause.

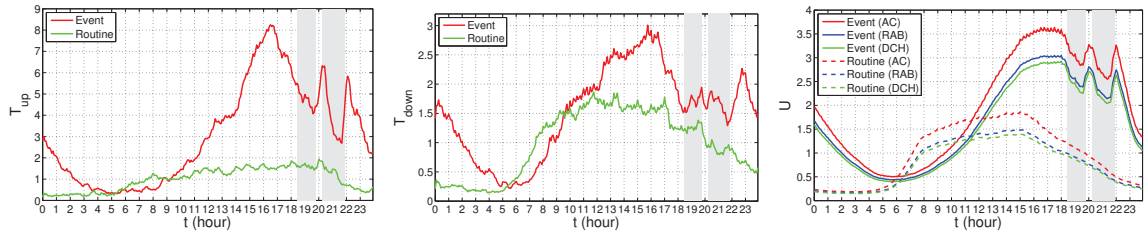
To quantify data performance, we study two key end-to-end TCP performance metrics: packet loss ratio and RTT for both events.[1] Packet loss ratio quantifies network reliability. We only have packet loss ratio measurements for TCP flows, which constitute approximately 95% of all flows in our data set. RTT quantifies network delay and is defined as the duration of time taken by a packet to reach the server from the UE plus the duration of time taken by a packet to reach the UE from the server. It is important to note that RTT measurements are biased by differences in the paths between different UEs and the external servers they communicate with. Similar to packet loss ratio measurements, we only have RTT measure-

[1]Because end-to-end TCP performance also involves additional parameters such as back-haul bandwidth and even remote server load, we leave a more detailed investigation of TCP performance to future work.

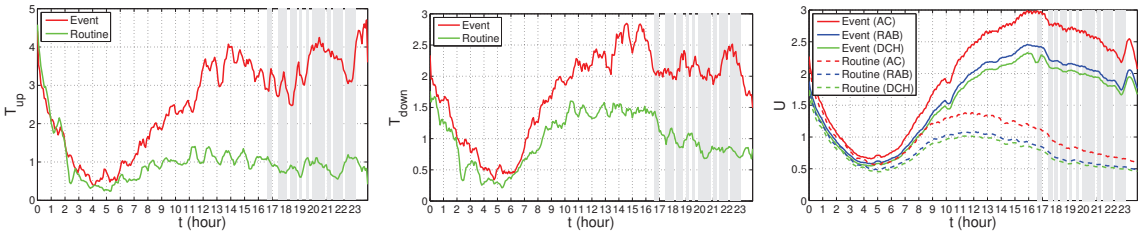
ments for TCP flows. RTT measurements for TCP flows are estimated by SYN, SYN-ACK, and ACK packets in the TCP handshake. In a cellular network, RTT essentially consists of two components: radio network RTT and Internet RTT. Radio network RTT (R-RTT) is the time duration between the SYN-ACK packet from server passing the Gn interface and the ACK packet from the UE passing the Gn interface. Internet RTT (I-RTT) is the time duration between the SYN packet from the UE passing the Gn interface and the SYN-ACK packet from the server passing the Gn interface. Thus, RTT = R-RTT + I-RTT.

Figure 3.4 shows the timeseries plots of packet loss ratio, Internet RTT, and radio network RTT for both events. Packet loss ratio peaks at 2x and 7x relative to its average on the routine days for events A and B, respectively. We observe different trends for radio network RTT and Internet RTT for both events. There is only a minor increase in radio network RTT on the event days. Internet RTT increases during the intermissions for both events; however, this increase indicates congestion at remote servers caused by increased event-driven traffic. Overall, data performance results indicate that users experience data connection performance issues to varying extents during the two events.

Post-connection performance degradation is observed for both voice and data network during the events. Specifically, voice call failures (dropped calls and call blocks) increase by a factor of as much as 30 (for event A) and 7 (for event B). Moreover, packet loss ratio increases by a factor of 2 (for event A) and 7 (for event B); while the RTT increases by a factor of 3.5 (for event A) and 1.5 (for event B). While these indicate a degradation in performance experienced by users already connected to the network, this is substantially smaller than the pre-connection failures discussed in Section 3.3.1. Overall, pre- and post-connection network performance results highlight that limited radio resources are the major bottleneck during crowded events.



(a) Event A



(b) Event B

Figure 3.5. (Normalized) Network load measurements

3.4 Understanding Performance Issues

Next, we characterize user network traffic to identify patterns that correlate with the observed pre- and post-connection performance degradation during the events. Using the insights obtained from this characterization, we aim to identify network optimization opportunities that can potentially improve end-user experience in crowded locations. We characterize network traffic in terms of both aggregate network load and user-level session characteristics.

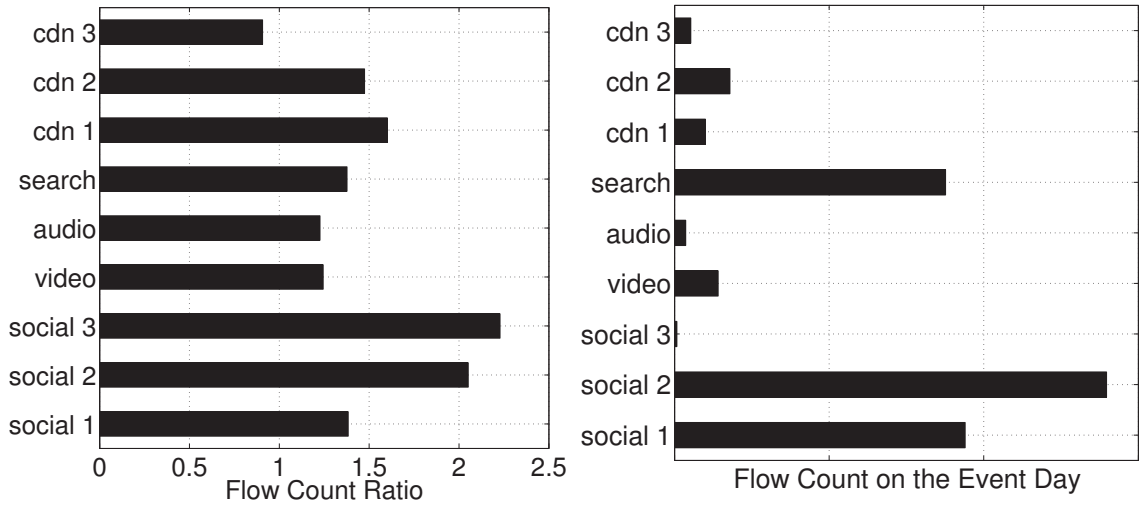
3.4.1 Aggregate Network Load

We quantify aggregate network load in terms of the following two metrics: throughput and user counters. Throughput or bit-rate is sampled for all UEs at the RNC every couple of seconds. Based on the direction of traffic, we can split the throughput into *uplink throughput* (T_{up}) and *downlink throughput* (T_{down}). Figure 3.5 plots the timeseries of uplink and downlink throughput on the event and routine days for both events. For the routine days, both uplink and downlink throughput peak around the noon time and decline steadily afterwards, reaching the bottom during late night and early morning. We observe a different

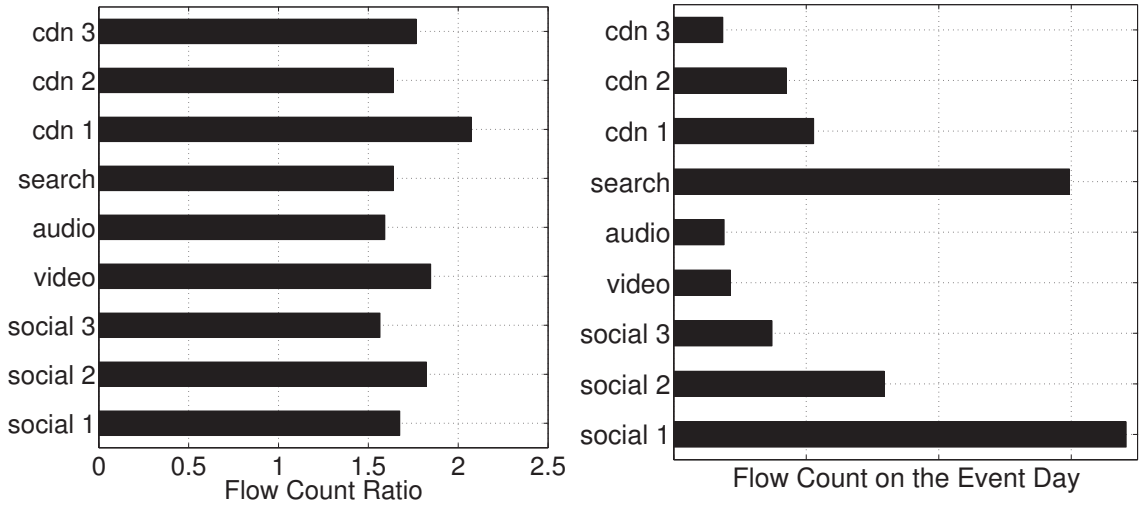
trend for uplink throughput on the event days. For instance, the peak uplink throughput on the event day is more than 8x and 4x the average throughput on the routine day for events A and B, respectively. We also observe that the uplink throughput peaks and event activities are approximately aligned. For instance, uplink throughput sharply increases at the start and end of the second segment for event A. Similar, though less pronounced, patterns are also observable for event B. In contrast to the uplink throughput, increases in the downlink throughput timeseries are steadier for both events.

To further analyze traffic volume characteristics, we plot the traffic flow count histograms for top content publishers in Figure 3.6. We focus on flows rather than bytes to avoid bias towards high volume applications, such as video streaming. We observe that flow counts of social networking content publishers more than double on the event day as compared to the routine day for event A. Likewise, social networking content accounts for most flows on the event day for event B. Our further investigation (not shown here) revealed that social networking content is at least 2x more upstream heavy as compared to other content types, which explains the increase in uplink throughput during both events.

We also analyze user counters for the event and routine days for both events. Users are classified into the following overlapping categories based on their RRC states: *admission control* (AC), *radio access bearer* (RAB), and *dedicated channel* (DCH). AC category includes the users who have completed the admission control procedure. RAB category includes the users who have been assigned a RAB after admission control. Such users are typically in either FACH or DCH state. Finally, DCH category only includes the users who are in DCH state. Let U denote the number of users, also let U_{AC}, U_{RAB}, and U_{DCH} denote the number of users in the aforementioned categories. As a general rule, $U_{AC} \geq U_{RAB} \geq U_{DCH}$. Figure 3.5 plots the timeseries of number of users in AC, RAB, and DCH categories. These timeseries show a trend similar to the throughput measurements. All user counters have higher values on the events days as compared to the respective routine days for both events. Specifically, the number of users with admission control peaks at more than 3x during the



(a) Event A

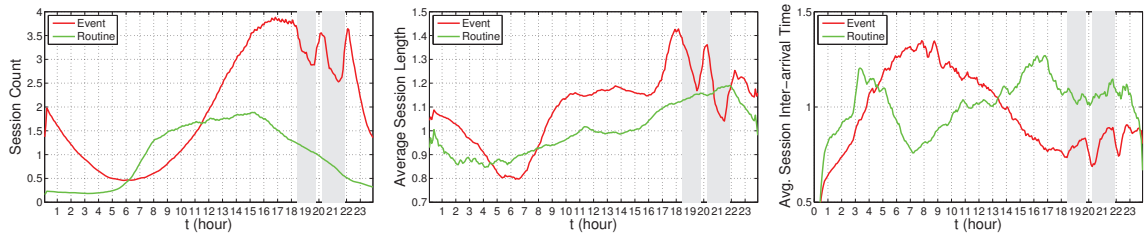


(b) Event B

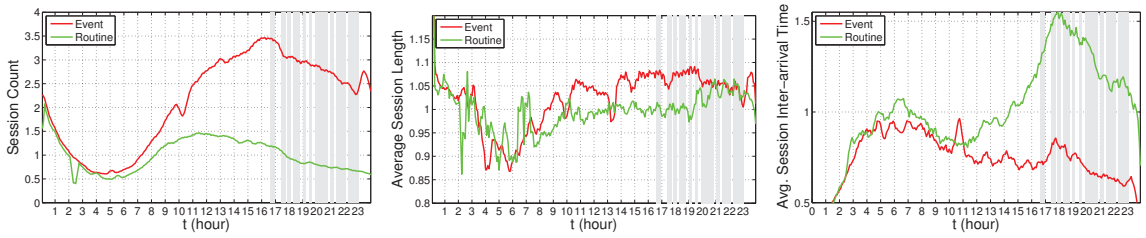
Figure 3.6. Flow count histograms for top content publishers in our data set

events as compared to its average on the routine days.

Both aggregate uplink and downlink throughput increase during the event days; uplink throughput increases by a factor of as much as 8 and 4 (for events A and B respectively), while downlink throughput increases by a factor of 3 (for both events). Moreover, there is a substantial increase in the traffic volume of social networking content during the events, which is relatively more upstream heavy. Likewise, number of users with admission control

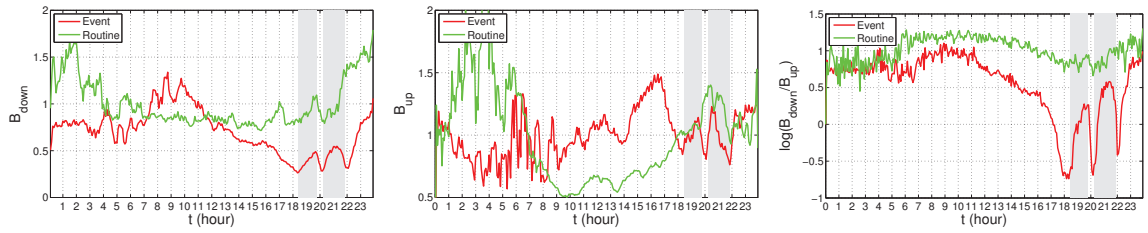


(a) Event A

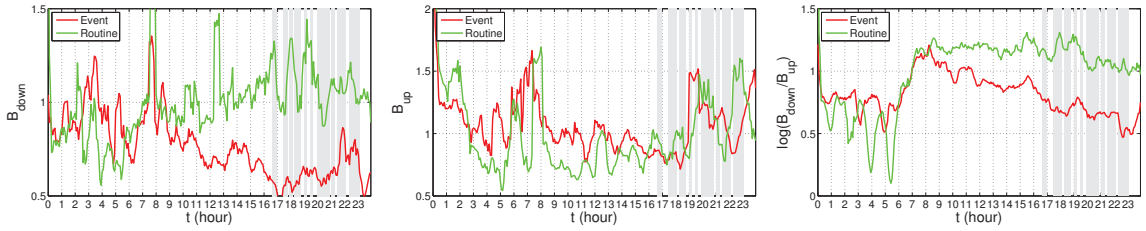


(b) Event B

Figure 3.7. (Normalized) Session Count, Average Length, Average Inter-arrival Time



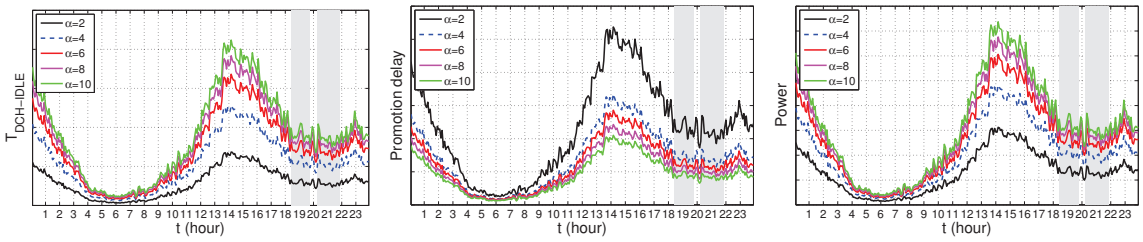
(a) Event A



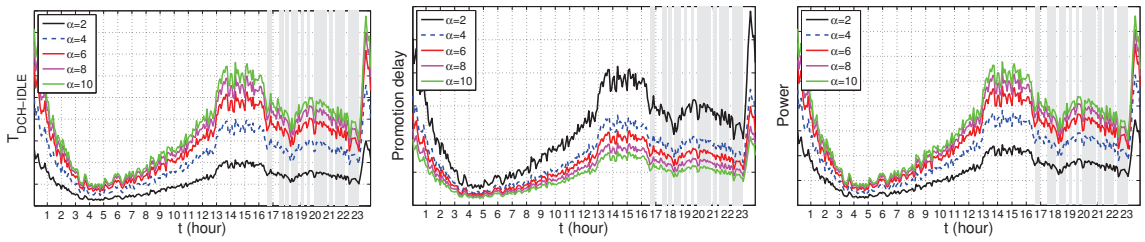
(b) Event B

Figure 3.8. (Normalized) Per-Session Downlink Bytes (B_{down}), Uplink Bytes (B_{up}), Ratio ($B_{down}/(B_{up}$)

increase by a factor of 3 for both event days. Overall, our aggregate network load characterization shows that increased user activity during the events, specifically in terms of uplink throughput and user counters, is correlated with increased pre-connection failures. To reduce the impact of increased network load during crowded events, we will investigate



(a) Event A



(b) Event B

Figure 3.9. Experimental results for radio network parameter tuning

the effectiveness of opportunistic connection sharing in Section 3.5.2.

3.4.2 User-level Sessions

We now analyze characteristics of user-level traffic sessions for both events. A session consists of consecutive time intervals with uplink or downlink byte transfer and its end is marked by an inactivity timeout of τ seconds. The results presented in this section are computed for $\tau = 5$ seconds. Changing the value of τ does not qualitatively affect the analysis results. Figure 3.7 shows the timeseries of session count, average session length, and average session inter-arrival time for both events. Session count follows a similar trend to the earlier aggregate network load metrics – at peak, there is more than 3.5x increase relative to the average on the routine days for both events. Furthermore, we observe an increase in average session length on the event days as compared to the routine days, *e.g.*, there is more than 1.4x increase for event A. On the contrary, average session inter-arrival time decreases sharply on the event days as compared to the routine days – this indicates that users are initiating sessions much more frequently during the events. To further investigate the nature of changing session patterns,

we plot the timeseries of average downlink bytes per session (B_{down}), average uplink bytes per session (B_{up}), and the average ratio of downlink bytes to uplink bytes per session in Figure 3.8. We observe that average downlink bytes per session sharply decreases up to 0.5x during the event days; whereas, average uplink bytes per session exhibits a mixed trend. The ratio (B_{down}/B_{up}) also sharply decreases during the events, which is due to the increased traffic volume of upstream-heavy social networking content.

User sessions are on average longer during both events (by a factor of as much as 1.4) – as well as more numerous and more frequently initiated. However, users exchange only as much as half the bytes per session on average. This change in workload is due to a change in the application usage during these events, such as greater proportion of social networking flows observed earlier. These trends point to potential waste of radio resources by UEs, which can be mitigated by tuning radio network parameters. Towards this end, we will investigate the effectiveness of varying RRC timeouts in Section 3.5.1.

3.5 Evaluating Mitigation Schemes

In this section, we evaluate two proposals to mitigate cellular network performance degradation during crowded events.

