Traffic Driven Analysis of Cellular and WiFi Networks

A Dissertation Presented
by

Utpal Kumar Paul

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

December 2012

Stony Brook University
The Graduate School

Utpal Kumar Paul

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dr. Samir R. Das, Dissertation Advisor
Professor, Department of Computer Science

Dr. Luis Ortiz, Chairperson of Defense
Assistant Professor, Department of Computer Science

Dr. Himanshu Gupta, Committee Member
Associate Professor, Department of Computer Science

Dr. Milind Buddhikot, External Committee Member
Distinguished Member of Technical Staff,
Alcatel-Lucent Bell Laboratories

This dissertation is accepted by the Graduate School

Charles Taber
Interim Dean of the Graduate School

Abstract of the Dissertation

Traffic Driven Analysis of Cellular and WiFi Networks

by

Utpal Kumar Paul

Doctor of Philosophy

in

Computer Science

Stony Brook University

2012

Since the days Internet traffic proliferated, measurement, monitoring and analysis of network traffic have been critical to not only the basic understanding of large networks, but also to seek improvements in resource management, traffic engineering and security. At the current times traffic in wireless local and wide area networks are facing similar upsurge. This calls for a similar vigor in traffic analysis studies. This thesis focuses on several traffic analysis studies in both cellular data networks and WiFi LANs. The broad goal is (i) to improve the understanding of the traffic dynamics, to explore structures in the traffic to help cost-effective monitoring and building of new traffic management strategies — in the context of cellular networks, and (ii) understanding the interference properties and detecting misbehavior — in the context of WiFi networks.

We first use a large-scale data set collected inside a nationwide 3G cellular data network and conduct a detailed measurement analysis of network resource usage and subscriber behavior. We characterize subscriber mobility and temporal activity patterns and identify their relation to traffic volume. We also investigate how efficiently radio resources are used by different subscribers as well as by different applications. Our analysis using different statistical techniques shows existence of significant spatial correlation in radio resource usage in the base stations. We also use the concept of Granger Causality to understand the underlying functional connectivity and flow of influence in the network. Broadly, our observations deliver important insights into network-wide resource usage.

Next, we propose a new traffic management technique for cellular data networks to improve networks' resource crisis situation in the face of exponential increase in mobile data traffic volume. Here we consider the existence of a higher-layer, agent-based scheduling system that could potentially delay scheduling of low priority flows at peak loads. The priorities are assumed to be user or application tagged, either automatically or manually. The general goal is to potentially move the low priority flows in time and space opportunistically to reduce the overall resource needs. We develop and evaluate two scheduling schemes. Simulation results using our large-scale cellular network trace data show the potential of these approaches in reducing base station resource requirements.

Next, we present a scalable traffic measurement and monitoring technique for cellular data networks. We use a machine learning technique to learn the underlying conditional dependence structure in the base station traffic loads to show how such probabilistic models can be used to reduce the traffic monitoring efforts. The broad goal is to exploit the model to develop a spatial sampling technique that estimates the loads on all the base stations based on actual measurements only on a small subset of base stations. To understand the tradeoff between the accuracy and monitoring complexity better, we also study the use of this modeling approach on real applications. Two applications are studied – energy saving and opportunistic scheduling. They show that load estimation via such modeling is quite effective in reducing the monitoring burden.

In the last part of our thesis, we turn our attention to WiFi networks. We present a tool to estimate the interference between nodes and links in a live WiFi network by passive monitoring of wireless traffic. Our approach requires deploying multiple sniffers across the network to capture wireless traffic traces. These traces are then analyzed using a machine learning approach to infer the carrier-sense relationship between network nodes. We also demonstrate an important application of this tool-detection of selfish carrier-sense behavior. This is based on identifying any asymmetry in carrier-sense behavior between node pairs and finding multiple witnesses to raise confidence. We evaluate the tool using extensive experiments and simulation which demonstrate the effectiveness of both the applications.

Dedicated to Ma, Babu and Prachi.