3.5.1 Radio Network Parameter Tuning

We first investigate whether tuning radio network parameters can result in more efficient radio resource usage during crowded events. As mentioned in Section 3.2, UEs acquire and release radio resources by transitioning to different RRC states. A UE is promoted to a higher energy state depending on buffer occupancy and it is demoted to a lower energy state depending on timeouts. Here, we study how RRC timeouts can be tuned for more efficient radio resource utilization, without explicit feedback from individual UEs. Recall from Figures 3.7 and 3.8 that average bytes per session decreases during the events, despite the increase

in average session length. This observation highlights potential waste of radio resources and UE energy consumption in crowded locations. Therefore, a natural suggestion would be to reduce RRC timeouts to mitigate the radio resource wastage. However, reducing RRC timeouts can result in more frequent state promotions, which can introduce state promotion delays resulting in degraded user experience [59, 80]. Hence, there is a tradeoff between performance and resource efficiency.

We conduct trace-driven simulations to quantitatively study the tradeoffs involved in changing RRC timeouts. We simulate the RRC state machine of every user using the RNC logs while focusing on the DCH state, which has the highest allocated radio resources and energy consumption among all RRC states. Specifically, we study DCH→FACH RRC timeout parameter, which is denoted by α hereafter. As mentioned earlier, changing RRC timeouts introduces tradeoffs among radio resource wastage, user experience, and UE energy consumption. We use the following three performance metrics to quantify these factors. (1) The DCH state idle occupation time, denoted by $T_{\text{DCH-IDLE}}$, quantifies the radio resources wasted by UEs in DCH state. (2) The promotion delay quantifies the additional delay caused when UEs transition to DCH state from FACH state. (3) The power consumption quantifies the total energy consumed by UEs during DCH state occupation and in FACH to DCH transitions. We use the following simulation parameters in our experiments (inferred by Qian *et al.* in [80]): (1) FACH→DCH promotion radio power = 700mW, (2) DCH state power = 800mW, (3) FACH→DCH promotion delay = 2 sec, and (4) RLC buffer threshold = 500 bytes.

Similar to the evaluation of opportunistic connection sharing, we evaluate radio network parameter tuning for a subset of cell sectors that are within 1 mile radius of the venues. We conduct trace-driven simulations of individual users' RRC state machines on this subset for both event and routine days. Figure 3.9 shows the timeseries plots of the aforementioned three performance metrics for varying α values. We observe that the DCH state idle occupation time and UE energy consumption increase for larger α values. On the other hand,

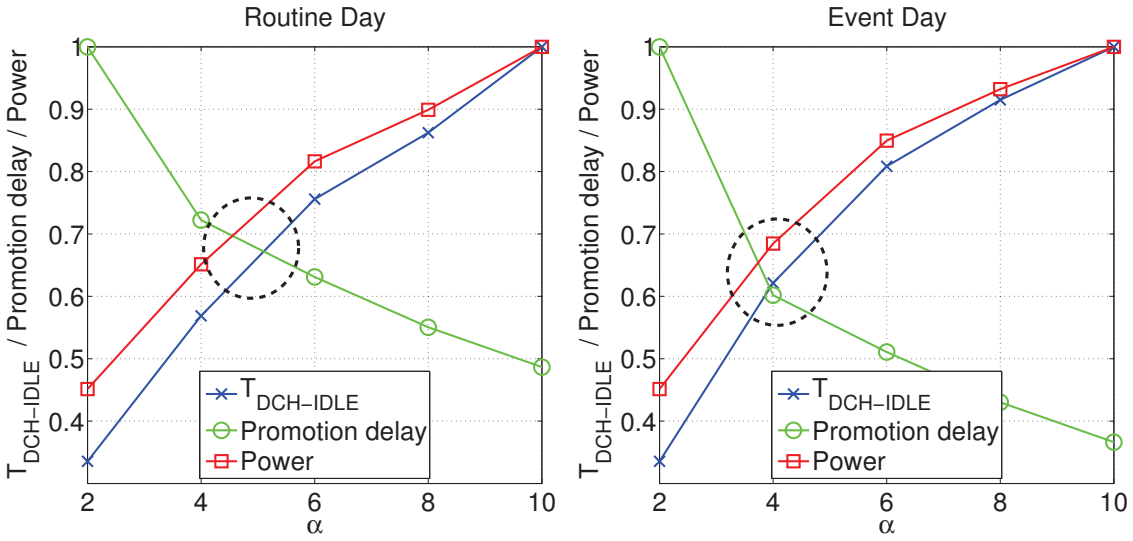
promotion delay decreases for larger α values. These observations indicate that decreasing the RRC timeout values reduces the waste of scarce DCH channels and UE energy consumption. However, this benefit is achieved at the cost of increased promotion delay that may degrade user experience, especially for applications that are not delay-tolerant.

To systematically study the tradeoffs between these performance metrics on the event days and compare them to routine days, we plot them as a function of α. Figure 3.10 plots the max-normalized average of the performance metrics as a function of α for the event and routine days. In theory, we want to select a value of α which simultaneously minimizes the values of all performance metrics. In this case, the crossover points (highlighted by black circles in Figure 3.10) and their corresponding α values represent suitable performance tradeoff. We find that these crossover points shift to smaller α values – by 1-2 seconds – on the event days as compared to the routine days. In practice, however, α is typically set to achieve a target delay or resource overhead. In this case, as observable from Figure 3.10, we can tune α to smaller values to achieve the same targets and achieve strictly better performance during crowded events.

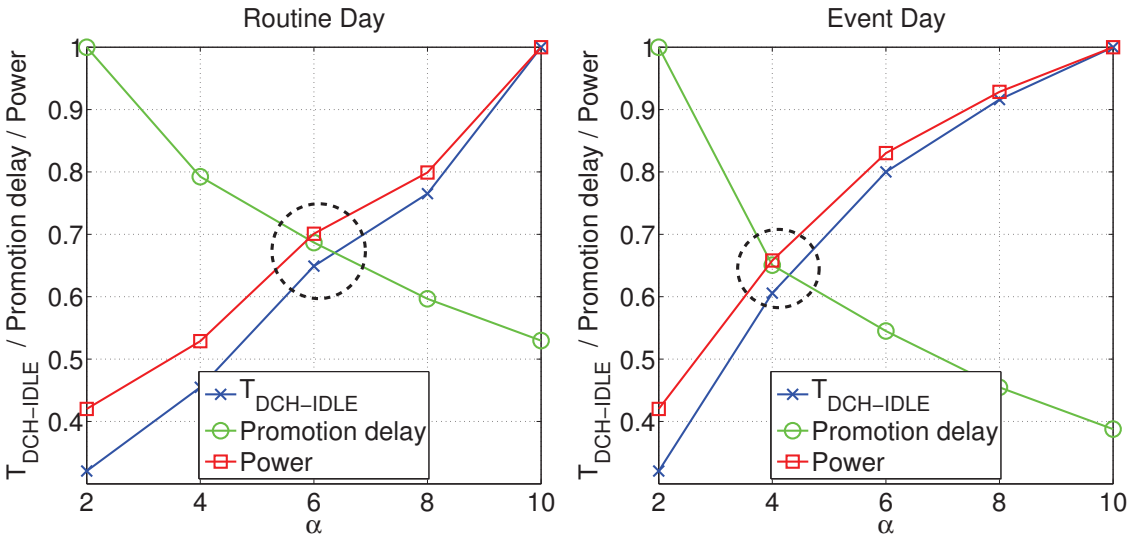
3.5.2 Opportunistic Connection Sharing

We now evaluate a simple opportunistic connection sharing scheme to reduce the network load at individual cell sectors for eradicating RRC failures observed in Section 3.3.1. The basic idea is that users can share their connection to NodeBs with other users to reduce the overall network load in terms of occupied radio channels. In this scheme, a selected set of UEs act as Wi-Fi hotspots for other UEs in their vicinity. Therefore, other UEs, instead of wastefully establishing separate connections, can connect to NodeBs via the UEs acting as Wi-Fi hotspots. Using this approach, we aim to reduce the number of UEs that are directly connected to NodeBs to free up channels, although the overall throughput carried by the network remains the same.

To evaluate the potential benefit of the opportunistic connection sharing scheme, we con-



(a) Event A



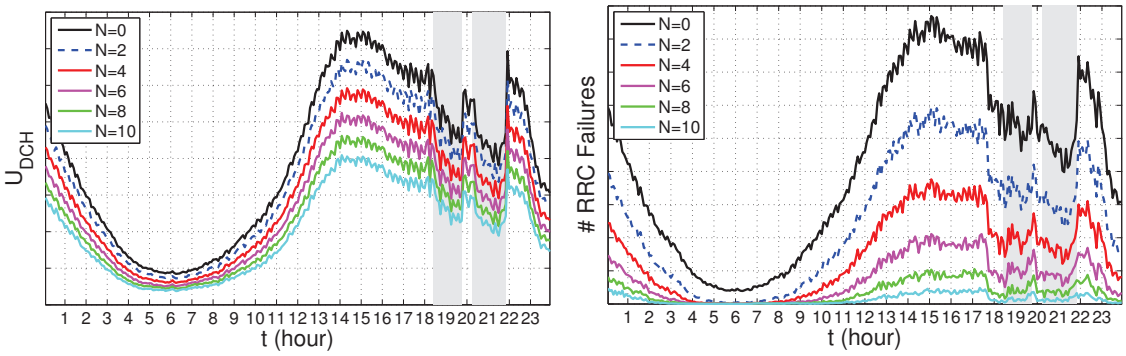
(b) Event B

Figure 3.10. Tradeoff between performance metrics for varying RRC timeout (α) values. Y-axis is max-normalized for each metric. α values corresponding to black circles achieve better performance tradeoff.

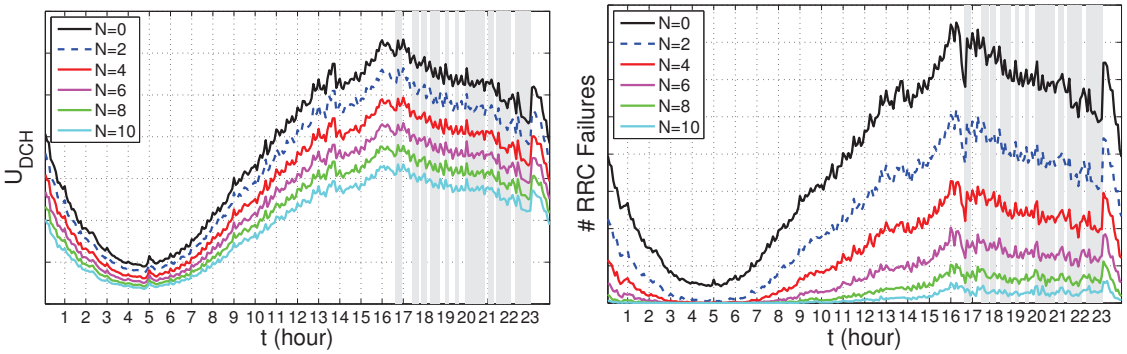
duct cell sector level trace-driven simulations. With respect to the mobility of the users, we assume that the users are static within 1 minute time bins. This is a reasonable assumption for crowded events in stadiums, auditoriums, and conference rooms. We do not have

fine-grained location information of users in our data set; therefore, we have to simulate the locations of users. In this work, we aim to generate the locations of users in a grid-like scenario – similar to how people are typically seated in stadiums and conferences. Towards this end, we use Complete Spatial Randomness (CSR) point generation model with hard-core inhibition [20]. CSR with hard-core inhibition does not allow neighbors within a pre-defined radius around the randomly generated points, resulting in a grid-like setting. The points in the realizations denote the locations of users in our simulations. In our simulations, Wi-Fi hotspots are selected randomly because we do not have access to other relevant information, such as battery life and signal strength, that may be used to optimize this selection. Once a UE connects to a Wi-Fi hotspot, it is disconnected after 1 minute of inactivity. The locations of inactive users are updated using the above CSR model. In our simulations, the cell sectors are set to have $2,250,000$ ft^2 coverage area, the inhibition radius is set to 2 ft, the Wi-Fi range of users is simulated as $\mathcal{N} \sim$ (200 ft, 20 ft), and the upper limit on the number of simultaneous connections for each Wi-Fi hotspot is set to 5. The cell sector coverage area is in typical range for crowded urban locations, the inhibition radius is set to be reasonably large, and the Wi-Fi range and the maximum number of simultaneous connections are conservatively set. We assess the benefit of the connection sharing scheme in terms of the following metrics: the number of users in DCH state (U_{DCH}) and the number of RRC failures.

Since we are primarily interested in deploying this scheme in congested locations, we focus our evaluations on a subset of cell sectors in our data set that are within 1 mile radius of the venues. We evaluate the opportunistic connection sharing scheme using trace-driven simulations on this subset on the event days. The results plotted in Figure 3.11 are the average of 1000 independent simulation runs. We plot the timeseries of the number of occupied DCH channels (U_{DCH}) for varying number of Wi-Fi hotspots per cell sector (denoted by N). As expected, we observe that U_{DCH} values become smaller for larger values of N, freeing up DCH channels that are now available for UEs unable to transition to the



(a) Event A



(b) Event B

Figure 3.11. Experimental results for opportunistic connection sharing

DCH state due to RRC failures. We also plot the number of RRC failures for varying values of N in Figure 3.11. Again, as expected, we observe that RRC failures decrease for increasing values of N. Consequently, based on instantaneous load conditions, the cellular network can dynamically vary the required number of users acting as Wi-Fi hotspots to minimize RRC failures. We note that this connection sharing scheme successfully eradicates more than 95% RRC failures for both events when $N = 10$. This substantial reduction in the number of RRC failures in congested cell sectors will likely result in improved performance for users.

Below, we discuss some practical issues of opportunistic connection sharing.

• *Wi-Fi Hotspot Selection*: The selection of Wi-Fi hotspots can be mediated by the cellular network based on a variety of factors, such as battery life and signal strength. UEs acting as Wi-Fi hotspots may experience high energy drain and may run out of battery power.

To address this issue, the role of Wi-Fi hotspot can be periodically rotated among the user pool by the cellular network. The cellular network should prefer UEs with better signal strength because UEs consume significantly more energy and suffer reduced effective bit rate when the signal strength is poor [87]. On the other hand, the UEs that are unable to get RAB assignments can discover Wi-Fi hotspots in their range using the standard Wi-Fi discovery methods. In case of multiple options, UEs should prefer hotspots with better signal strength.

• *Initial Connection Delay*: After a device connects to a Wi-Fi hotspot, similar to RRC protocol, it disconnects after a pre-defined inactivity timer expires. However, the value of this timer should be set much higher than the corresponding RRC timers so that the device does not have to incur initial delay, which is up to several seconds, for every data transfer. In our simulations, the inactivity timer was set to be 1 minute.

• *Out of Range*: A device has to request RAB assignment when it moves out of a hotspot's Wi-Fi range. If it is unable to get a RAB due to congestion then the RNC can dynamically assign more Wi-Fi hotspots in the cell sector to provide connectivity to more users.

• *Radio Technologies*: Opportunistic connection sharing is only usable when a majority of devices in the cellular network have built-in Wi-Fi capability. In our simulations, we assume that all devices have Wi-Fi capability. In case Wi-Fi is not available, other technologies such as Bluetooth can also be used. Bluetooth has lower power consumption, smaller radio range, and supports less data rate as compared to Wi-Fi. Consequently, it can be used as a low power alternative for small transmissions such as tweets.

• *Wi-Fi-Cellular Handovers*: Working extensions to the Wi-Fi standard already address the issue of smooth handovers between Wi-Fi and cellular networks, including 3GPP Access Network Discovery and Selection Function (ANDSF), Hotspot 2.0 initiative [76], and other techniques [4].

• *Voice Traffic Offloading*: In this opportunistic connection sharing scheme, voice traffic can be tunneled via the Wi-Fi connection using the well-known Voice over Wi-Fi solutions, such

as Wi-Fi certified Voice-Enterprise [12].

• *Incentives*: Cellular network operators may provide billing based incentives to users for participating in this opportunistic connection sharing scheme.

3.5.3 Limitations

Below, we briefly mention two limitations of our trace-driven simulation evaluations. First, our simulation based evaluations cannot account for changes in traffic workload resulting from different network conditions due to our proposed mitigation schemes. Second, they also cannot account for low-level dependencies between performance metrics and network load. For example, some types of RRC failures are impacted by interference, which in turn is a function of network load. Addressing these limitations requires experiments on operational cellular networks, which are beyond the scope of this work. However, despite these limitations, we believe that the sheer magnitude of the improvements observed in our simulations indicates that the mitigation schemes discussed in this work would accrue some benefit in practice.

3.6 Related Work

We divide related work into the following categories.

Cellular Performance Characterization: The areas of cellular performance characterization have recently received much attention by the research community. For example, small-scale studies have characterized application performance [44, 114] and fairness [14]. Large-scale studies have characterized throughput and airtime [78], smartphone traffic [94], M2M device traffic [91], smartphone app traffic [92,116], and heavy users [25]. In contrast to these studies, we believe that we are the first to analyze cellular performance changes specifically during crowded events.

Radio Network Parameter Tuning: Prior work on radio network parameter tuning study

the impact of RRC timers on network performance and smartphone energy consumption. Most prior work is based on user-end measurements performed using a few cellular devices. For instance, Liu *et al.* characterized performance in a 1xEV-DO network using measurements obtained from two laptops equipped with Sierra Wireless data cards [65]. Balasubramanian *et al.* proposed a UE based approach, called TailEnder, to alter traffic patterns based on the prior knowledge of RRC state machine [17]. Some studies are based on theoretical analysis and simulation. For instance, Liers *et al.* proposed a scheme to adaptively tune RRC timeout parameters based on the demand and load situation, and validated it using simulations [64]. Yeh *et al.* proposed a scheme to tune RRC timeout parameters using analytical models based on available radio resources, energy consumption, quality of service, and processing overheads of the radio access network [118]. Qian *et al.* conducted trace-driven RRC state machine simulations using network-end measurements to investigate the optimality of RRC timeout parameters [80]. Furthermore, they proposed an application-aware tail optimization protocol to simultaneously optimize radio and energy resources [79]. Similar to the prior work by Qian *et al.* [79, 80], our analysis of radio network parameter tuning is based on trace-driven RRC state machine simulations. However, we focus on network-end tuning of RRC timeouts without any cooperation from UEs.

Opportunistic Connection Sharing: We build on existing work on opportunistic traffic offloading [42, 67]. Luo *et al.* proposed a unified architecture, where mobile clients use both 3G cellular link and Wi-Fi based peer-to-peer links for routing packets via peer-to-peer links to the appropriate destinations [67]. Han *et al.* proposed content-specific opportunistic communication scheme to offload cellular traffic via Wi-Fi or Bluetooth [42]. However, neither of these proposals were evaluated using real-world traces, and both approaches require architectural changes to network protocols and hardware. Our work complements these proposals by showing that their simplest and most practical instantiation — a simple one-hop connection sharing scheme that does not require architectural changes — can be very effective in real-life crowded events. To the best of our knowledge, this work is the first to

evaluate practical connection sharing techniques on real-world traces.

3.7 Conclusion

This work presents the first performance characterization of an operational cellular network during crowded events. We make three key contributions in this study based on the real-world voice and data traces that we collected from a tier-1 cellular network in the United States during two high-profile crowded events in 2012. First, we measured how cellular network performance degrades during crowded events as compared to routine days. Second, we analyzed what causes the observed performance degradation. Third, we evaluated how practical mitigation schemes for the observed performance degradation would perform in real-life crowded events using trace-driven simulations. Our findings from this study are crucial for cellular design, management, and optimization during crowded events.

4 Breaching Privacy in Encrypted Instant Messaging Networks

4.1 Introduction

The proliferation of online social networks has attracted the interest of computer scientists to mine the available social network data for developing behavior profiles of people. These profiles are often used for targeted marketing [62, 117, 119], web personalization [77], and even price discrimination on e-commerce portals [70, 75]. Recently, there has been increased interest in more fine-grained profiling by leveraging information about people's friendship networks. It has been shown that information from people's friendship networks can be used to infer their preferences and religious beliefs, and political affiliations [18, 43, 72, 120].

There has been a lot of research on de-anonymizing people's friendship networks in online social networks such as Facebook, MySpace, Twitter [29, 63]. Surprisingly, little prior work has focused on de-anonymizing people's friendship link in instant messaging (IM) networks. IM services – such as Yahoo! Messenger, Skype, IRC, and ICQ – are popular tools to privately communicate with friends and family over the Internet. IM networks are different than other online social networks in various respects. For example, in contrast to online social networks, communication among users in IM networks is synchronous in nature and messages between two communicating users are routed through relay servers of the IM service provider.

The goal of this work is to identify the set of most likely IM users that a given user is communicating with during a fixed time period. Note that packet payloads in IM traffic are encrypted; therefore, payload information cannot be used for the identification. Therefore, to infer who a user is talking to, we will rely only on the information in packet header traces. Packet header traces contain information such as timestamp, source IP address, destination IP address, source port, destination port, and protocol type, and payload size of each packet. It is noteworthy that each packet in the IM traffic has as its source and destination IP addresses of a user computer and an IM relay server (or vice versa). At no point do two users exchange packets directly with each other, *i.e.*, there are no packets in which the two communicating users' IP addresses appear in the same packet. For this attack, we assume that IM service acts neutral, *i.e.*, it neither facilitates the attacker nor actively participates in providing anonymity to the users using non-standard functionality. Our specific goal is to accurately identify a candidate set of top-k users with whom a given user possibly talked to using only the information available in packet header traces.

A natural approach to tackle this problem would be to match header information of packets entering and leaving IM relay servers. However, simply matching header information of packets entering and leaving IM servers is not feasible due the following reasons. First, a user may be talking to multiple users simultaneously. Second, IM relay servers typically serve thousands of users at a time. Third, the handling of duplicate packets that are the result of packet losses followed by re-transmissions. Forth, the handling of out-of-order packets. Finally, the handling of variable transmission delays, which are introduced by the IM relay servers.

In this work, we propose a wavelet-based scheme, called COmmunication Link De-anonymization (COLD), to accurately infer who's talking to whom using only the information available in packet header traces. Wavelet transform is a standard method for simultaneous time-frequency analysis and helps to correlate the temporal information in one-way (*i.e.* user-to-server or server-to-user) traffic logs across multiple time scales [68]. Wavelet anal-

ysis allows decomposition of traffic time series between a user and an IM relay server into several levels. All levels are associated with a coefficient value and contain different levels of frequency information starting from low to high. The original traffic time series can be reconstructed by combining all levels after weighing them with their respective coefficients. COLD leverages the multi-scale examination of traffic time series provided by wavelet analysis to overcome the aforementioned technical challenges. Given two candidate time series between an IM relay server and two users, COLD computes correlation between the vectors of wavelet coefficients for both time series to determine whether these users talked to each other.

We evaluate the effectiveness of COLD on a Yahoo! Messenger data set comprising of traffic collected over 10, 20, 30, 40, 50 and 60 minute periods. We also compare COLD's performance to a baseline time series correlation (TSC) scheme, which represents the state of the art. The effectiveness is quantified in terms of hit rate for a fix-sized candidate set. The results of our experiments show that COLD achieves a hit rate of more than 90% for a candidate set size of 10. For slightly larger candidate set size of 20, COLD achieves almost 100% hit rate. In contrast, a baseline method using time series correlation could only achieve less than 5% hit rate for similar candidate set sizes.

We summarize the major contributions of this work as follows.

1. We define an attack for breaching communication privacy in encrypted IM networks using only the information available in packet header traces.

2. We propose COLD to to infer who's talking to whom using wavelet based multi-scale analysis.

3. We conducted experiments using a real-world Yahoo! Messenger data set to evaluate the effectiveness of our proposed approach.

The rest of this chapter is organized as follows. Section 4.2 summarizes the related work. A detailed description of attack scenarios is provided in Section 4.3. Section 4.4 provides

details of the proposed attack. In Section 4.5, we present the evaluation results on a real-world Yahoo! Messenger data set. Possible evasion techniques and their countermeasures are discussed in Section 4.6. Finally, Section 4.7 concludes the chapter.

4.2 Related Work

In this section, we provide details of the research work related to our study. To the best of our knowledge, no prior work has reported a successful attack to breach users' communication privacy in encrypted IM networks using only the information available in packet header traces. However, there is some relevant work in the area of mix network de-anonymization. We discuss it and other related studies below.

4.2.1 Mix Network De-anonymization

In the area of mix network, several studies have used correlation techniques for de-anonymization. However, most of these studies are limited to computing temporal correlation between traffic of two user-network links to find user-user links. Furthermore, de-anonymization of mix networks is fundamentally different from our problem in the following two aspects. First, mix network de-anonymization techniques require traffic logs from multiple points inside a mix network. In contrast, this study treats IM relay servers as a black box. Second, the size of user populations in mix network de-anonymization studies is only of the order of tens or hundreds. However, in real-life IM networks, thousands of users can simultaneously communicate with other users; therefore, presenting a more challenging problem. In [108], Troncoso and Danezis build a Markov Chain Monte Carlo inference engine to calculate probabilities of who is talking to whom in a mix network using network traces. However, they log network traces from multiple points in a mix network and the maximum network size studied in their paper is limited to 10. In [122], Zhu *et al.* compute mutual information between aggregate inflow and outflow traffic statistics to decide if two users are

talking to each other in a mix network. Similar to this study, they also log traffic from the edges of a mix network. However, their proposed approach requires traffic logs for longer time durations. In this work, we compare the results of COLD and the method proposed by Zhu *et al.* [122].

4.2.2 Social Network De-anonymization

There is also some related work in the field of social network de-anonymization. Narayanan and Shamitkov developed an algorithm to utilize sparsity in high-dimensional data sets for de-anonymization [73]. Later they developed a user re-identification algorithm that operated on anonymized social network data sets [74]. Other related studies use group membership information to identify users in a social network [115, 120]. IM networks also fall under the broader category of online social networks; however, our problem and the nature of the data available to us is different from those tackled in the aforementioned papers. These studies focus on user identification using mainly topological information; whereas, we focus on link identification using dynamic user communication traffic.

4.3 Problem Description and Attack Scenarios

In this section, we first provide a summary of architectural details of IM services. We then provide the details of information available from traffic traces logged near IM relay servers. Finally, we describe two scenarios in which traffic can be logged for link de-anonymization.

4.3.1 IM Service Architecture

We first describe the architecture of a typical IM service. Consider the scenario depicted in Figure 4.1 where two users v_1 and v_2 are communicating with each other via an IM service. When v_1 sends a message to v_2, the source IP address in packets containing this message correspond to v_1 and the destination IP address correspond to the IM relay server. These

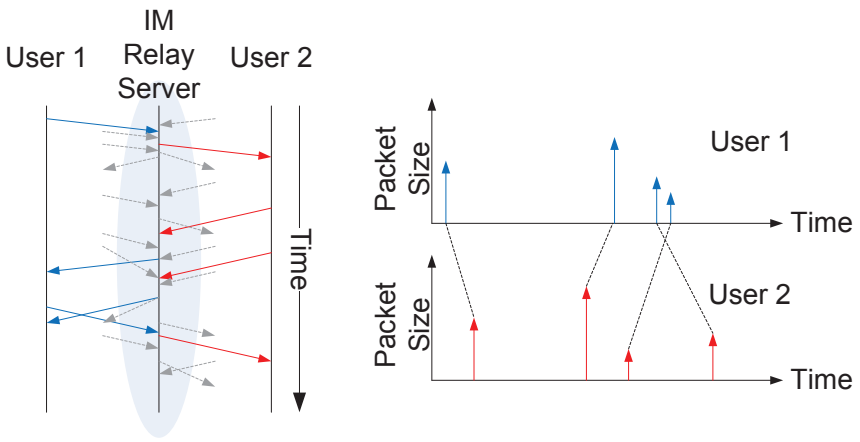


Figure 4.1. Transforming logged traffic traces to user traffic signals

packets are received by the IM relay server after a random delay. After receiving a packet from v_1, the IM server first looks up the IP address of v_2. It then creates new packets with its IP address as source and IP address of v_2 as destination. These packets containing message from v_1 are then relayed by the IM relay server to v_2 and have the same contents. This process incurs an additional delay after which the new packet reaches v_2.

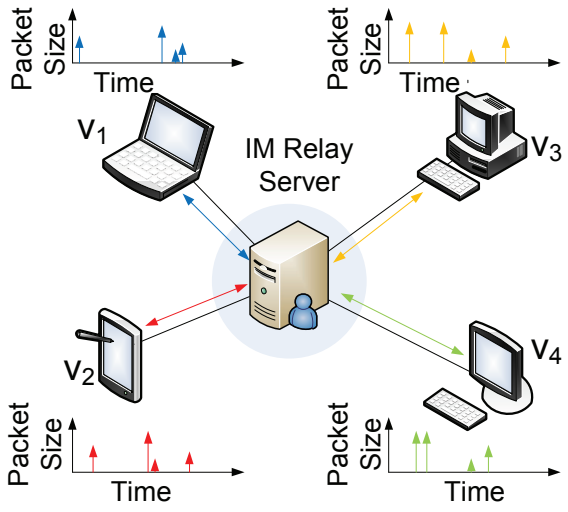
The network traffic logged near the IM relay server only contains header information because the packet payload contents are not useful due to encryption. The statistics recorded by the well-known traffic logging tools like Cisco's NetFlow include IP addresses, port numbers, protocol, packet size, and timestamp information [28]. As mentioned before, IP addresses are used to identify individual users of the IM service. IM traffic is filtered from rest of the traffic using a combination of protocol and port number information. We are left with only aggregated packet sizes and timestamp information for each flow. A logged entry for a flow is an aggregation of packets which may be sent to or received from the IM server. Due to aggregation, information about the direction of flow is lost for individual packets. Therefore, we make a realistic assumption that the direction information is not available in the logged traffic. An example of a similar publicly available data set is the Yahoo! Network Flows Data [7].

Referring to Figure 4.1, each flow in the data set comprises of information about incoming and outgoing packets between an IM relay server and a user. Furthermore, individual users can be distinguished based on IP addresses in the IM traffic. In Figure 4.1, traffic exchanged between v_1 and the IM relay server is represented by blue arrows and traffic exchanged between v_2 and the IM relay server is represented by red arrows. The timestamps and packet sizes are both discrete and in units of milliseconds. The packet sizes are typically recorded in bytes. The resulting signal for each flow is discrete in both time and amplitude as shown in Figure 4.1. These sparse time domain traces of network traffic are referred to as *traffic signals* from now-onwards. It is interesting to simultaneously analyze traffic signals for both users v_1 and v_2. Note that every entry in v_1's traffic signal has a time-shifted (time delayed or advanced) matching entry of equal magnitude in v_2's traffic signal. These matches between each pair of entries are marked by broken lines joining traffic signals in Figure 4.1. Matching entries across both traffic signals may not have the same order due to random end-to-end delays. For example, 3^{rd} message flow entry in v_2's trace appears as 4^{th} entry in v_1's trace in Figure 4.1.

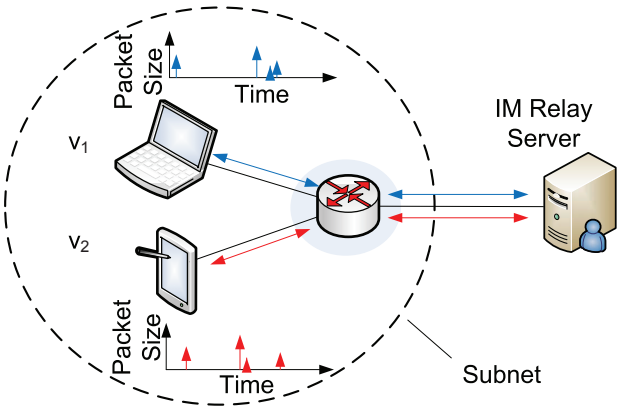
4.3.2 Attack Scenarios

We now consider two different scenarios in which traffic information necessary for the proposed attack can be obtained.

The first scenario assumes the capability to monitor incoming and outgoing traffic of an IM relay server or server farm. Figure 4.2(a) shows four users v_1, v_2, v_3 and v_4 connected to an IM relay server. The shaded circular region around the IM relay server marks the boundary across which network traffic is logged. For the scenario depicted in Figure 4.2(a), v_1 is communicating with v_2 and v_3 is communicating with v_4. Traffic signals for all users that are obtained after pre-processing their traffic flow logs. For each flow represented in a user's traffic signal, a corresponding flow entry can be observed in the traffic flow log. The IM relay servers also introduces a random delay between the time a message arrives at



(a) Collecting all incoming and outgoing traffic from IM relay server



(b) Collecting all incoming and outgoing traffic near border gateway routers of an organizational network

Figure 4.2. Two attack scenarios

the IM relay server and the time it is relayed to the other user. Therefore, there will be a mismatch in the timestamps of the occurrences of a message in communicating users' traffic signals.

The second scenario assumes that all IM users communicating with each other are located in the same network. Many organizations, such as universities, connect to external networks and the Internet through one or more gateway routers. The incoming and outgoing traffic

has to pass through a small number of gateway routers. In this scenario, it is possible to collect flow logs near the gateway routers of an organizational network. Figure 4.2(b) depicts the above-mentioned scenario. Here, v_1 and v_2 are in the same network and are communicating with each other via an IM relay server. All incoming and outgoing traffic of the network passes through the border gateway router near which it can be logged. The region near border gateway router is represented by the shaded region in Figure 4.2(b). The traffic signals obtained from pre-processing the flow logs have the same characteristics as described for the first scenario.

4.4 COLD: COmmunication Link De-anonymization

In this section, we present the details of our proposed method (COLD) to detect communication links between users in IM networks. We first introduce the overall architecture of COLD. We then provide details of each of its modules. Finally, we provide an easy-to-follow toy example of COLD on a small set of three IM users.

4.4.1 Architecture

As mentioned in Section 4.3, the logged traffic traces are separated for all users based on IP address information. These user-wise separated traffic traces are further pre-processed and converted to traffic signals. The traffic signals for all users are stored in a database. Note that traffic signals of users may span different time durations. To overcome this problem, we use zero-padding so that the lengths of traffic signals are consistent for all users. After this pre-processing, wavelet transform is separately applied to all users' traffic signals [68]. We then construct feature vectors for all users using the computed wavelet coefficients. Now, to compare two given users, we compute the correlation coefficient between their constructed feature vectors. Finally, the values of the correlation coefficient are sorted to generate the candidate set. The details of all modules of COLD are separately discussed below.

4.4.2 Details

After pre-processing the traffic traces, we compute the discrete wavelet transform (DWT) of each user's traffic signal. The wavelet transform enables us to conduct a simultaneous time-frequency analysis. A traffic signal is decomposed into multiple time series, each containing information at different scales that range from coarse to fine. A time series at a coarse scale represents the low frequency or low pass information regarding the original time series. Likewise, a time series at a fine scale represents the high frequency or high pass information regarding the original time series. This allows us to compare traffic patterns of users at multiple time scales.

We have to select an appropriate wavelet function for our given problem. Since we are processing traffic signals of a large number of users, we want to select an efficient wavelet type. For our study, we have chosen the Haar wavelet function for wavelet decomposition [66]. We have chosen the Haar wavelet function because it is simple and is computationally and memory-wise efficient. Furthermore, the wavelet transform can be applied for varying decomposition levels to capture varying levels of detail. Choosing the optimal number of decomposition levels is important because this may lead to suppressing relevant and critical information that might be contained in one or more levels of the wavelet decomposition. Below, we discuss the method to select the optimal number of decomposition levels.

Let $D \in \mathbb{Z}^+$ denote the optimal number of decomposition levels. Different methods have been proposed in the literature to select the optimal number of decomposition levels. In this work, we have used Coifman and Wickerhauser's well-known Shannon entropy-based method to select the optimal number of decomposition levels [30]. We applied this method to traffic signals of all users and then selected the optimal decomposition level at the 95th percentile. Now that we have selected the optimal number of decomposition levels, we can apply the wavelet transform on user traffic signals.

Once we have obtained the wavelet coefficients after applying the wavelet transform to a user's traffic signal, we need to convert them to a standard feature vector so that we can

compare users' signals. Let $\mathcal{F}_X$ denote the feature vector of a user X. The coefficients that contain high frequency information are more numerous and such coefficients are assigned lower weights. Similarly, the coefficients that contain low frequency information are fewer and are assigned higher weights. The time signal corresponding to level 1 of the wavelet decomposition represents the coarsest features containing low frequency information, and level D refers to the highest level describing the most detailed features containing high frequency information. The level D feature coefficients are assigned weight 1, the level $D-1$ coefficients are assigned weight 2, etc., and the level 1 coefficients are assigned weight 2^{D-1}. In general, the level d features are assigned a weight of 2^{D-d-1}. To produce the standard feature vector in which each coefficient is given the appropriate weight, we replace each coefficient by a vector of its copies of length equal to its weight, *i.e.* a wavelet coefficient of decomposition level d is replaced by a vector containing 2^{D-d-1} copies. This is equivalent to using the undecimated wavelet transform of users' traffic signals. By following this procedure, the total length of the feature vectors of all traffic signals becomes consistent.

After applying the wavelet transform and post-processing coefficients to a user X's traffic signal, we obtain a feature vector denoted $\mathcal{F}_X$. To compare the feature vectors $\mathcal{F}_X$ and $\mathcal{F}_Y$ for two users X and Y, we have to compute their correlation. The sample correlation coefficient $r_{X,Y}$ of two discrete signals $\mathcal{F}_X$ and $\mathcal{F}_Y$, both of length L, is defined as,

$$r_{X,Y} = \frac{\sum_{i=1}^{L}(\mathcal{F}_X(i) - \overline{\mathcal{F}_X})(\mathcal{F}_Y(i) - \overline{\mathcal{F}_Y})}{(L-1)s_X s_Y}. \tag{4.1}$$

Here, $\mathcal{F}_X(i)$ is the ith element of the feature vector $\mathcal{F}_X$, $\overline{\mathcal{F}_X}$ is the sample mean of its elements, and s_X is the sample standard deviation of its elements. The values of the correlation coefficient lie in the closed interval $[-1, 1]$. The correlation coefficient values close to zero indicate no correlation; whereas, the values close to 1 and -1 respectively highlight strong correlation and anti-correlation. For this study, we only consider the magnitude of the correlation coefficient and discard its sign. After computing the correlation coefficient

for all pairs of users, we get the upper triangular correlation matrix $\mathbf{R}_0$. $r_{i,j}$ is written into the ith row and the jth column of the correlation matrix $\mathbf{R}_0$. Conceptually, this correlation matrix is similar to the adjacency matrix of a weighted graph. We add to $\mathbf{R}_0$ its transpose to obtain $\mathbf{R}$

After obtaining the correlation matrix $\mathbf{R}$ whose elements are in the range of $[0, 1]$ we need to generate, for each node, a sorted list of nodes in decreasing order of probability of communicating. This is done by sorting the node indices in descending order of correlation coefficients in every column of $\mathbf{R}$. The resulting matrix will have the same size as $\mathbf{R}$ and is labeled $\mathbf{I}\downarrow$. Suppose that the S most likely users that are communicating with user i is required. Then the user IDs contained in the top S rows of the i-th column of $\mathbf{I}\downarrow$ is the sorted list of users i is most likely communicating with.