Contents

List of Figures

Chapter 1

Introduction

For the last two decades, traffic measurement and analysis have been one of the major fields of research in the networking community. Specifically, the Internet has gained a lot from this understanding. A lot has been learnt about traffic and connectivity properties of the Internet [27, 33, 57, 137], benefiting the design of network architectures, protocols and devices such as routers and switches. Moreover, a wide range of important problems including traffic engineering, anomaly detection, attack detection, traffic forecasting and capacity planning have been addressed. This has been critical in handling the traffic surge in the Internet. Broadband cellular networks and WiFi networks are currently experiencing similar traffic surges and can benefit from similar vigor in measurement and analysis. In this dissertation, our focus is on analyzing cellular data networks and WiFi networks by conducting extensive traffic measurement, analysis and modeling.

Cellular networks have become one of the most popular means of communication and data access now-a-days. Traditionally, cellular networks were used for voice-based communication and they were designed primarily to support constant rate voice-based traffic. Recently there is a substantial increase in speed and capacity in 3G networks. For example, GSM [19] based HSPA WCDMA based HSPA (High Speed Packet Access) networks and CDMA [18] based EVDO (Evolution-Data Optimized) networks are quite common. Higher capacity networks such as LTE [12] and WiMax [56] are also emerging. Out of 6.8 billion world population, the number of mobile phone subscriptions has crossed the 5 billion mark in 2010, roughly 1 billion of them already broadband capable [14]. A new ecosystem of devices (e.g., smart-phones, e-readers and tablets), software (apps) and services (e.g., cloud computing) is fueling the growth of mobile data. Predictions from industry analysts indicate that the volume of data through cellular data networks will increase exponentially in near future [13, 6]. In terms of numbers, these forecasts suggest that cellular data traffic volume will reach several exabytes per month by 2014 (1 exabyte = 1 million terabytes), roughly equaling the traffic volume in the entire global Internet back in 2006. By any means all these are stupendous numbers. But the capacity of the network is not increasing at the same rate, even with the introduction of the latest technologies.

While operators are scrambling to add capacity, there is an apparent lack of understanding of the nature of mobile traffic in the large scale. Most of the prior studies on cellular data network characterize network performance and capacity based on small scale measurements done in the client side of the network. For example, Joyce *et al.* [75] have presented single cell and network capacity measurements using a commercial network in UK. Yao *et al.* [141] have evaluated bandwidth predictability for HSDPA networks. Tan *et al.* [132] have studied the capacity of 3G networks in terms of throughput, latency, video and voice call handling ability. These studies capture the performance of the network from a narrow perspective. To understand the network usage pattern and subscriber behavior from a global perspective, a large scale comprehensive analysis of an operational 3G data network needs to be done. Though there are few such analyses in literature based on voice calls [138], cellular data networks lack such detailed investigation. Such understanding can give answers to different questions regarding subscriber traffic dynamics, spatial and temporal behavior of network resource usage. It is only recently that a limited number of papers [81, 112, 105] had made an attempt to understand the global view of the network. Researchers have also proposed several analysis approaches that are focused on specific applications, such as energy saving [59] or network trouble shooting [74].

1.1 Focus of the Dissertation

In this dissertation, we focus on understanding traffic dynamics in cellular data networks both from spatial and temporal perspectives. We posit that understanding of the mobile data traffic via measurement and analysis is critical for resource management and future development of wireless broadband access networks, just like the Internet measurement research. Our goal is to discover any possible structure or relationships in the network so that we can develop a comprehensive knowledge of resource usage in cellular data networks. Such understanding will bring new insights that in turn will help to deploy and manage future generation cellular data networks.

Instead of just limiting our focus in understanding the dynamics of the network and subscribers, we also investigate how to improve the user perceived performance in the network and identify possible and efficient mechanism to cope with the exponential growth of mobile data usage. This is imperative as adding resources and increasing capacity in the network with the same pace may not be cost effective. We consider this as an immediate challenge to the capacity of the cellular data networks and develop a traffic management scheme as an alternative to address the problem, as oppoesed to plainly increasing capacity in the network. The current traffic outburst in cellular data networks poses another issue for the service providers. It is now a complex problem to do real time measurement and monitoring of the network because of its large volume and size. Thus we also focus on designing scalable traffic measurement and monitoring techniques that can aid the service providers in efficient resource management.