4.5 Experimental Results

In this section, we first describe the data set used for evaluating COLD, then define evaluation metrics, and finally present evaluation results.

4.5.1 Data Set

We collected a data set from Yahoo! Messenger IM network to validate our proposed approach. To keep the volume of logged data manageable, the users of Yahoo! Messenger were filtered by geographic location and restricted to the New York City area. This data set consists of traffic logs of Yahoo! Messenger user activity over a period of 60 minutes from the greater New York area, between 8 a.m. to 9 a.m. Using this data set, we create six data sets that are the subsets of the entire data. These consist of data over the only the first 10, 20, 30, 40, 50 and 60 minutes, *i.e.* from $8-8:10$ a.m., $8-8:20$ a.m., $8-8:30$ a.m., $8-8:40$ a.m., $8-8:50$ a.m. and $8-9$ a.m. To gauge the effect of the duration over which a data set is collected we evaluated our proposed COLD scheme on all six data sets. Table

Table 4.1. Data set statistics

Time	Duration	Users	Messages	Sessions
8-8:10a	10 mins	3,420	15,370	1,968
8-8:20a	20 mins	5,405	33,192	3,265
8-8:30a	30 mins	7,438	53,649	4,661
8-8:40a	40 mins	9,513	75,810	6,179
8-8:50a	50 mins	11,684	99,721	7,669
8-9a	60 mins	13,953	126,694	9,264

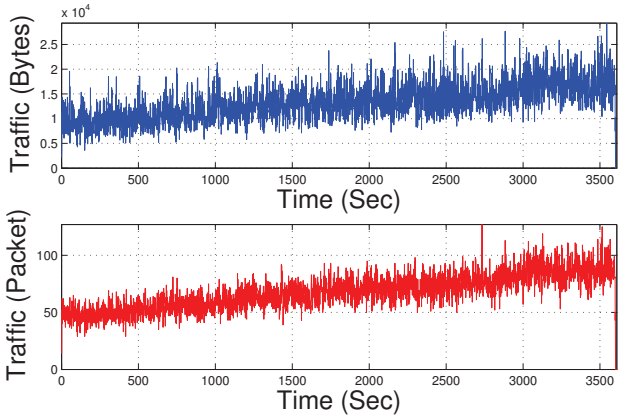


Figure 4.3. Time series plot of traffic volume, in bytes and number of packets, over the entire 60 minute time period from $8-9$ a.m.

4.1 lists, along with the time of day and duration, the number of logged users, number of messages exchanged between them, and the number of instant messaging sessions included in each data set.

The collected data is divided into two parts: input data and ground truth data, to systematically evaluate our proposed approach. Both data sets were collected with the assistance of Yahoo! and are described in the following text.

The input data consists of *user-to-server* traffic traces that were collected similar to the scenario described in Figure 4.2(a). Figure 4.3 plots the volume of traffic logged in these traffic traces. The figure on top plots number of bytes per second against time. Similarly, the plot in the bottom figure plots the traffic volume in packets per second for the same period of time.

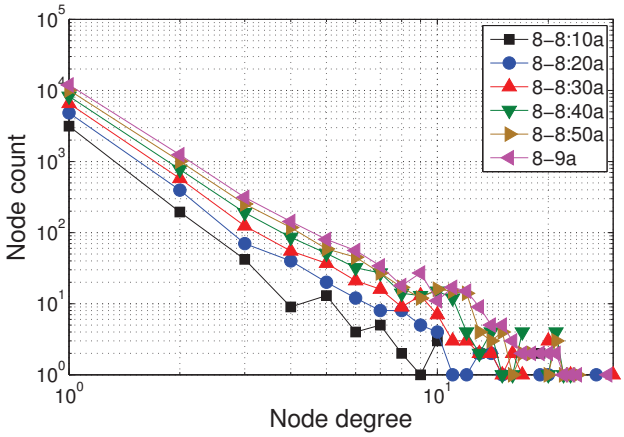


Figure 4.4. Node degree distribution in our Yahoo! Messenger data set.

The verification data contains a record of the actual *user-to-user* connections resulting from conversations between users. Therefore, the verification data contains the ground truth for given problem. Our proposed COLD scheme attempts to recreate the link structures between users contained in the verification data by only using information in the input data. Figure 4.4 is a plot of the degree distribution of users observed in the verification data collected over 10 and 60 minute time periods. The distribution is approximately linear on log-log scale over the range of degrees from 1 to 9 for the 10 minute data, and from 1 to 11 for the 60 minute data.

4.5.2 Evaluation Metrics

Let V denote the set of Yahoo! Messenger users $v_1, v_2, \ldots, v_N$. Furthermore, let E denote the set of actual communication links $u_1, u_2, \ldots, u_M$ of size M between N users captured in the verification data. Then $G(V, E)$ is the graph of users (or vertices) connected by the communication links (or edges) between them. Recall that the goal of the attack is to detect communication links $\widehat{U}$ that estimates the actual set of communication links in the verification data U. The graph $\widehat{G}(V, \widehat{U})$ is the outcome of the scheme that constitutes the attack. In the rest of this section, we compare our proposed COLD scheme with the

baseline time series correlation (denoted by TSC here onwards). A graph that is obtained using COLD will be denoted by $\widehat{G}_C(V, \widehat{U}_C)$. A graph obtained using TSC is denoted by $\widehat{G}_T(V, \widehat{U}_T)$.

Consider the subset of vertices with degree δ in a graph $\widehat{G}\left(V, \widehat{U}\right)$ obtained using either schemes. Now consider a candidate set C_i of size $S \geq \delta$ for every vertex v_i of degree δ. The candidate set C_i of a vertex v_i contains S vertices most likely to be v_i's neighbors, as determined by the COLD or TSC. We also define a neighborhood function denoted by $\Gamma_G()$. $\Gamma_G(v_i)$ returns the set of vertices in the graph G that are connected to vertex v_i. Furthermore, we define the node hit rate of a vertex v_i as the fraction of vertices in $\Gamma_G(v_i)$ that are also elements of candidate set C_i of size S. The node hit rate of vertex v_i is denoted $h_i(S)$ an is defined formally as follows.

$$h_i(S) = \frac{|\Gamma_G(v_i) \cap C_i(S)|}{|\Gamma_G(v_i)|} \tag{4.2}$$

The node hit rate can take values in the range of the closed interval $[0, 1]$. We also define the hit rate $H_{\widehat{G}}(S, \delta)$ for degree δ vertices of a graph $\widehat{G}\left(V, \widehat{U}\right)$ as the average of their node hit rates $h_i(S)$ when candidate set sizes are S.

$$H_{\widehat{G}}(S, \delta) = \frac{\sum_{i=1, \delta_i=\delta}^{N} h_i(S)}{n_d} \tag{4.3}$$

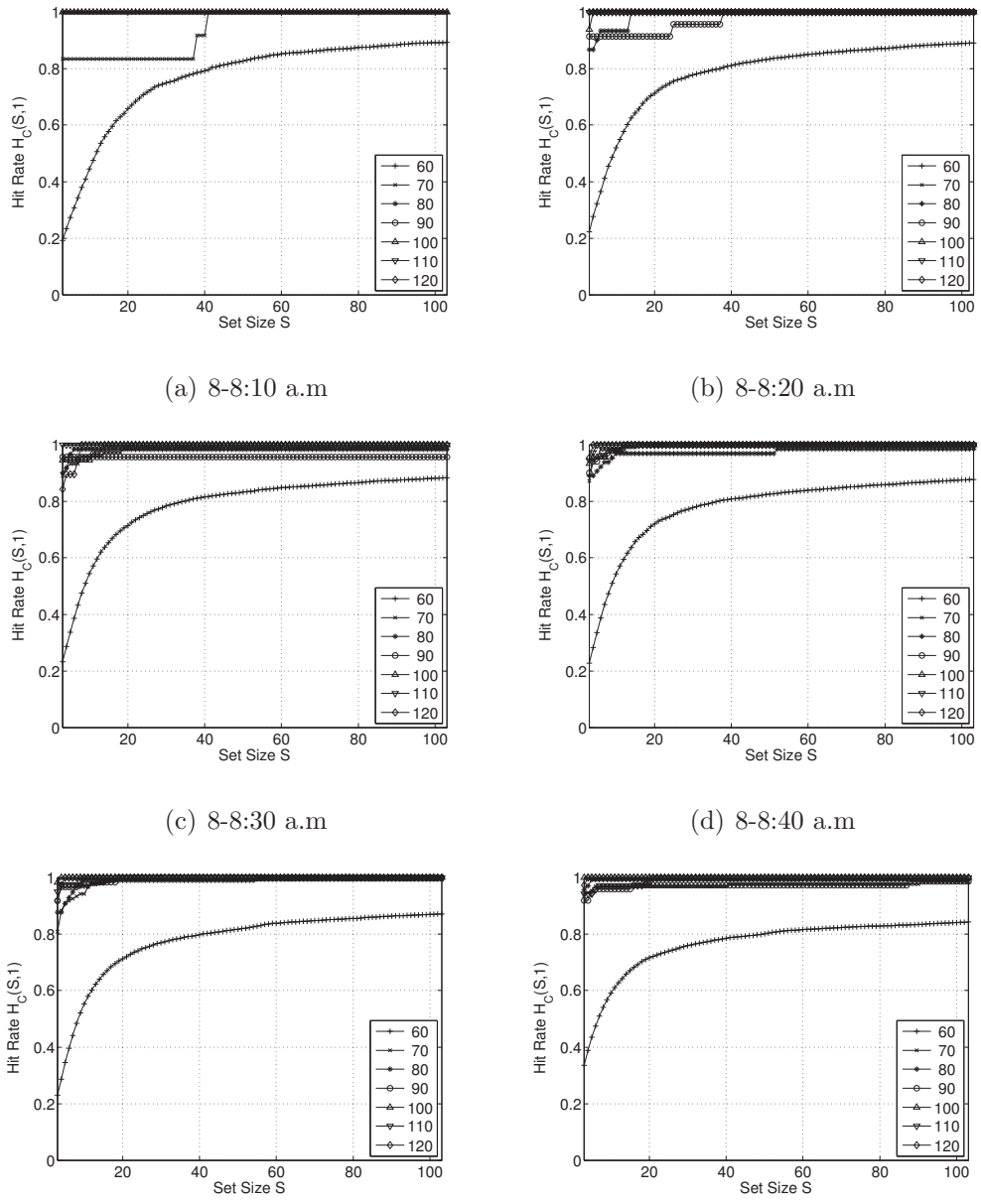
Here n_d is the number of vertices in $\widehat{G}$ of degree δ. Just like the node hit rate, the hit rate can take values in the range of the closed interval $[0, 1]$.

4.5.3 Results

We compute the hit rates achieved using COLD on the 10, 20, 30, 40, 50 and 60 minute data sets and compare them with the hit rates achieved by TSC. We further separate vertices by the number of packets they exchange over the duration of the data set, *i.e.* hit rates are computed separately for vertices exchanging $1 - 60$, $61 - 70$, $71 - 80$, $81 - 90$, $91 - 100$, $101 - 110$, and $111 - 120$ packets. As we observed in the degree distributions of nodes in

figure 4.4, data sets for all six durations are dominated by nodes of degree 1. Therefore, in our evaluation we focus primarily on degree 1 vertices. Figures 4.5(a), 4.5(b), 4.5(c), 4.5(d), 4.5(e), and 4.5(f) plot the hit rates of degree 1 vertices as a function of set size S for COLD on 10, 20, 30, 40, 50, and 60 minute data sets, respectively. Within each figure, hit rates are segregated according to the number of packets users send and receive over the duration the data was collected. As these six figures consistently show, the hit rate reaches between 0.9 and 1.0 for users exchanging 71 or more packets over the duration of the data sets. In case of the 20, 30, 40, 50, and 60 minute data sets in Figures 4.5(b), 4.5(c), 4.5(d), 4.5(e), and 4.5(f), this set of users is further extended to those exchanging 61 or more packets. In the 10 minute data set in figure 4.5(a) users with 61-70 packets in their trace have a high hit rate of more than 0.80. However, the candidate set size S has to be increased all the way to 40 for the hit rate to reach close to 1.0. For users exchanging between 1-60 packets the hit rate starts out between 0.20 and 0.40. As the candidate set size is increased from 1 upward, the hit rate rises at a very similar rate in all six data sets.

We compare the accuracy of our proposed approach to that of the time series correlation (TSC) method. Similarly, figures 4.6(a), 4.6(b), 4.6(c), 4.6(d), 4.6(e) and 4.6(f) plot the hit rates of degree 1 vertices as a function of set size S for TSC on 10, 20, 30, 40, 50 and 60 minute data sets, respectively. The baseline TSC method, which represents the state of the art, fails to deliver sufficient performance to be useful for any conceivable application, across all six data sets. With one slight exception, TSC fails to achieve a hit rate of even 0.20 even for candidate set size of as large as 100. The only exception is the group of users exchanging between 71-80 packets in the 10 minute data set. However, even for this subset of users, TSC provides a hit rate of less than 0.30 at a set size greater than 70, *i.e.* at best, for users messaging with only one other person, in a set of 70 candidates TSC will include the actual instant messaging partner with a probability of only 0.30.



(a) 8-8:10 a.m

(b) 8-8:20 a.m

(c) 8-8:30 a.m

(d) 8-8:40 a.m

(e) 8-8:50 a.m

(f) 8-9 a.m

Figure 4.5. Hit rates of COLD for vertices of degree 1 in the (a) 10 minute data set, (b) 20 minute data set, (c) 30 minute data set, (d) 40 minute data set, (e) 50 minute data set, and (f) 60 minute data set.

4.5.4 Discussions

These results provide us with several insights into the working of COLD. We separately discuss these insights in the following text.

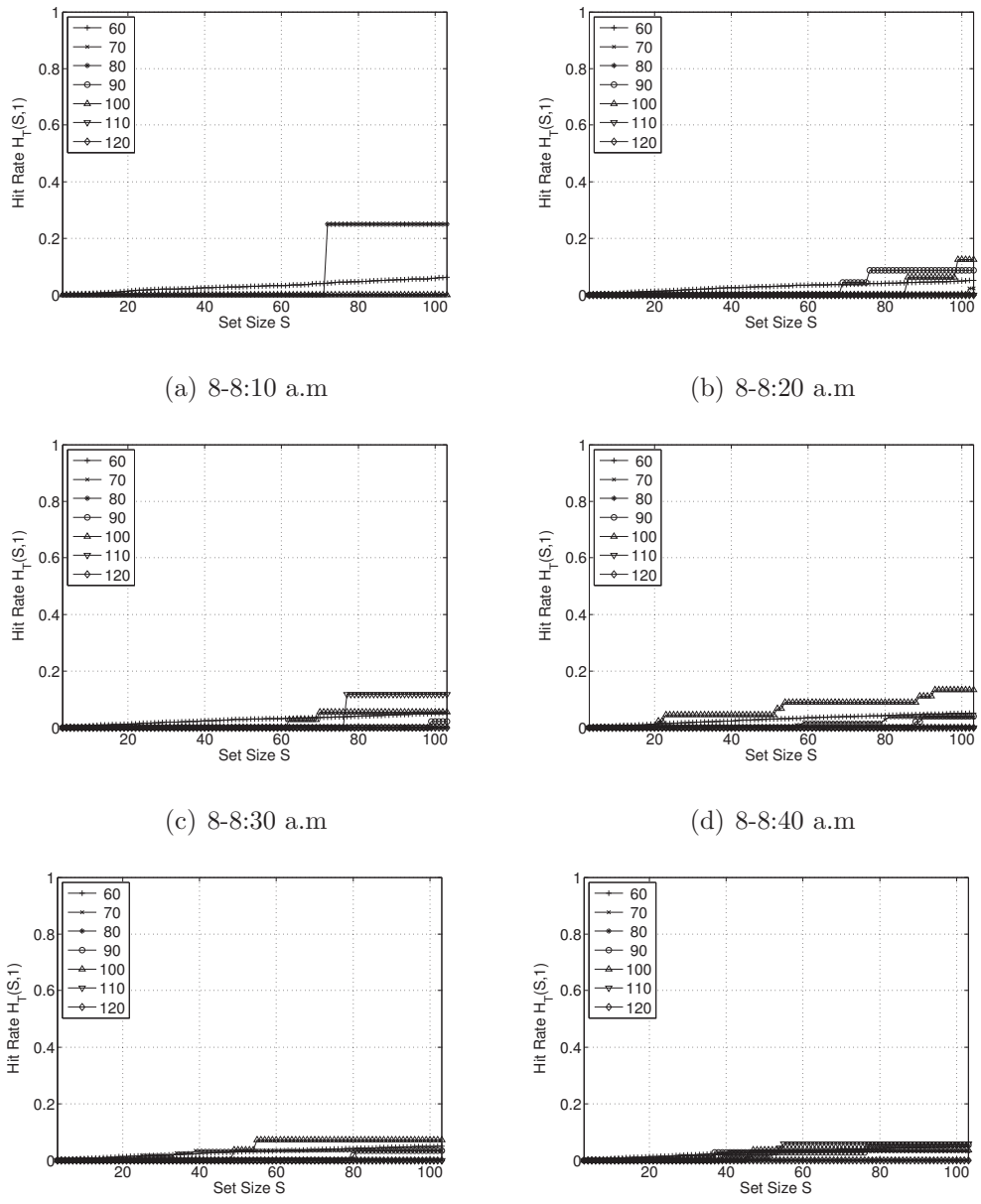
First, there appears to be a very clear threshold value for the number of recorded packets beyond which the de-anonymization attack using COLD yields high hit rates. From the plots in figure 4.5 we observe that the hit rate for users containing more than 60 packets in their traffic traces is significantly higher, above 90%, even at very small candidate set sizes. On the other hand, the hit rate of users containing 60 packets or less in their traffic trace is significantly lower. This threshold value holds across all six data sets of different durations. More packet entries in traffic traces provide more points to match two communicating users' traces with each other. The greater number of data points also reduces the probability of a false match. Therefore, it is easier to identify communicating users that message each other more frequently.

Second, the hit rate of users, classified by the traffic they generate, is largely independent of the time duration over which the traces were collected. Rather, it is the actual number of message packets exchanged during that period that determines the hit rate. Hit rates for users exchanging the same number of packets over different periods of time are very similar. Therefore, we can state that we can identify two communicating users using COLD with great certainty as soon as they exchange more than 60 message packets.

Third, while we have already stated that the time period over which traffic traces are collected have only a weak effect on the hit rate. However, looking at the hit rate functions of users with $61-70$, $71-80$ and $81-90$ packets in their traffic trace across different data sets, we observe that the hit rate function rises close to 1.0 at a faster rate in data sets collected over longer durations.

Fourth, judging by the time durations of the data sets (between 10-60 minutes), we conclude that the amount of data necessary to achieve a high hit rate by COLD can be collected in a relatively short period of time. Therefore, COLD does not require an extensive data collection effort to achieve high accuracy.

Finally, we observe that when TSC is applied to all data sets, the hit rate remains almost 0 for vertices of all traffic levels. This leads us to the conclusion that TSC is effectively unable



(a) 8-8:10 a.m

(b) 8-8:20 a.m

(c) 8-8:30 a.m

(d) 8-8:40 a.m

(e) 8-8:50 a.m

(f) 8-9 a.m

Figure 4.6. Hit rates of TSC for vertices of degree 1 in the (a) 10 minute data set, (b) 20 minute data set, (c) 30 minute data set, (d) 40 minute data set, (e) 50 minute data set, and (f) 60 minute data set.

to detect any communication links among users. We attribute this failure to the random phase delay of packet entries in traffic traces of two communicating users. These delays are a result of the bidirectional flow of traffic and jitter in the end-to-end delay.

4.6 Evasion and Countermeasures

This section presents some possible techniques that an adversary may utilize to evade the de-anonymization attack by COLD. We also discuss countermeasures to such evasion techniques below.

1. *Evasion by using proxy or NAT.* An adversary may access instant messaging network behind a proxy or a NAT to bypass the detection by the COLD attack algorithm. However, in this scenario, COLD will still detect the external IP address, which appears in the traffic traces collected outside the proxy or NAT. Once the external IP address is detected, our proposed approach will require additional traces collected inside the proxy or NAT to specifically pin-point the end-host.

2. *Evasion by IP spoofing.* An adversary may try to spoof source IP address to evade COLD. However, IP spoofing will not be successful because every end-user has to setup a connection with the IM relay server, which is not possible with spoofed IP address.

3. *Evasion by fragmentation/aggregation.* An adversary may try to break-down a large message into multiple smaller messages. However, fragmentation at the end-host into smaller packets will not adversely affect COLD because our approach relies on correlating the traffic traces that are collected entering and leaving the IM service. The smaller packets created due to fragmentation will appear the same in both sets of traffic traces. In fact, the increased number of packets would improve COLD's accuracy. On the other hand, an adversary may try to aggregate as many messages as possible into a single message to minimize the data available. However, the maximum packet size is limited by the IM service provider and maximum transmission unit (MTU) of the network.

4. *Evasion by changing packet sizes.* If an adversary tries to deliberately change packet sizes, *e.g.* by inserting garbage, they will appear the same in the two sets of traffic

traces correlated by COLD. Therefore, changing packets sizes will not affect COLD.

5. *Evasion by random delays.* Adversaries may also add random small or long delays between their communications. The time delays introduced by end-host will not affect COLD because these delays appear the same in the two sets of traffic traces. In another scenario, random delays may be introduced by the IM network due to network congestion or other processing delays. These delays will affect COLD because they will be different across the two correlated traffic traces. However, COLD is robust to such delays as well because it utilizes binning techniques, which reduces their effect.

6. *Evasion by injecting noise packets.* Injecting random noise packets is unlikely to affect the accuracy of COLD as long as the noise packets follow the protocol utilized by the IM network. Packets that do not follow the protocol utilized by the IM network will be discarded by the IM network after sanity checks and will not appear in the second traffic trace collecting traffic exiting the IM network. To mitigate the effect of such noise packets, similar sanity checks can be deployed to check if the logged packets follow the protocol utilized by the IM network under study.

7. *Evasion by encryption.* Encryption is only applicable to the packet payloads and packet headers remain unaffected. The use of encryption cannot evade COLD because our proposed approach only utilizes fields in the packet header.

4.7 Conclusions

In this work, we present a novel attack to breach the privacy of IM communication services that allows an attacker to infer who's talking to whom with high accuracy. We proposed a wavelet-based scheme, called COLD, that allows us to examine and compare the time series of one-way (user-server) traffic logs at multiple scales. We evaluated the COLD attack algorithm using a real-world Yahoo! Messenger data set, which was specifically collected for

this study. Our experimental results showed that COLD clearly outperforms the baseline time series correlation scheme.

Our proposed approach can also be applied to the related problems such as mix network de-anonymization. In the mix network de-anonymization problem, a set of mix servers can be treated as the black box and the traffic logs at the edges of the mix network can be correlated using COLD to detect communication links among end-users [109, 123].

5 Information Hub Identification in Social Networks

5.1 Introduction

5.1.1 Background and Motivation

In a social network, a user that has a large number of interactions with other users is defined as an *information hub* (or simply a *hub*) [26]. An interaction refers to the transmission of information by one user to another user. For example, an interaction from user A to user B in online social networks may be the action when user A posts a message or comment on user B's profile. Hubs play important roles in the spread or subversion of propaganda, ideologies, or gossips in social networks. Taking the advertising industry as an example, instead of giving free product samples to random people, to improve the effectiveness of word of mouth advertising and increase recommendation based product adoption, they may want to give free samples to hubs only [40]. For example, CNN reported that Samsung used social networks information to target dissatisfied owners of Apple iPhone 4 in a recent advertisement campaign [38]. Samsung first monitored Twitter feeds to identify dissatisfied iPhone 4 owners who are the most active in terms of communicating with their friends (*i.e.* hubs) and are therefore most influential in spreading word of mouth recommendation, then offered free GalaxyS phones to some of them. Furthermore, observing adoption of products or trends at hubs helps to predict the eventual total sale of a product [40]. For instance,

advertisers can observe the impact of distributing free samples to hubs to predict the future successfulness of a product. Due to limited advertisement budget (*e.g.*, free product samples), advertisers want to identify the top-k nodes in a social network. Therefore, identifying top-k information hubs in social networks is an important problem.

5.1.2 Limitations of Prior Art

Prior methods for computing top-k information hubs (*e.g.*, [61] and [41]), are mostly centralized assuming the availability of either interaction or friendship graphs. The interaction graph of a social network is a directed multigraph [21] whose nodes represent users and directed links represent the existence of a directed pair-wise interaction. Each link is labeled with a time stamp that indicates when the interaction occurred. The friendship graph of a social network consists of nodes representing users and undirected links representing the friend relationship among users. Figure 5.1 shows the conceptual depiction of the friendship graph between users and the overlaid interaction graph. However, centralized computation of top-k information hubs is mostly unrealistic for parties such as advertisers because online social networking companies are reluctant to share their interaction or friendship graphs due to privacy concerns and regulations [8]. Furthermore, advertisers cannot even directly collect interaction or friendship information from social network sites by means such as crawling because for many online social networking companies such as Facebook [9], unauthorized data collection is a violation of their terms of service.

5.1.3 Proposed Solution

In this work, we propose a distributed and privacy preserving algorithm for computing top-k information hubs in social networks. Distributed algorithms for computing top-k information hubs have to be privacy preserving because users are typically hesitant to disclose explicit information about their friendship links or interaction information due to privacy concerns [121]. To preserve the privacy of user interactions, our algorithm is distributed and

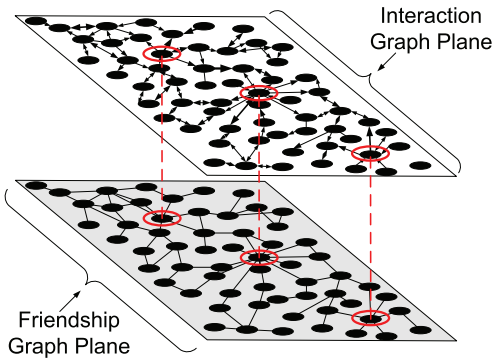


Figure 5.1. Conceptual depiction of the friendship graph between users and the overlaid interaction graph.

does not require the advertiser to know users' friendship associations or interactions. There are three technical challenges in designing such an algorithm. First, the problem of inferring a user's salience (whose ground truth resides in the interaction graph) from the corresponding friendship graph is inherently difficult because an interaction graph has more information than its corresponding friendship graph. Furthermore, a friendship graph is undirected and un-weighted, whereas the interaction graph is a directed multigraph. Second, the complete friendship graph itself may not be available to the parties interested in identifying hubs. Third, preserving users' privacy in this computation is difficult as any information exchange involved in this computation should not contain any personal information.

We now present an overview of our proposed solutions to the above-mentioned technical challenges. To address the first challenge, we apply principal component centrality (PCC), a new measure of centrality we introduced in [45], to the friendship graphs. The intuition behind PCC is that a user who is connected to well-connected users (even if the user himself is poorly connected) has a more central status. For example, a poorly connected person who has a direct connection with a well-connected representative in some population may be capable of propagating an opinion well by simply convincing the representative. Unlike other measures of user influence (*e.g.*, eigenvector centrality [22,23]), PCC takes into consideration the fact that social networks can be multi-polar consisting of multiple communities with few

connections between them.

To address the second challenge of friendship graph data availability, we distribute the computation of PCC among users and therefore do not require a central entity to access the friendship graph. Advertisers can utilize existing functionality in popular online social networks (such as *groups* and *pages* in Facebook [8]) to implement our proposed distributed method. Motivation for user participation in decentralized PCC computation may range from tangible incentives such as receiving free samples from advertisers (*e.g.* [38]), to intangible incentives such as bragging rights about one's popularity (*e.g.* [102]). We decentralize the PCC computation using the Kempe-McSherry (KM) algorithm [50]. These iterative algorithms compute eigenvalues and eigenvectors that are essential for computing nodes' PCCs. Our decentralized algorithm restricts the set of users that a particular user has to communicate with to its immediate friends. Furthermore, the memory requirement at each user of this algorithm grows only linearly with the number of friends. Hence, one of the contributions of this work is extending the original centralized PCC approach to a more practical distributed PCC form. This new distributed PCC form is an accurate and robust centrality measure that is capable of identifying all salient users in a social graph using a truly decentralized and scalable method.

Finally, to address the issue of user privacy, only real numbers representing PCC intermediate scores are exchanged between users. It is impossible to reverse-engineer users' friendship associations from these intermediate scores.

5.1.4 Experimental Results and Findings

We evaluated the effectiveness of our proposed technique using real-world Facebook data sets [111] containing about 6 million users and more than 40 million friendship links. We have four major findings from our experimental results. First, there is indeed close correlation between the PCC of nodes in the friendship graph and corresponding dynamic user interaction data. We envision that this correlation can be exploited for other purposes as well. Second, the

computation of PCC can be effectively distributed across individual users in a social graph without compromising its accuracy. This eliminates the requirement of a central authority for identifying hubs. Third, the accuracy of PCC improves as we use more eigenvectors in its computation. Further, the appropriate number of eigenvectors required in the computation of PCC for real-world social networks is around 10-20. Fourth, the accuracy (in terms of number of correctly identified top-k users and their estimated rank) of PCC improves as the duration of interaction data used for comparison is increased from 1 month, to 6 month to more than a year. This essentially shows that PCC scores reflect the flow of information between users of a social network over long time periods.

5.1.5 Key Contributions

We make four key contributions in this work. First, we propose a novel method to infer information lying in the interaction graph (*i.e.* hub identification) from the friendship graph in social networks without using the interaction data. Earlier works are limited to solving this problem using complete interaction graph data. Second, our proposed method, first of its kind, allows third parties (other than social network owners) to solve this problem. We use a distributed method to overcome the requirement of a central authority. Third, our proposed method preserves the privacy of users, *i.e.*, users do not release any personal information to other users. We achieve this objective by letting each user share only some real numbers that cannot be reverse-engineered. Finally, we evaluate the effectiveness of our proposed technique using real-world Facebook data sets that are publicly available. The results of our experiments show that the proposed approach improves the accuracy of finding the top-k user set by approximately 50% over existing measures. Furthermore, the proposed technique accurately estimates the rank of individual users.

The rest of this chapter proceeds as follows. In Section 5.2, we present an overview of related work. We provide the details of our proposed approach in Section 5.3. We also provide the analysis of the data set used for evaluating our proposed technique in Section

5.4.1. We then provide the detailed results of our evaluation in the rest of Section 5.4. Finally, we conclude the chapter in Section 5.5.

5.2 Related Work

Besides work on hub identification using user interaction data ([41,61]) mentioned in Section 5.1, we provide an overview of rest of the relevant research on influence maximization.

Several algorithms have been proposed for identifying top-k influential users in social networks [34,51–53,106,124]. The objective function for this influence maximization problem is to maximize the number of users that are part of information flows initiated by the top-k influential users. In contrast, our method uses friendship graphs, is fully distributed, and is privacy-preserving, while such work uses user interaction data and is centralized and is not privacy-preserving.

Kempe *et al.* studied this influence maximization problem for the first time, proved it is NP-hard, and proposed a heuristics-based algorithm that achieves 63% of the optimal result in most cases and outperforms degree and distance centrality heuristics [51]. Suri and Narahari later proposed Shapley value based heuristic for solving this problem [106]. Zou *et al.* studied the same problem with an additional constraint of latency [124]. Estevez *et al.* proposed an algorithm called the Set Covering Greedy (SCG) algorithm, which takes into account the intuition that we should prefer to select nodes in different neighborhoods rather than selecting highly connected nodes lying in the same neighborhood [34]. Kimura *et al.* studied the influence maximization problem with respect to two widely-used fundamental stochastic information diffusion models in networks, and proposed a solution utilizing tools from bond percolation and graph theory [52, 53].

Algorithms that forgo using interaction data use structural information like the friendship graph. They are based on centrality measures computed from friendship graph topologies. Marsden [69] used degree, closeness, betweenness and eigenvector centrality measures. This

is followed by Shi *et al.* in [104] who used the same centrality measures, *i.e.* degree, closeness, betweenness and pagerank [54,57,85] (which is just an iterative algorithm to compute eigenvector centrality). However, as Borgatti showed in [24], degree, closeness and betweenness centrality are inappropriate measures of centrality for influence processes. Degree centrality is a good measure of the rate of immediate rate of spread of influence from nodes in the short-term. Betweenness and closeness centrality are ill-suited for the problem at hand because the definitions underlying them assume that the flow on the network does not replicate and occurs only along shortest paths. Therefore, the performance of Marsden's use of eigenvector centrality and Shi's use of Pagerank form the baseline for comparison against our proposed algorithm. Canright, Engø-Monsen and Jelasity [27] described a distributed and privacy preserving algorithm for the computation of eigenvector centrality/PageRank.

5.3 Our Proposed Solution

This section presents our proposed technique for identifying information hubs in social networks. We model the information flow as an influence process. The underlying rationale for doing so is rooted in the assumption that in social networks people (nodes) with more friends (connections) send and receive more messages. Furthermore, people will receive more messages from friends that send/receive a lot of traffic than from those that send/receive fewer messages. This information flow can be modeled as an influence process. According to Borgatti's two dimensional taxonomy of node centrality measures in [24], the appropriate measure to quantify nodes' influence is eigenvector centrality (EVC) [22,23].

5.3.1 Eigenvector Centrality

Let $\mathbf{A}$ denote the adjacency matrix of a graph $G(V, E)$ consisting of the set of nodes $V = \{v_1, v_2, v_3, \ldots, v_N\}$ of size N and set of undirected edges E. When a link is present between two nodes v_i and v_j, both $A_{i,j}$ and $A_{j,i}$ are set to 1 and set to 0 otherwise. Let $\Gamma(v_i)$

denote the neighborhood of v_i, the set of nodes v_i is connected to directly. EVC of a node is recursively defined as proportional to the number of its neighbors and their respective EVCs. Let $x(i)$ be the EVC score of a node v_i. Then,

$$x(i) = \frac{1}{\lambda_1} \sum_{v_j \in \Gamma(v_i)} x(j) = \frac{1}{\lambda_1} \sum_{j=1}^{N} A_{i,j} x(j) \tag{5.1}$$

Here λ_1 is a constant (later found to be the principal eigenvalue of $\mathbf{A}$). Equation 5.1 can be rewritten in vector form Equation 5.2 where $\mathbf{x}_1 = [x(1), x(2), x(3), \ldots, x(N)]^T$ is the vector of EVC scores of all nodes.

$$\mathbf{x}_1 = \frac{1}{\lambda_1} \mathbf{A}\mathbf{x}_1 \iff \quad \lambda_1 \mathbf{x}_1 = \mathbf{A}\mathbf{x}_1 \tag{5.2}$$

Equation 5.2 is the well-known eigenvector equation where this centrality takes its name from. Obviously several eigenvalue/eigenvector pairs exist for an adjacency matrix $\mathbf{A}$. Here, λ_1 is the largest of all eigenvalues of $\mathbf{A}$ by magnitude. If λ_i is any other eigenvalue of $\mathbf{A}$ then $|\lambda_1| > |\lambda_i|$. The eigenvector $\mathbf{x}_1 = [x_1(1), x_1(2), \ldots, x_1(N)]^T$ corresponding to the principal eigenvalue is the principal eigenvector. Thus, the vector of node EVCs is equivalent to the principal eigenvector. The EVC of a node v_i is the corresponding element $\mathbf{x}_1(i)$ of the principal eigenvector $\mathbf{x}_1$.

5.3.2 Motivation for Principal Component Centrality

As we demonstrated in [45], in networks of multiple communities with sparse connectivity between communities, EVC assigns centrality scores to nodes according to their location with respect to the most dominant community. When applied to large networks, EVC fails to assign significant scores to a large fraction of nodes. The principal eigenvector is "pulled" in the direction of the largest community. The motivation for using PCC as a measure of

node influence may be understood by looking at EVC in the context of principal component analysis (PCA) [32]. In PCA, when feature vectors are extracted from an $N \times N$ covariance matrix of N random variables, the principal eigenvector is the most dominant feature vector, i.e. the direction in N-dimensional hyperspace along which the spread of data points is greatest. Similarly, the second eigenvector (corresponding to the second largest eigenvalue) is representative of the second most significant feature of the data set. The second eigenvector may also be thought of as the most significant feature after the data points are collapsed along the direction of the principal eigenvector. When the covariance matrix is computed empirically from a set of data points, the eigendecomposition is the well-known PCA. Here PCA is used to find the eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N$ and eigenvalues $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_N$ of the graph G's adjacency matrix $\mathbf{A}$.