Wireless Local Area Networks (WLANs) also have become commonplace, especially on university and corporate campuses, and increasingly in public WiFi hotspots as well. Even with the presence of cellular data networks, WiFi is often preferred because of license-free operation that makes the access costs much lower. WiFi performance is also typically much superior, especially in the indoor settings. This is why most of the mobile devices including smart phones, tablets are equipped with network adapters that can access one or more types of IEEE 802.11 network. Understanding usage patterns in WiFi networks is critical for those who develop, deploy, and manage WLAN technology, as well as those who develop systems and application software for wireless networks. There has been a large body of works focusing on characterizing the usage patterns and performance in live WiFi networks by collecting and analyzing network traces. Representative sutdies in this space includes [35, 94, 85, 68]. There are also a number of measurement studies for the purpose of learning various properties of live network such as congestion [73], protocol behavior in a hotspot setting [122], etc. The DAIR system also proposes mechanisms for troubleshooting [22] and security [23].

But still there is lot to learn and investigate in WiFi networks. Specifically, understanding interference in WiFi networks is still an important problem. Poor WiFi performance is often attributed to wireless interference in highly loaded networking scenarios. While a lot of research has been conducted in understanding wireless interference in a theoretical context, real network deployments are yet to gain from it. Learning the interference relationship among network elements can help the system managers to perform capacity planning and appropriate radio resource management, such as assignment of channels, transmit power levels or directions when using directional antennas. But this should be done in the most unobtrusive fashion possible — without disturbing the live network and doing any active measurement. In this dissertation, our focus is to develop an approach to estimate the interference between nodes and links in a live wireless network by passive monitoring of wireless traffic.

Another important aspect of WiFi networks that has not been adequately addressed is the possible ways of 'cheating' in the protocol design. There are nearly not enough effort to police the spectrum to ensure that all radio devices follow a prescribed protocol or rule and to develop techniques so that selfish behavior can be detected. With the advent of programmability in radios, it is becoming easier for wireless network nodes to cheat to obtain an unfair share of the bandwidth by manipulating different MAC layer protocol parameters in different ways. In this dissertation, we focus on designing a solution to detect one such behavior, selfish carrier-sensing behavior where a node raises the CCA (clear channel assessment) threshold for carrier sensing, or simply does not sense carrier (possibly randomly to avoid detection). Just like the previous case of learning interference relationship in the network, we like to develop the detection mechanism in a completely passive manner.

1.2 Contributions

1.2.1 Understanding Traffic in Cellular Data Networks

In this dissertation, we present one of the first large scale studies of cellular data networks. Our goal here is to develop an understanding of the nature of the spatio-temporal traffic dynamics in cellular data networks by developing models from large-scale measurement data collected from a nation-wide network. We analyze individual subscriber behavior and base station characteristic in terms of traffic generation and total load, and observe significant asymmetry. We characterize subscriber mobility and temporal activity patterns and identify their relation to traffic volume. We observe significant asymmetry in resource usage among subscribers and applications. We also analyze the network traffic from the of view of the base stations and find significant temporal and spatial variations in different parts of the network, while the aggregated behavior appears predictable. These observations can be important for the service providers in efficient protocol design, resource and spectrum management as well as better pricing plan.

We also extend our focus to understand the spatial characteristics in resource usage in cellular data networks. We use a broad range of state-of-the-art tools for modeling, learning, clustering, predicting and forecasting developed within the statistical machine learning community. We observe the existence of significant spatial correlation in resource usage among the base stations using different statistical techniques in the network. We then use this relation to cluster the base stations. Such clustering should help the network provider in resource planning, as the provider now can think in terms of clusters or groups instead of individual base stations. We use the concept of Granger causality [62] to understand the underlying functional connectivity and flow of influence in the network. This understanding can be helpful in predicting the base stations loads, and thus allocating the spectrum accordingly in advance.

1.2.2 Traffic Management and Monitoring in Cellular Data Networks

Accommodating the exponential increase in traffic volume is now becoming a challenge for the service providers as increasing network capacity involves major capital investment in terms of new spectrally efficient technology, additional spectrum and/or additional base stations. This is why new traffic management techniques are needed to improve network and user perceived performance. We consider prioritizing traffic flows and propose a higher-layer, agent based scheduling system that could potentially delay scheduling of low priority flows at peak loads. The general goal is to potentially move the low priority flows in time and space opportunistically to reduce the overall resource needs. We develop and evaluate two scheduling schemes — one based on a straightforward greedy method that requires real-time load monitoring and the other based on model-based estimation of future traffic loads and subscriber mobility based on historical data. The ultimate goal of this approach is to accommodate a significant number of additional subscribers in the same network without expending any additional resource.