5.3.3 Definition of PCC

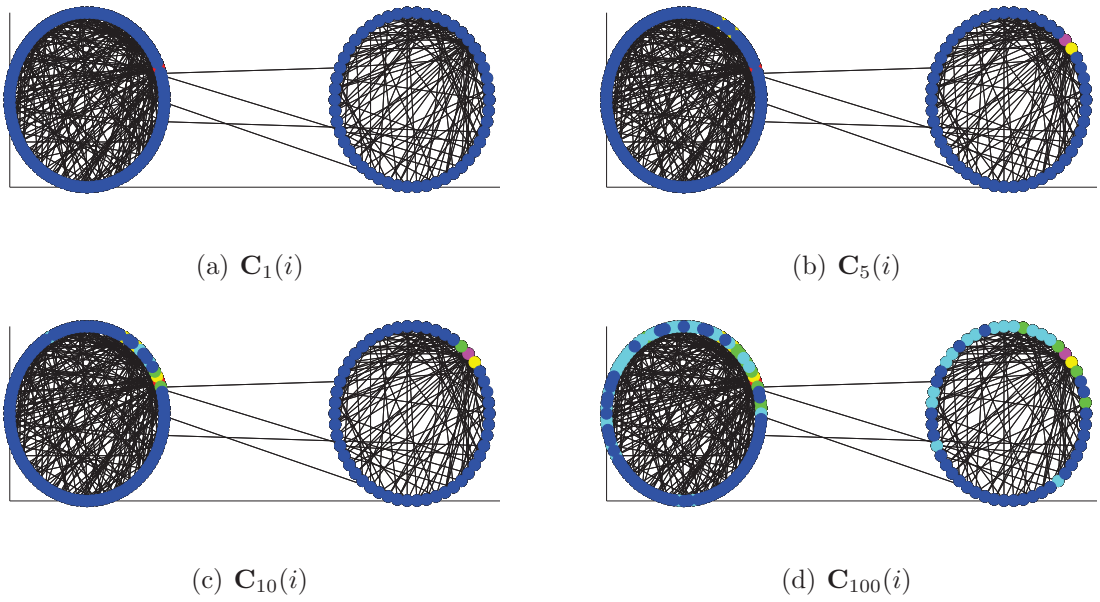
While EVC assigns centrality to nodes according to their location with respect to the most dominant community in a graph G, PCC takes into consideration additional communities. We define the PCC of a node in a graph as its Euclidean distance/ℓ^2 norm from the origin in the P-dimensional eigenspace. The basis vectors of that eigenspace are the P most significant eigenvectors of the adjacency matrix A of the graph G under consideration. For a graph G, its N eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_N|$ correspond to the normalized eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, respectively. The eigenvector/eigenvalue pairs are indexed in descending order of magnitude of eigenvalues. When $P = 1$, PCC equals a scaled version of EVC. The parameter P in PCC can be used as a tuning parameter to adjust the number of eigenvectors included in PCC.

Let $\mathbf{X}$ denote the $N \times N$ matrix of concatenated eigenvectors $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N]$ and let $\mathit{\Lambda} = [\lambda_1 \lambda_2 \ldots \lambda_N]^T$ be the vector of eigenvalues. Furthermore, if $P < N$ (typically $P \ll N$) and if matrix $\mathbf{X}$ has dimensions $N \times N$, then $\mathbf{X}_{N \times P}$ will denote the submatrix of $\mathbf{X}$ consisting of the first N rows and first P columns. Then PCC can be expressed in matrix form as:

$$\mathbf{C}_P = \sqrt{((\mathbf{A}\mathbf{X}_{N\times P}) \odot (\mathbf{A}\mathbf{X}_{N\times P}))\,\mathbf{1}_{P\times 1}} \tag{5.3}$$

The '$\odot$' operator is the Hadamard (or entrywise product or Schur product) operator and $\mathbf{1}_{P\times 1}$ is a vector of 1s of length P. Equation 5.3 can also be expressed in terms of the eigenvalue and eigenvector matrices Λ and $\mathbf{X}$, of the adjacency matrix $\mathbf{A}$:

$$\mathbf{C}_P = \sqrt{(\mathbf{X}_{N\times P} \odot \mathbf{X}_{N\times P})\,(\Lambda_{P\times 1} \odot \Lambda_{P\times 1})}. \tag{5.4}$$



(a) $\mathbf{C}_1(i)$ (b) $\mathbf{C}_5(i)$

(c) $\mathbf{C}_{10}(i)$ (d) $\mathbf{C}_{100}(i)$

Figure 5.2. PCC of nodes in a network consisting of two Barabási-Albert graphs of 100 and 50 nodes connected by a few links when computed using the most significant (a) 1, (b) 5, (c) 10, and (d) 100 eigenvectors.

Node PCCs as defined in equations 5.3 and 5.4 are not normalized. To allow interpretation of centrality scores, Ruhnau advocated in [84] that they should be normalized by either the Euclidean norm (ℓ_2 norm) or the maximum norm (ℓ_∞ *i.e.* the maximum centrality score)

of the centrality vector. For the remainder of this work the PCC vector will be normalized by the ℓ_∞ norm, thereby restricting all entries to the range $[0, 1]$.

We demonstrate PCC on a small-scale example, a graph consisting of two Barabási-Albert graphs [19], one consisting of 100 nodes in one community that is sparsely connected with another Barabási-Albert graph of 50 nodes. Figure 5.2 demonstrates the effect of changing number of eigenvectors P for PCC $\mathbf{C}_P$ from 1 (Figure 5.2(a)) for EVC to 5 (Figure 5.2(b)), 10 (Figure 5.2(c)) and 100 (Figure 5.2(d)). As this example shows, EVC is only able to assign significant centrality to the most well connected node in the larger of the two subgraphs. As P is raised to 5 and 10, gradually more nodes are assigned significant centrality scores, even some in the smaller subgraph of 50 nodes.

5.3.4 Selection of Number of Eigenvectors

The cost of computing an eigenvector can be significant for large matrices, favoring the use of as few eigenvectors for PCC as are necessary. To determine appropriate number of eigenvectors (P_{app}), we consider the phase angle ϕ as a function of P. The phase angle $\phi(P)$ of a PCC vector $\mathbf{C}_P$ is defined as its angle with the EVC vector $\mathbf{C}_E$ and is defined mathematically in equation 5.5.

$$\phi(P) = \arccos\left(\frac{\mathbf{C}_P}{|\mathbf{C}_P|} \cdot \frac{\mathbf{C}_E}{|\mathbf{C}_E|}\right) \tag{5.5}$$

When the phase angle function is plotted for a range of P, the value of P at which ϕ begins approaching its final steady value is used for that particular graph (P_{app}) [45]. The selection of P_{app} can be made as,

$$P_{app} = min\{\phi(P+1) - \phi(P)\} \quad \in [-\epsilon, \epsilon], \forall\ [P, N], \tag{5.6}$$

where ϵ is a small real number. It is our observation that the value of P_{app} is close to the number of well-connected communities in a social graph.

5.3.5 Decentralized Eigendecomposition Algorithm

The massive sizes of social networks require a method whose space and time complexity scales well with the number of nodes and links between them. According to Equation 5.3, for a node to compute its own PCC score it needs to know its corresponding entries in the first P eigenvectors $\mathbf{x}_1$, $\mathbf{x}_2$ ... $\mathbf{x}_P$ of the adjacency matrix $\mathbf{A}$, as well as who its neighbors are, i.e. the entries in its corresponding row of A. Although many decentralized algorithms for computing eigenvectors of a matrix exist, many of them are not designed to minimize the communication overhead between participating nodes. We discuss 2 well-known distributed algorithms in the following text.

A method that lends itself to a decentralized implementation is the power iteration algorithm [56]. This algorithm avoids computing a matrix decomposition and is suitable for very large, sparse matrices. Canright, Engø-Monsen and Jelasity [27] describe a fully distributed implementation of the power iteration algorithm for computing the principal eigenvector. The power iteration algorithm exploits the fact that the principal eigenvector of a symmetric matrix can be interpreted as the steady state distribution of an ergodic Markov chain. The transition matrix of this Markov model is the stochastic, row-normalized version of the adjacency matrix $\mathbf{A}$. The power iteration algorithm initializes $\mathbf{x}$ as a random vector and iteratively computes $\frac{\mathbf{Ax}}{\|\mathbf{Ax}\|_\infty} \rightarrow \mathbf{x}$ until $\|\mathbf{x}\|_\infty$ becomes stable within a margin of tolerance (tol). The denominator term $\|\mathbf{Ax}\|_\infty$ is the eigenvalue λ corresponding to the eigenvector $\mathbf{x}$. To compute subsequent eigenvectors using the power iteration algorithm, as mentioned in [60], the adjacency matrix is "deflated," i.e. $\mathbf{A}_{i+1} = \mathbf{A}_i - \lambda_i \frac{\mathbf{x}_i \mathbf{x}_i^T}{\mathbf{x}_i^T \mathbf{x}_i}$. The deflation operation removes from adjacency matrix $\mathbf{A} = \mathbf{A}_1$ the structure that is explained by the first eigenvector $\mathbf{x}_1$. The principal eigenvector of the deflated matrix $\mathbf{A}_2$ is computed using the power iteration algorithm again. $\mathbf{A}_2$'s principal eigenvector is then the second eigenvector of the original matrix $\mathbf{A} = \mathbf{A}_1$. Thus, starting from $\mathbf{A}$, P applications of the power iteration algorithm and $P-1$ deflations will produce the P most significant eigenvalues $\lambda_1, \lambda_2, ..., \lambda_P$ and eigenvectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_P$.

The pseudo code in Algorithm 1 describes the power iteration and deflation algorithms. In the application at hand, the algorithm takes friendship graph's adjacency matrix $\mathbf{A}$ and P number of eigenvectors as input. Typically, friendships in graphs obtained from online social networks are recorded as bidirectional, making $\mathbf{A}$ a symmetric matrix. This ensures that all its eigenvalues will be real. The principal eigenvector $\mathbf{x}_1$ can be initialized arbitrarily [27]. Note that according to Algorithm 1, the messages exchanged between nodes do not contain any link information or other structural information about the graph. Nodes do not reveal either their own or any other nodes' friends. The fact that only a numeric value is exchanged by communicating nodes preserves privacy.

While the power iteration algorithm provides sufficiently accurate results for the principal eigenvector, it is not suitable for computing many subsequent eigenvectors/eigenvalues for two reasons. First, it suffers from rounding error that becomes progressively worse with the computation of each subsequent eigenvector. Second, the number of messages that have to be exchanged by a node grows larger for every successive eigenvector that is computed. Consider an illustrative example in Figure 5.3 where the node under consideration is colored black. The neighbor nodes it is connected with in the friendship graph are colored grey and non-neighbor nodes are colored white. Figure 5.3(a) illustrates the exchange of messages to/from the node when the first eigenvector is computed. In this scenario the node does not need to communicate outside its circle of friends. Recall that to compute the second eigenvector and eigenvalue the power iteration algorithm is applied to the once deflated adjacency matrix $\mathbf{A}_2$, which will include non-zero entries that will require nodes to venture outside of their circle of neighbors and exchange messages with nodes to whom they are not connected to in the friendship graph. This is shown in Figure 5.3(b). However, the content of the message exchange is still only numeric values. After a few deflation operations $\mathbf{A}_i$ will not remain sparse anymore, thus increasing the number of messages that have to be sent/received by every node in the friendship graph. This case is illustrated in Figure 5.3(c). The number of message exchanges can be reduced at the cost to the rounding error by rounding very small

Algorithm 1 Power iteration algorithm with deflation

Input: N = Number of nodes in graph

Input: $\mathbf{A}$ = Friendship graph adjacency matrix of size $N \times N$

Input: P = Number of eigenvectors to be computed

Output: $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_P = P$ largest eigenvectors

Output: $\lambda_1, \lambda_2, ..., \lambda_P = P$ largest eigenvalues

$\mathbf{x}_1 \leftarrow [1\ 1\ 1\ ...\ 1]^T$

$\mathbf{A}_1 = \mathbf{A}$

for $i = 1$ to P **do**

  $\lambda_{i,prev} \leftarrow 0$

  $\lambda_i \leftarrow 1$

  while $\frac{\lambda_i - \lambda_{i,prev}}{\lambda_i} > tol$ **do**

    $\lambda_{i,prev} = \lambda_i$

    $\mathbf{x}'_i \leftarrow \mathbf{A}_i \mathbf{x}_1$

    $\lambda_i = \|\mathbf{x}'_i\|_\infty$

    $\mathbf{x}_i = \frac{\mathbf{x}'_i}{\lambda_i}$

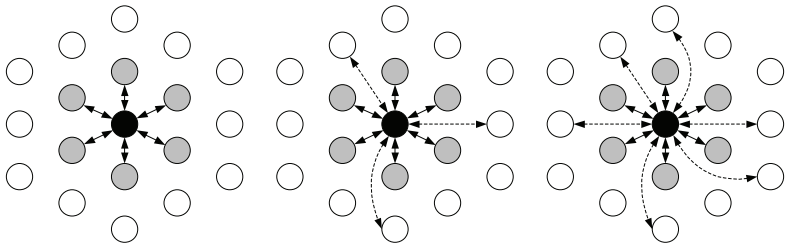
  end while

  $\mathbf{A}_{i+1} \leftarrow \mathbf{A}_i - \lambda_i \frac{\mathbf{x}_i \mathbf{x}_i^T}{\mathbf{x}_i^T \mathbf{x}_i}$

end for

entries in eigenvectors $\mathbf{x}_i$ and deflated matrices $\mathbf{A}_i$ to zero.

In [50] Kempe and McSherry developed a decentralized algorithm for the computation of the first P most significant eigenvectors. Their approach differs from other algorithms in that each node is only required to communicate with neighbor nodes, as shown in Figure 5.3(a). This means that the computational complexity of the algorithm at every node, and the volume of messages exchanged by each node scales only linearly with the number of its neighbors and linearly with the number of eigenvectors that are computed. Furthermore, the time for the algorithm to converge is $O(\tau_{mix} \log N)$, where N is the total number of nodes in



(a) 1st eigenvector (b) 2nd eigenvector (c) 3rd (subsequent) eigenvectors

Figure 5.3. An illustrative example of message exchanges in power iteration algorithm for (a) 1st eigenvector, (b) 2nd eigenvector, and (c) subsequent eigenvectors. Reference node is colored black, neighbor nodes grey and non-neighbor nodes white.

the graph and τ_{mix} is the mixing time of the Markov chain with a transition matrix that is the row-normalized version of $\mathbf{A}$. Although the power method's overhead and convergence properties vary greatly from those of the KM algorithm, the iterative components of the KM algorithm are very similar to those of the power method (lines 6 to 11 in Algorithm 1) when it is used in the computation of the principal eigenvector only. Both algorithms perform a deterministic simulation of a random walk. For a detailed coverage of the KM algorithm we refer the reader to [50].

Kempe reported the error of the ℓ_2 norm of the space spanned by R_P, the projection of P most significant eigenvectors on $\mathbf{A}$, and $R_{P\prime}$, the projection of P most significant eigenvectors by KM-algorithm onto $\mathbf{A}$, with high probability as follows.

$$\|\mathbf{R}_P - \mathbf{R}_{P\prime}\|_2 \leq O\left(\left|\frac{\lambda_{P+1}}{\lambda_P}\right|^t \cdot N\right) + 3\epsilon^{4t} \tag{5.7}$$

Here, t denotes the number of iterations for which the KM algorithm executes. Clearly, since $\lambda_{P+1} < \lambda_P$, the fractional term will be decreasing with t at a geometric rate. Figure 5.4 shows a plot of average mean squared error (MSE) between the actual and estimated top-k eigenvectors (using the KM algorithm) for random graphs of 100 nodes. We report average

MSE values for varying number of eigenvectors (k) and number of iterations (t). Each point in the plot is an average of 1000 independent runs and the confidence intervals are too small to be shown. As expected from equation 5.7, we observe that average MSE values sharply decrease approximately at a geometric rate for increasing number of iterations. Furthermore, for a given number of iterations, average MSE values increase for larger k values.

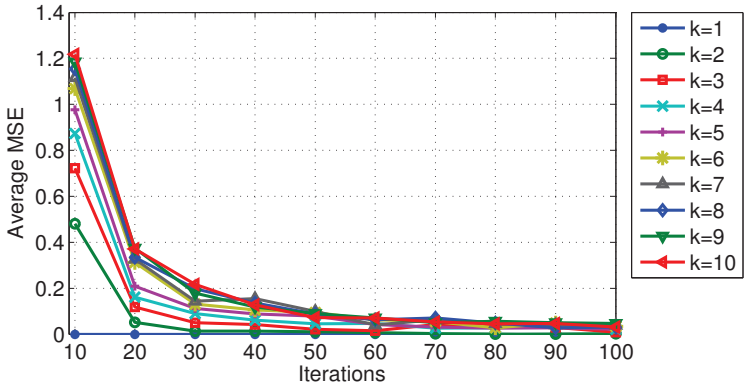


Figure 5.4. Average Mean Squared Error (MSE) for the KM algorithm reported for varying values of number of eigenvectors (k) and number of iterations (t).

From the above discussions of the power method with deflation and the KM algorithm, we conclude the following:

- The power method with deflation has a communication overhead that grows exponentially with each additional eigenvector computation. This limits the number of eigenvectors that can be used in the computation of PCC. On the other hand, the KM algorithm never requires a node to communicate beyond its immediate neighbors. This implies that the communication overhead of the KM algorithm scales linearly with the number of computed eigenvectors.

- The power method with deflation suffers from compounding of round-off errors in the computation of subsequent eigenvectors. This limits the maximum number of eigenvectors that can be used in computation of PCC. On the other hand Kempe *et*

al. reported near perfect convergence for their algorithm, which is also verified from our observations in Figure 5.4.

For these reasons, we choose to use the KM algorithm for the distributed computation of eigenvectors for PCC.

5.4 Experimental Results

5.4.1 Data Set

We now present details of the data sets used to evaluate the efficacy of our proposed technique. In our study, we use two independently collected data sets from Facebook [111]. We use both data sets to demonstrate that our proposed solution is not biased in favor of any particular data set. The data sets are labeled data set A and data set B here-onwards. As Wilson *et al.* describe in [111], at the time of collection in April 2008 Facebook had 67 million subscribers of whom 44.3 million belonged to a regional network (regional networks were defined on the basis of geography and institutions). Each regional network forms a community of nodes that are strongly intra-connected but sparsely connected to other communities. Their crawler performed a breadth-first-search and collected data from the 22 largest regional networks. The crawler was initialized with 50 randomly seeded user profiles. Wilson *et al.* verified the completeness of their coverage of regional networks by performing 5 simultaneous crawls of the San Francisco regional network, each seeded by a different number of seed user IDs varying from 50 to 5000. The difference in the number of users discovered between crawls was a mere 0.1%. Therefore, we can conclude that the coverage of users in these data sets is fairly complete. The data contained in data set A and data set B is from different regions.

Each data set further consists of two types of graphs. First, we have an undirected friendship graph where the nodes represent users and links represent the friendship between two users. Second, we have a directed pair-wise user interaction graph where the nodes repre-

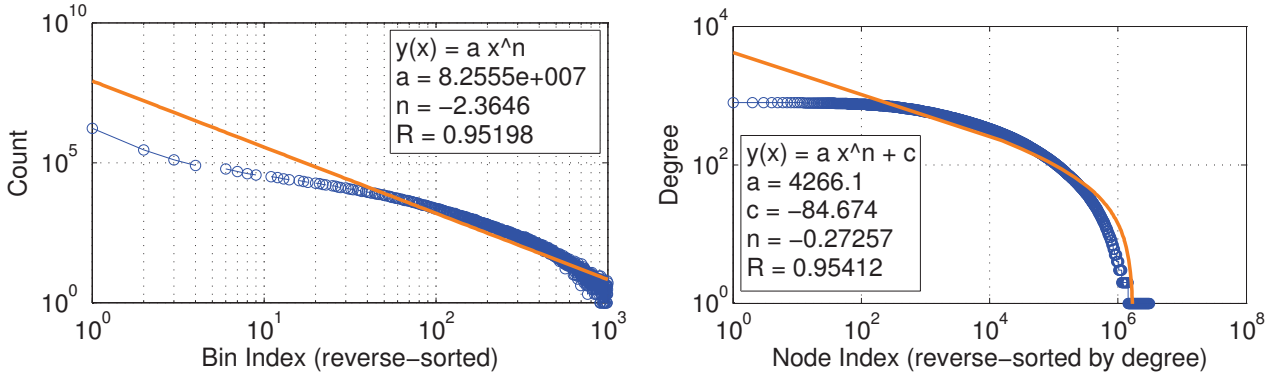
sent users and the directed links represent the interaction from one user to another. The interaction data spans a time duration of one year. Note that we use the interaction data only to evaluate the ground truth.

Table 5.1 provides the basic statistics of the friendship graphs analyzed in this study. We note that the number of users in data set A are slightly more than those in data set B. We also note that the ratio of the number of friendship links to the number of users for data set A is ~ 7.6, which is slightly more than ~ 7.1 for data set B. This statistic is also reflected in the values of average clustering coefficients of both data sets. Moreover, the number of cliques in the friendship graph of data set A is more than those in data set B. However, we observe that the transitivity value (defined as the fraction of possible triangles that are actually triangles) for data set A is less than the respective value of data set B.

Table 5.1. Basic statistics of the friendship graphs analyzed in this study

Property	**Data set A**	**Data set B**
# Users	3097165	2937612
# Friendship Links	23667394	20959854
Average Clustering Coefficient	0.0979	0.0901
# Cliques	28889110	27593398
Transitivity	0.0477	0.04832

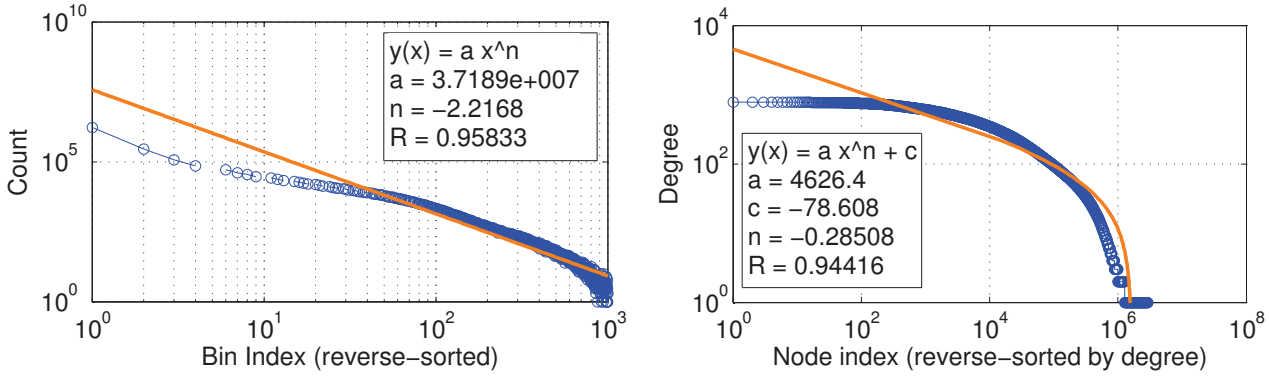
Figures 5.5 and 5.6 show the plots of degree distributions for friendship graphs of both networks. In Figures 5.5(a) and 5.6(a), we plot the histograms of one thousand bins in which we have grouped users. Although the distribution does not follow a power-law exactly, it fits it reasonably well as shown by straightness on log-log scale and verified by high goodness-of-fit (R^2) values for data set A and data set B. This observation is in accordance with the result of recent studies that have shown that the degree distribution of many online social networks is power-law [71]. An equivalent representation is shown in Figures 5.5(b) and 5.6(b) where users are reverse-sorted by their degree. Note that the estimated values of model parameters are similar for both data sets.



(a) Power law

(b) Zipf-like

Figure 5.5. Degree distribution of friendship graph for Facebook data set A



(a) Power law

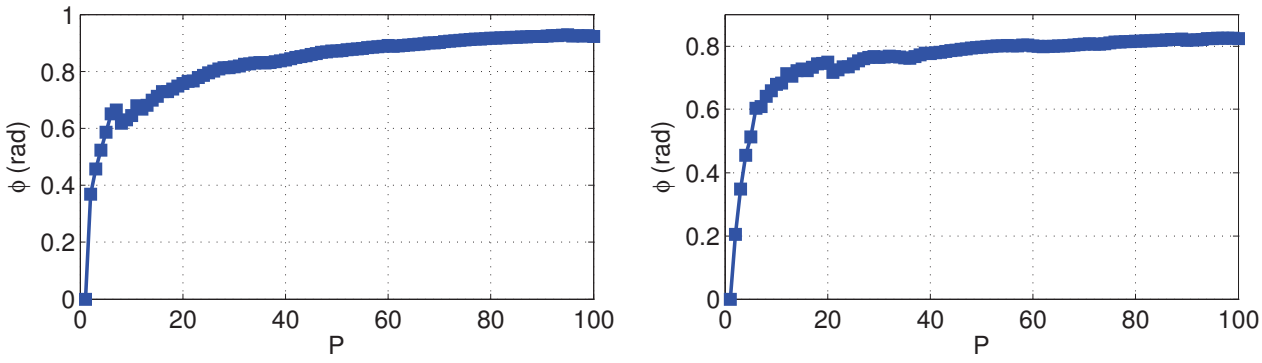
(b) Zipf-like

Figure 5.6. Degree distribution of friendship graph for Facebook data set B

5.4.2 Selection of PCC Parameter

We can compute the PCC vector $\mathbf{C}_P$ for a range of number of eigenvectors P. Note that at $P = 1$ the PCC $\mathbf{C}_1$ is the EVC $\mathbf{C}_E$, which serves as the measure of baseline comparison, as mentioned in [104] and [69]. Although we will be comparing PCC with EVC for a range of values of P in some of our subsequent analysis, we will try to determine the "appropriate" number of eigenvectors for PCC (denoted by P_{app}). We do this by means of plotting the phase angle function defined in Equation 5.5. Figures 5.7(a) and 5.7(b) plot the phase angle functions of Facebook data sets A and B, respectively, for the range of $P = 1$ to 100. For

data sets A and B, the phase angle function rises quickly initially until $P = 6$ and rises only very slowly thereafter. Using Equation 5.6, the P_{app} values are 10 and 20 for data sets A and B, respectively. The difference in P_{app} values for data sets A and B can be explained by differences in their network structures. In Table 5.1, we showed that the friendship graph of data set A has higher average clustering coefficient values than the friendship graph of data set B. This essentially shows that the friendship graph of data set A is on average more tightly connected than the friendship graph of data set B. In terms of PCC computation, this indicates that all nodes can be reached from a given node in lesser number of hops on average. In other words, fewer number of eigenvectors (denoted by P_{app}) are enough to approximate the steady-state PCC value.



(a) Facebook data set A

(b) Facebook data set B

Figure 5.7. Plot of the phase angle $\phi(P)$ between PCC vectors $\mathbf{C}_P$ and EVC vector $\mathbf{C}_E$ plotted against number of feature vectors P for (a) Facebook data set A, and (b) Facebook data set B.

5.4.3 Comparison With Ground Truth

Now that we have identified an appropriate number of eigenvectors for PCC for both data sets, we devote the remaining section to evaluating its accuracy by comparing the results to the ground truth, *i.e.* interaction data. For both data sets A and B, we have interaction

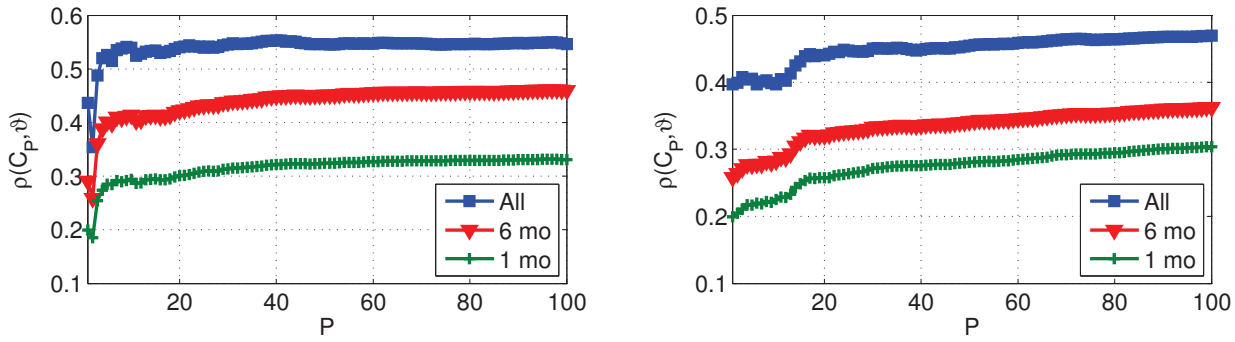
graphs spanning 1 month, 6 months, and 1 year time periods.

Recall that the PCC scores for individual users are calculated using only information from the friendship graph. We compare the PCC of nodes against their actual flows over various time periods to get a sense of the time period over which PCC best predicts the flow.

We have used a symmetric measure called Pearson's product-moment coefficient to quantify the similarity between the output of PCC and the ground truth from interaction data. The Pearson's product-moment coefficient ρ is defined in Equation 5.8. Here E is the expectation operator, σ refers to standard-deviation, and μ denotes mean value.

$$\rho(\mathbf{C}_P, \vartheta) = \frac{E[(\mathbf{C}_P - \mu_{\mathbf{C}_P})(\vartheta - \mu_\vartheta)]}{\sigma_{\mathbf{C}_P}\sigma_\vartheta} \tag{5.8}$$

Figure 5.8 shows the plots of correlation coefficients $\rho(\mathbf{C}_P, \vartheta)$ as a function of number of eigenvectors for the range $1 \leq P \leq 100$. Figure 5.8(a) plots $\rho(\mathbf{C}_P, \vartheta)$ for flows collected over 1 month, 6 months and the entire collection time period (labeled 'All') for data set A. Figure 5.8(b) does the same for data set B.



(a) Facebook data set A

(b) Facebook data set B

Figure 5.8. Correlation coefficients ρ of PCC $\mathbf{C}_P$ and, (a) flow count of Facebook data set A ($\vartheta(A)$), (b) flow count of Facebook data set B ($\vartheta(B)$). The correlation coefficients are plotted as functions of the number of eigenvectors P and plotted separately for each interaction graph.

We make two major observations from these plots. First, we note that the value of ρ

generally increases with increasing number of eigenvectors P for computing PCC. It rises quickly to reach its steady-state value for both data sets. For Facebook data set A, ρ reaches close to its steady-state value at around 10 eigenvectors. Whereas, for Facebook data set B, ρ reaches close to its steady-state value at around 20 eigenvectors. Note that the steady-state values for ρ are reached at P_{app} values selected in the previous subsection. This observation verifies the merit of using phase angle for selection of appropriate value of P in PCC computation.

Second, we note that the correlation coefficients are higher for interaction data collected over longer periods of time. This observation follows our intuition that the trends in short-term interaction data can deviate from our expectations in steady-state friendship graph; however, the trends in long-term interaction data show greater similarity with the underlying friendship graph. This observation remains consistent across both Facebook data sets.

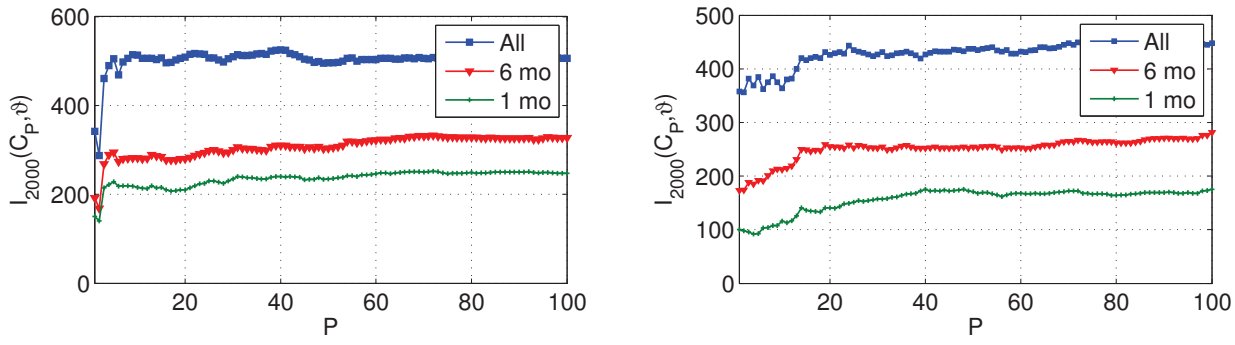
To further evaluate the accuracy of PCC in finding information hubs, we analyze the overlap between the set of top-2000 users by PCC (denoted by $S_{2000}(\mathbf{C}_P)$) and the ground truth. Note that the choice of 2000 nodes in the following analysis is purely arbitrary. The results of our analysis for different set sizes are qualitatively similar. Let the cardinality of the intersection set of the first k nodes by PCC and the first k nodes by flow ϑ be denoted by $I_k(\mathbf{C}_P, \vartheta)$ and defined in Equation 5.9 below.

$$I_k(\mathbf{C}_P, \vartheta) = \frac{|S_k(\mathbf{C}_P) \cap S_k(\vartheta)|}{k} \tag{5.9}$$

Figures 5.9(a) and 5.9(b) plot I_{2000} for data sets A and B, respectively. We evaluate separately for interaction data of different durations. As expected, the cardinality of the intersection set increases with the number of eigenvectors used in computation of PCC. In both figures, the data points at $P = 1$ represent the baseline for our comparison, *i.e.* EVC.

For data set A, the cardinality of the intersection set of the top-2000 nodes by EVC and top-2000 nodes by flow ϑ, the cardinality of the intersection set $I_{2000}(\mathbf{C}_E, \vartheta)$ is 342. At $P = 10$, $I_{2000}(\mathbf{C}_{10}, \vartheta) = 513$ for data set A. These represent increases of 50.0%. For

Facebook data set B intersection cardinality of EVC set with flow are $I_{2000}(\mathbf{C}_E, \vartheta) = 358$. At $P = 20$, $I_{2000}(\mathbf{C}_{20}, \vartheta) = 426$ for data set A, an increase of 19.0%. For the remainder of this section, we fix the values of P at P_{app} for both data sets A and B. We see greater agreement between the list of nodes generated by PCC score with flow data collected over a longer durations.



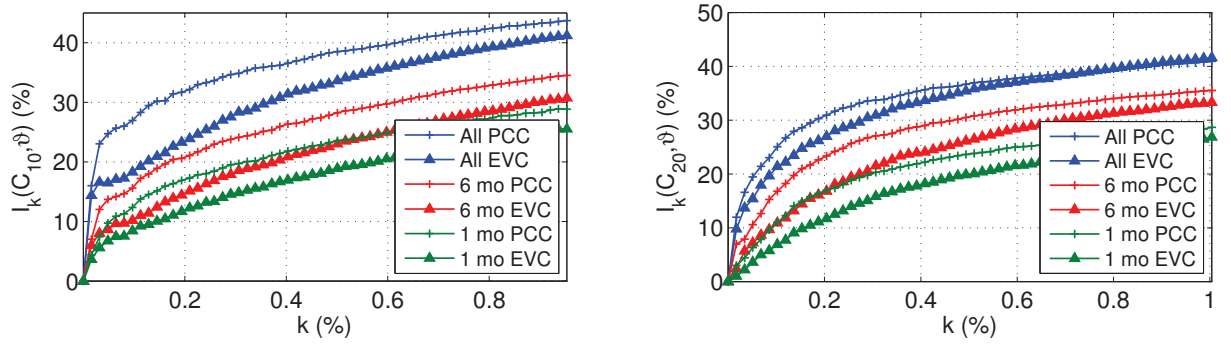
(a) Facebook data set A

(b) Facebook data set B

Figure 5.9. Size of the intersection set in (a) Facebook data set A, and (b) Facebook data set B, for varying number of eigenvectors used in computation of PCC.

Figures 5.10(a) and 5.10(b) plots I_k set for top-k users for data sets A and B, respectively. We observe an increasing trend for I_k as we increase the bracket size of top-k users. We also note that the cardinality of the intersection set increases for increasing durations of interaction data. The overlap approaches 40% of k mark for top-1% users. Moreover, we observe that the results for Facebook data set A are slightly better than those of Facebook data set B.

The evaluation described till now focuses on the number of users that are common in top-k set assembled with respect to PCC scores and node degree in directed interaction graph. In a more fine-grained analysis, we are also interested in quantifying the accuracy of ranks assigned using PCC scores. Towards this end, we compute the difference between ranks assigned by PCC and those determined using data from interaction graph. Moreover, the significance of correct ranking of high-ranked users is more important than low-ranked users.



(a) Facebook data set A

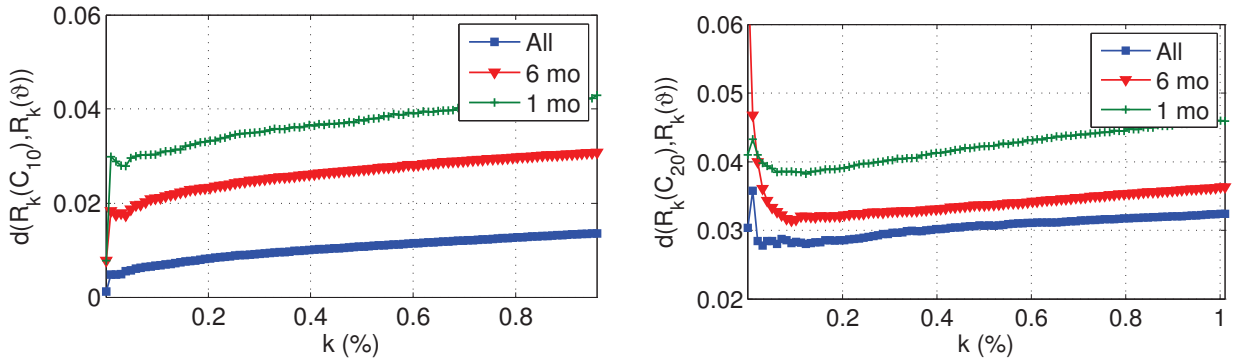
(b) Facebook data set B

Figure 5.10. Cardinality of the intersection set in (a) Facebook data set A, and (b) Facebook data set B, for varying fraction of nodes in graph.

To accomplish these objectives, we have devised a distance metric to compare the relevance of two ordered lists. We denote the list of nodes of length k in descending order of $\mathbf{C}_P$ by $\mathcal{R}_k(\mathbf{C}_P)$ and the list of nodes of length k in descending order of interaction graph degree by $\mathcal{R}_k(\vartheta)$. The distance is normalized in the range $[0, 1]$, where 0 correspond to the perfect match between two given order lists, and vice-versa. We define the normalized distance $d \in [0, 1]$ between these two ordered lists as:

$$d\left(\mathcal{R}_k(\mathbf{C}_P), \mathcal{R}_k(\vartheta)\right) = \frac{\sum_{i \in \mathcal{R}_k(\vartheta)} \left[\frac{w_i |\mathcal{R}_k(\mathbf{C}_P(i)) - \mathcal{R}_k(\vartheta(i))|}{N - 2i + 1}\right]}{\sum_{i \in \mathcal{R}_k(\vartheta)} w_i} \tag{5.10}$$

Here w_i is the degree of user i in the interaction graph and N is the total number of users. Figures 5.11(a) and 5.11(b) show the variation in distance between two ordered lists as we increase its size k for data sets A and B, respectively. Similar to the intersection results, we first note that the best results are achieved when comparison is done with interaction data of longer time duration. Second, we note that the results slightly degrade for increasing values of k. Third, it is evident that the results for Facebook data set A are better than those for Facebook data set B. For example, $d \approx 0.01$ at $k = 0.5\%$ of N for data set A, whereas $d \approx 0.03$ at $k = 0.5\%$ of N for data set B.