On the other hand, the exponential growth of traffic volume is not only raising the capacity issue, but also creating problems for the network providers for real time monitoring of the network. Given this situation, it is imperative that scalable traffic measurement and monitoring techniques be developed to aid various resource management methods. We approach this problem from modeling perspective. We learn the underlying dependency relationship of load among the base stations in the network using a machine learning technique. The idea is to exploit the model to develop a spatial sampling technique that estimates the loads on all the base stations based on actual measurements only on a small subset of base stations. This approach has the similarity in typical network monitoring technique by sampling packets or flows to reduce the overhead. But here we use the concept of sampling at a higher level in the context of spatially separated entities like base stations to make it scalable.

1.2.3 Interference Relationship and Selfish Behavior in WiFi Networks

Deconstructing interference relationship in WiFi network is critical for the network administrator to improve network performance. Several theoretical works and techniques requiring active measurement have been proposed in literature. But the concern here is to understand the interference relationship among network nodes and links in a live network in the most unobtrusive fashion possible. We propose an approach of deploying a set of sniffers in the network and collecting network trace. These traces are then analyzed using a machine learning technique to infer the sender-side interference. This coupled with an estimation of collision probabilities helps us to deduce the interference relationships.

We observe that this interference relationship can be used to identify a certain type of selfish behavior where a selfish node in the network does not carrier-sense proporly. This is based on identifying any asymmetry in carrier-sense behavior (sender-side interference) between node pairs and finding multiple witnesses to raise confidence. Here the asymmetry in carrier-sense behavior means that between a given pair of nodes, while one node can sense the transmission of the other node, the converse is not true. We also propose two simple heuristics to choose proper witness nodes while identifying a selfish node. The network administrator can use our approach as a toolbox with two important applications: understanding the interference properties, and detecting selfish behavior in an arbitrary WiFi network, regardless of the topology or architecture.

1.3 Outline

The rest of the dissertation is organized as follows. First, we present our analysis to understand the traffic dynamics both from network and subscriber perspective in cellular data network in Chapter 2. Then we continued our analysis with specific focus only on understanding the spatial relationship in resource usage in the cellular data network in Chapter 3. Then we shift our focus on designing efficient resource management and monitoring techniques in the network. In Chapter 4, we propose and evaluate a new traffic management scheme to improve the user perceived performance and reduce resource requirement in the network. In Chapter 5, we present an efficient and scalable monitoring approach in the cellular data network by learning the underlying probabilistic model of the network. In Chapter 6, we propose a tool of passive monitoring of WiFi network with two applications of understanding the interference properties, and detecting selfish behavior in an arbitrary WiFi network. Finally, we conclude in Chapter 7.

Chapter 2

Traffic Dynamics in Cellular Data Networks

2.1 Introduction

Cellular networks have become one of the most popular means of communication and data access now-a-days because of their wide spread coverage and availability. As mentioned in Chapter 1, the volume of data through cellular data networks is increasing exponentially. In order to support such increases, it is important to understand the traffic dynamics and its impact on resource allocation on the service provider's network. This will lead to better resource planning and network designs that finally benefit the end users.

There have been several works in the past that study spectrum usage and application characteristics in cellular data networks (see, e.g, [110, 119, 95]). Most of these prior studies try to understand wireless spectrum usage and characterize network performance and capacity using small scale measurements using a few mobile clients. To understand the network usage pattern and subscriber behavior, a large scale comprehensive measurement and analysis of network-wide data traffic must be performed. Though there have been a few studies recently based on network-wide data collected 'in-network' such as [138, 81], these studies consider voice traffic [138] or users' browsing behavior [81]. A detailed network-wide study of data traffic is still lacking.

Our focus in this chapter is to address this limitation and provide a measurement-driven analysis of the data traffic collected *at the core* of a nation-wide 3G network. Our goal is to provide answers to important questions regarding subscriber traffic patterns, subscriber mobility, and spatio-temporal behavior of network resource usage. The answers to such questions are very important from the network providers' perspective in case of resource management specifically spectrum allocation and network design.