(a) Facebook data set A

(b) Facebook data set B

Figure 5.11. Distance between ordered lists computed by PCC and interaction data using (a) Facebook data set A, and (b) Facebook data set B, for varying fraction of nodes in graph.

5.5 Conclusions

Information hubs in social networks play important roles in the speed and depth of information diffusion. Identifying hubs helps us to harness their power to pursue social good at local, national, and global levels. In this work, we propose the first friendship graph based, fully distributed, and privacy-preserving method for identifying hubs in online social networks. Unlike prior work, our method can be used to identify hubs by parties other than social network owners as we do not require any entity to access interaction or friendship graphs. We conducted experiments using data collected from Facebook. The data sets used in this study were collected over the period of more than a year and contain data from about 6 million users. The results of our experiments using this data showed that our proposed protocol accurately (in terms of number of correctly identified top-k nodes and their estimated rank) identifies the top-k information hubs in a social network.

6 Conclusion

In this thesis, I measured and modeled various network systems using a combination of theoretical and empirical methods. Specifically, I focused on understanding various aspects of cellular networks and online social networks. For cellular networks, I first presented an approach to modeling QoE for mobile video and then characterized cellular network performance during crowded events. For online social networks, I first presented an approach to de-anonymize communication among users in an IM network and then presented a model to identify information hubs in online social networks. The primary challenge in characterizing and modeling large scale networks is "dynamics" — dynamics of user behavior, dynamics of networks, and dynamics of external factors. In this thesis, I show that we can effectively model these dynamics using machine learning and signal processing approaches to gain substantial performance benefits.

The vision of this thesis can be extended to many other similar research directions. For cellular networks, several performance optimization opportunities arise due to unintended interactions among protocols operating at different layers. For example, TCP's performance generally suffers on wireless networks due to their high link layer losses. These cross-layer inefficiencies present characterization and modeling opportunities, e.g., protocol parameter tuning, traffic routing, caching, etc. Many of these research directions are part of this thesis and subject to my ongoing work on malware detection [95, 96, 100, 101, 107], social network modeling [46–48, 89, 98], operational network performance optimization [88, 90–94, 99], and protocol vulnerability analysis [97, 103, 110].

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] 3GPP Self-Organizing Networks. http://www.3gpp.org/SON.

[2] Architecture enhancements for non-3GPP accesses. `http://www.3gpp.org/ftp/Specs/html-info/23402.htm`.

[3] Multi-Resolution Land Characterization (MRLC) consortium, National Land Cover Database 2006 (NLCD 2006). `http://www.mrlc.gov/nlcd2006.php`.

[4] Offload service. `http://www.devicescape.com/offload-service`.

[5] RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. `http://tools.ietf.org/html/rfc2046`.

[6] U.S. Geological Survey, National Elevation Dataset. `http://ned.usgs.gov/`.

[7] Yahoo! network flows data, version 1.0. Yahoo! Research Webscope Data Sets.

[8] *Facebook Advertising,* ***http://www.facebook.com/advertising/***, 2010.

[9] Statement of rights and responsibilities. *http://www.facebook.com/terms.php*, 2010.

[10] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015. White Paper, February 2011.

[11] Actix Press Release. `http://www.actix.com/sites/www.actix.com/files/Actix_Hotspots_Study_Findings.pdf`, June 2012.

[12] Wi-Fi CERTIFIED Voice-Enterprise, Delivering Wi-Fi voice to the enterprise. White Paper, May 2012.

[13] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2012–2017. Technical report, Cisco, 2013.

[14] Vaneet Aggarwal, Rittwik Jana, Kadangode Ramakrishnan, Jeffrey Pang, and N. Shankaranarayanan. Characterizing fairness for 3G wireless networks. In *IEEE LANMAN*, 2011.

[15] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. A quest for an internet video quality-of-experience metric. In *11th ACM Workshop on Hot Topics in Networks (HotNets-IX)*, 2012.

[16] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. In *ACM SIGCOMM*, 2013.

[17] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *ACM IMC*, 2009.

[18] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *Recent Advances in Intrusion Detection*, 2010.

[19] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.

[20] Roger S. Bivand, Edzer J. Pebesma, and Virgilio Gomez-Rubio. *Applied Spatial Data Analysis with R*. Springer, 2008.

[21] B. Bollobas. *Modern graph theory*. Springer Verlag, 1998.

[22] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.

[23] Phillip Bonacich. Technique for analyzing overlapping memberships. *Sociological Methodology*, 4:176–185, 1972.

[24] S.P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[25] Alessio Botta, Antonio Pescape, Giorgio Ventre, Ernst Biersack, and Stefan Rugel. Performance footprints of heavy-users in 3G networks via empirical measurement. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2010.

[26] Guido Caldarelli. *Scale-Free Networks*. Oxford University Press, 2007.

[27] G. Canright, K. Engø-Monsen, and M. Jelasity. Efficient and robust fully distributed power method with an application to link analysis. *Department of Computer Science, University of Bologna, Tech. Rep. UBLCS-2005-17*, pages 2005–17, 2005.

[28] B. Claise. Cisco systems NetFlow services export version 9. Wikipedia, the free encyclopedia, October 2004.

[29] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[30] RR Coifman and MV Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2 Part 2):713–718, 1992.

[31] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. In *ACM SIGCOMM*, 2011.

[32] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley New York, 2nd edition, 2001.

[33] Jeffrey Erman, Alexandre Gerber, K.K. Ramakrishnan, Subhabrata Sen, and Oliver Spatscheck. Over the top video: The gorilla in cellular networks. In *ACM IMC*, 2011.

[34] Pablo A. Estevez, Pablo Vera, and Kazumi Saito. Selecting the most influential nodes in social networks. In *proceedings of IJCNN*, 2007.

[35] Tom Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. Technical report, HP Laboratories, 2004.

[36] A. Finamore, M. Mellia, M. Munafo, R. Torres, and S. R. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *ACM IMC*, 2011.

[37] J. Fry, G. Xian, S. Jin, J. Dewitz, C. Homer, L. Yang, C. Barnes, N. Herold, and J. Wickham. Completion of the 2006 National Land Cover Database for the Conterminous United States. *Photogrammetric Engineering & Remote Sensing (PE&RS)*, 77(9):858–864, 2011.

[38] Duncam Geere. Samsung offers free phones to frustrated iPhone users. *CNN Tech*, http://www.cnn.com/2010/TECH/mobile/07/24/ samsung.replacing.iphones/, 2010.

[39] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From the Edge. In *ACM IMC*, 2007.

[40] Jacob Goldenberg, Sangman Han, Donald R. Lehmann, and Jae Weon Hong. The role of hubs in the adoption processes. *Journal of Marketing, American Marketing Association*, 2008.

[41] A. Goyal, F. Bonchi, and L.V.S. Lakshmanan. Discovering leaders from community actions. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, 2008.

[42] Bo Han, Pan Hui, V. S. Anil Kumar, Madhav V. Marath, Guanhong Pei, and Aravind Srinivasan. Cellular traffic offloading through opportunistic communications: A case study. In *ACM MobiCom Workshop on Challenged Networks*, 2011.

[43] Raymond Heatherly, Murat Kantarcioglu, and Bhavani M. Thuraisingham. Preventing private information inference attacks on social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2012.

[44] Junxian Huang, Q. Xu, B. Tiwana, Z. Morley Mao, Ming Zhang, and Victor Bahl. Anatomizing application performance differences on smartphones. In *ACM MobiSys*, 2010.

[45] M. U. Ilyas and H. Radha. A KLT-inspired node centrality for identifying influential neighborhoods in graphs. In *Conference on Information Sciences and Systems*, Princeton, NJ, 2010. Princeton University.

[46] 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 *30th Annual IEEE Conference on Computer Communications (INFOCOM) Mini-Conference*, 2011.

[47] Muhammad U. Ilyas, M. Zubair Shafiq, Alex X. Liu, and Hayder Radha. A Distributed Algorithm for Identifying Information Hubs in Social Networks. In *IEEE Journal on Selected Areas in Communications (JSAC)*, 2013.

[48] Muhammad Usman Ilyas, M. Zubair Shafiq, Alex X. Liu, and Hayder Radha. Who are You Talking to? Breaching Privacy in Encrypted IM Networks. In *21st IEEE International Conference on Network Protocols (ICNP)*, 2013.

[49] Hao Jiang and Constantinos Dovrolis. Passive estimation of TCP round-trip times. *SIGCOMM CCR*, 32(3), 2002.

[50] D. Kempe and F. McSherry. A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1):70–83, 2008.

[51] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *proceedings of KDD*, 2003.

[52] Masahiro Kimura, Kazumi Saito, and Ryohei Nakano. Extracting influential nodes for information diffusion on a social network. In *proceedings of AAAI*, 2007.

[53] Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda. Finding influential nodes in a social network from information diffusion data. *Springer Social Computing and Behavioral Modeling*, 2009.

[54] C. Kohlschütter, P. Chirita, and W. Nejdl. Efficient parallel computation of pagerank. *Lecture Notes in Computer Science*, 3936:241, 2006.

[55] S. Shunmuga Krishnan and Ramesh K. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *ACM IMC*, 2012.

[56] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45(4):255–282, 1950.

[57] A.N. Langville, C.D. Meyer, and P. FernÁndez. Googles pagerank and beyond: the science of search engine rankings. *The Mathematical Intelligencer*, 30(1):68–69, 2008.

[58] Stephen Lawson. Wireless networks are near capacity. `http://www.pcworld.com/businesscenter/article/235964/survey_wireless_networks_are_near_capacity.html`, July 2011. IDG News Service.

[59] Patrick P. C. Lee, Tian Bu, and Thomas Woo. On the detection of signaling DoS attacks on 3G wireless networks. In *IEEE Infocom*, 2007.

[60] R.B. Lehoucq and D.C. Sorensen. Deflation Techniques for an Implicitly Restarted Arnoldi Iteration. *SIAM Journal on Matrix Analysis and Applications*, 17:789, 1996.

[61] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[62] Xiaotong Li. Informational cascades in IT adoption. *Communications of the ACM*, 47(4), 2004.

[63] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the12th International Conference on Information and Knowledge Management*, pages 556–559, New York, NY, USA, 2003. ACM.

[64] F. Liers and A. Mitschele-Thiel. UMTS data capacity improvements employing dynamic rrc timeouts. In *16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2005.

[65] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang. Experiences in a 3G network: Interplay between the wireless channel and applications. In *ACM MobiCom*, 2008.

[66] Wei Lu and Ali A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing*, 2009.

[67] Haiyun Luo, Ramachandran Ramjeey, Prasun Sinhaz, Li (Erran) Liy, and Songwu Lu. UCAN: A unified cellular and adhoc network architecture. In *ACM MobiCom*, 2003.

[68] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[69] P.V. Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407–422, 2002.

[70] Jakub Mikians, Laszlo Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. Detecting price and search discrimination on the internet. In *HotNets*, 2012.

[71] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of IMC*, San Diego, CA, October 2007.

[72] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2010.

[73] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[74] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[75] A. M. Odlyzko. Privacy, economics, and price discrimination on the internet. In *Fifth International Conference on Electronic Commerce (ICEC)*, 2003.

[76] Barbara Orlandi and Frank Scahill. Wi-Fi Roaming – Building on ANDSF and Hotspot 2.0. Technical report, Alcatel-Lucent and BT, 2012.

[77] Eli Pariser. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think [Paperback]*. Penguin Books, 2012.

[78] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R. Das. Understanding traffic dynamics in cellular data networks. In *IEEE Infocom*, 2011.

[79] Feng Qian, Zhaoguang Wang, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. TOP: Tail optimization protocol for cellular radio resource allocation. In *IEEE ICNP*, 2010.

[80] Feng Qian, Zhaoguang Wang, Alexandre Gerber, Zhuoqing Morley Mao, Subhabrata Sen, and Oliver Spatscheck. Characterizing radio resource allocation for 3G networks. In *ACM IMC*, 2010.

[81] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1993.

[82] Ross J. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, 1992.

[83] Ashwin Rao, Yeon sup Lim, Chadi Barakat, Arnaud Legout, Don Towsley, and Walid Dabbous. Network characteristics of video streaming traffic. In *ACM CoNEXT*, 2011.

[84] B. Ruhnau. Eigenvector-centrality–a node-centrality? *Social networks*, 22(4):357–365, 2000.

[85] K. Sankaralingam, S. Sethumadhavan, and J.C. Browne. Distributed pagerank for p2p systems. In *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, pages 58–68, 2003.

[86] Raimund Schatz, Tobias Hobfeld, and Pedro Casas. Passive YouTube QoE Monitoring for ISPs. In *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012.

[87] Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N. Padmanabhan. Bartendr: A practical approach to energy-aware cellular data scheduling. In *ACM MobiCom*, 2010.

[88] 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.

[89] M. Zubair Shafiq, Muhammad U. Ilyas, Alex X. Liu, and Hayder Radha. Identifying Leaders and Followers in Online Social Networks. In *IEEE Journal on Selected Areas in Communications (JSAC)*, 2013.

[90] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, Shobha Venkataraman, and Jia Wang. A First Look at Cellular Network Performance during Crowded Events. In *ACM SIGMETRICS*, 2013.

[91] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. A First Look at Cellular Machine-to-Machine Traffic - Large Scale Measurement and Characterization. In *ACM SIGMETRICS/Performance*, 2012.

[92] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. Characterizing Geospatial Dynamics of Application Usage in a 3G Cellular Data Network. In *IEEE INFOCOM*, 2012.

[93] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. Large Scale Measurement and Characterization of Cellular Machine-to-Machine Traffic. In *IEEE/ACM Transactions on Networking (ToN)*, 2013.

[94] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. In *ACM SIGMETRICS*, 2011.

[95] M. Zubair Shafiq, Syed Ali Khayam, and Muddassar Farooq. Embedded Malware Detection using Markov n-grams. In *International Conference on Detection of Intrusions, Malware and Vulnerability Assessment (DIMVA)*, 2008.

[96] M. Zubair Shafiq, Syed Ali Khayam, and Muddassar Farooq. Improving Accuracy of Immune Inspired Malware Detectors using Intelligent Features. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2008.

[97] M. Zubair Shafiq, Franck Le, Mudhakar Srivatsa, and Alex X. Liu. Cross-Path Inference Attacks on Multipath TCP. In *Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII)*, 2013.

[98] M. Zubair Shafiq and Alex X. Liu. A Random Walk Approach to Modeling the Dynamics of the Blogosphere. In *10th IFIP/TC6 Networking*, 2011.

[99] M. Zubair Shafiq, Alex X. Liu, and Amir Khakpour. Caching in content delivery networks: Measurement, design, and evaluation. In *ACM SIGMETRICS*, 2014.

[100] M. Zubair Shafiq, Syeda Momina Tabish, and Muddassar Farooq. PE-Probe: Leveraging packer detection and structural information to detect malicious portable executables. In *Virus Bulletin Conference (VB)*, 2009.

[101] M. Zubair Shafiq, Syeda Momina Tabish, Fauzan Mirza, and Muddassar Farooq. PE-Miner: Mining Structural Information to Detect Malicious Executables in Realtime. In *12th International Symposium On Recent Advances In Intrusion Detection (RAID)*, 2009.

[102] Rumah Shahbaz. how high is your popularity? *http://www.facebook.com/apps/application.php?id=174042725891*, 2010. 12120 monthly users.

[103] Muhammad Shahzad, M. Zubair Shafiq, and Alex X. Liu. A Large Scale Exploratory Analysis of Software Vulnerability Life Cycles. In *34th International Conference on Software Engineering (ICSE)*, 2012.

[104] X. Shi, M. Bonner, L.A. Adamic, and A.C. Gilbert. The very small world of the well-connected. In *Proceedings of the 19th ACM conference on Hypertext and hypermedia (HT)*, 2008.

[105] Lingyang Song and Jia Shen, editors. *Evolved Cellular Network Planning and Optimization for UMTS and LTE*. CRC Press, 2010.

[106] N. Rama Suri and Y. Narahari. Determining the top-k nodes in social networks using the shapley value. In *proceedings of AAMAS*, 2008.

[107] Syeda Momina Tabish, M. Zubair Shafiq, and Muddassar Farooq. Malware Detection using Statistical Analysis of Byte-Level File Content. In *Workshop on CyberSecurity and Intelligence Informatics (CSI), ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[108] Carmela Troncoso and George Danezis. The Bayesian traffic analysis of mix networks. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[109] M-H. Wang V. Shmatikov. Timing analysis in low-latency mix networks: Attacks and defenses. In *European Symposium on Research in Computer Security (ESORICS)*, 2006.

[110] Yipeng Wang, Xiaochun Yun, M. Zubair Shafiq, Liyan Wang, Alex X. Liu, Zhibin Zhang, Danfeng(Daphne) Yao, Yongzheng Zhang, and Li Guo. A Semantics Aware Approach to Automated Reverse Engineering Unknown Protocols. In *20th IEEE International Conference on Network Protocols (ICNP)*, 2012.

[111] 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.

[112] Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[113] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[114] Mike P. Wittie, B. Stone-Gross, K.C. Almeroth, and E.M. Belding. MIST: Cellular data network measurement for mobile applications. In *IEEE BROADNETS*, 2007.

[115] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[116] Qiang Xu, Alexandre Gerber, Z. Morley Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *ACM IMC*, 2011.

[117] Wan-Shiou Yang, Jia-Ben Dia, Hung-Chi Cheng, and Hsing-Tzu Lin. Mining social networks for targeted advertising. In *39th Annual Hawaii International Conference on System Sciences (HICSS)*, 2006.

[118] J.-H. Yeh, J.-C. Chen, and C.-C. Lee. Comparative analysis of energy saving techniques in 3GPP and 3GPP2 systems. *IEEE Transactions on Vehicular Technology*, 58(1):432–438, 2009.

[119] Yu Zhang, Zhaoqing Wang, and Chaolun Xia. Identifying key users for targeted marketing by mining online social network. In *Advanced Information Networking and Applications Workshops (WAINA)*, 2010.

[120] E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *World Wide Web (WWW) Conference*, 2009.

[121] Elena Zheleva and Lise Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International World Wide Web conference (WWW)*, 2009.

[122] Y. Zhu, Xinwen Fu, R. Bettati, and Wei Zhao. Anonymity analysis of mix networks against flow-correlation attacks. In *IEEE Global Communications Conference (GLOBECOM)*, 2005.

[123] Ye Zhu, Xinwen Fu, Bryan Gramham, Riccardo Bettati, and Wei Zhao. Correlation-based traffic analysis attacks on anonymity networks. *IEEE Transactions on Parallel and Distributed Systems*, 2009.

[124] Feng Zou, Zhao Zhang, and Weili Wu. Latency-bounded minimum influential node selection in social networks. In *proceedings of WASA*, 2009.