Our study in this chapter is organized as follows: We present the network architecture and data collection procedure in Section 2.2. We focus on the spatial and temporal dynamics of data traffic from both the subscriber's (Section 2.3) and the network's (Section 2.4) perspectives. We examine individual subscriber behavior and usage patterns. We also characterize subscriber mobility and temporal activity patterns, and analyze their relationships to subscriber traffic. From the network's perspective, we study traffic patterns at different parts of the network (base stations) and understand spatial and temporal dynamics. Finally, we describe the implications of our observations related to traffic spread, mobility and efficiency in connection to subscriber pricing, protocol design, spectrum allocation and energy savings (Section 2.5).

2.2 Network Architecture and Data Collection

The CDMA based cellular data network architecture standardized by the 3GPP2 [18] standardization body is shown in Figure 1. The mobile station (MS) functions as the IP client device that a mobile user (*subscriber*) uses to connect to the Internet through the cellular data network. The subscriber accesses the data network through the Base Transceiver Station (BTS) (also called *base station*) using a wireless channel allocated for communication. The Base Station Controller (BSC) or Radio Network Controller (RNC) controls allocation of radio resources for each subscriber and routes packets to and from the subscriber to the core network. Both the BTS and BSC (or RNC) constitute the Radio Access Network (RAN). The Packet Data Serving Node

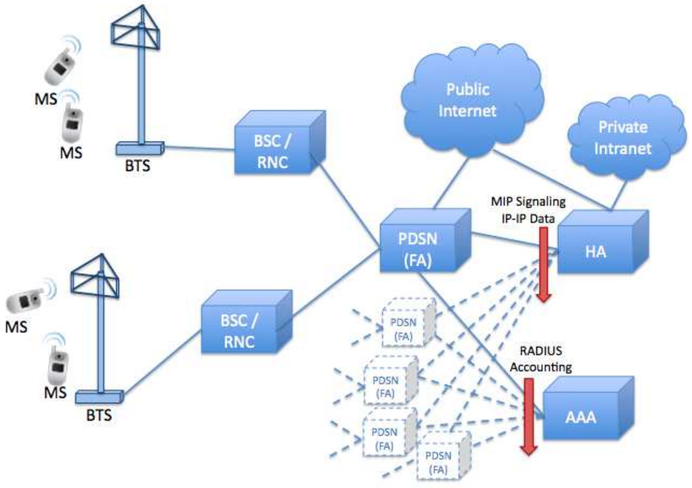


Figure 1: Architecture of a CDMA data network.

(PDSN) acts as the gateway from the RAN to the public or private packet network. In a simple IP network, it directly acts as the gateway to the public Internet. In the case of mobile IP network, it can be configured as a Home Agent (HA) or a foreign agent (FA). The Home Agent serves as the IP anchor for the subscriber and forwards any data from the public internet to the subscriber through the appropriate network elements. It also provides seamless mobility for each subscriber inside the cellular data network. The AAA (Authentication, Authorization and Accounting) server is used to authenticate and authorize each subscriber for network access and to store data usage statistics.

The data set used in this work spans one week in 2007 and consists of all data traffic associated with the entire subscriber base (in the order of hundreds of thousands) in a nation-wide network with thousands of base stations. It is collected from the core of the cellular data network by tapping at the link between the FA and HA and at the link between FA and AAA server. The red color arrows in Figure 1 show the points where packets are captured. All data packet and various signalling and accounting packets are captured and later post processed using a tool. [1] The packet capture device uses general purpose multi-core computer with four high speed endace packet capture cards [9] that can capture packets at a maximum speed of 4 Gbps. By tapping south of HA, the IP-IP data packets that are tunneled between the HA and FA as well as the mobile IP (MIP) signaling packets are obtained. Similarly, RADIUS accounting packets are collected by tapping at the link between FA and AAA server. The captured MIP and RADIUS signaling packets are used to determine when a mobile connects and disconnects to the network and when it hands off between different base stations. Our entire packet capture only logs the packet headers and not the user payload.

Signaling and data packets are associated to create subscriber sessions. A *subscriber session* for each subscriber[2] is defined as the time between subscriber mobile IP registration and de-registration with a HA. A subscriber session is identified by a tuple that consists of the subscriber IP address and the IP address of the HA it is connected to. Within each subscriber session, the data traffic for each subscriber is accumulated as *flow* records. A flow is defined by a five tuple consisting for the source IP, destination IP, source port, destination port and protocol. For TCP flows, the flow starts when a TCP SYN packet is received and ends when a TCP FIN packet is received. For UDP flows, the first packet of the flow starts the flow and the flow ends when it does not get any packet for 30 minutes. This is because UDP flows do not have any explicit packets to identify termination of the flow.

In CDMA networks, a subscriber requests and is in turn allocated a radio channel whenever it has data to send. The allocated radio channel is revoked by the network when the subscriber is dormant for certain period known as the dormancy period (typically about 10 seconds) that is configurable for different networks [18]. A subscriber can go between active (with a channel allocated) and dormant state multiple times within a single mobile IP session. We call the amount of time a subscriber holds onto a radio channel (regardless of whether

[1] For proprietary reasons, we are unable to provide further details about the network location, data set, packet capture and post-processing techniques. This is not unusual in recent published network-wide studies [138]. In any case, the missing details are not relevant to understanding our analysis for commercially operated networks.

[2] The terms ‘client’, ‘user’ and ’subscriber’ are used interchangeably in this dissertation.

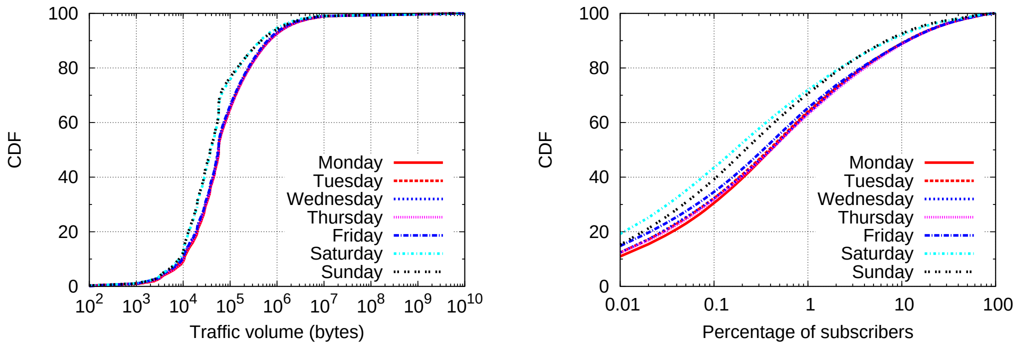
it actually communicates) as the *airtime*. Effectively, the airtime gives us the amount of time a subscriber uses network radio resources. This will play a role in our later analysis. For studying base station loads, all sectors in that base station are combined. Finally, the signaling packets we capture provide enough information for us to track base station and the cell sector the mobile is associated to at all time instants. This provides us with a rich data set to study subscriber mobility. In this aspect, our data set is far richer than that used in some related literature [61, 138].

2.3 Subscriber Traffic Dynamics

We study the behavior of mobile subscribers in terms of the traffic they generate, their mobility and their activity on the temporal scale. We draw relations between traffic generated by subscribers to their mobility as well as activity level. Finally, we present important implications of subscriber traffic dynamics on resource planning in cellular data networks.

2.3.1 Subscriber Traffic Distribution

We start with analyzing the amount of traffic generated by subscribers in the network. Figure 2(a) shows the cumulative distribution function (CDF) of traffic generated per subscriber. Each curve represents data for one day in a week. This figure shows a wide range of traffic generated by different subscribers in the network. The median traffic generated is close to 100 KB per day. However, there are heavy users who generate as high as 10 GB per day (10^5 × median) as well as light users generating less than 1 KB per day. We see the CDF slightly shifted towards the left for weekends (Saturday and Sunday) indicating less traffic relative to working days. We also present a normalized view of traffic over a percentage of subscribers in Figure 2 (b). It is interesting to see that only 1% of the subscribers (out of approximately about 500K unique subscribers who appear in each day) create more than 60% of the daily network traffic and less than 10% of the subscribers create 90%



(a) (b)

Figure 2: (a) CDF of total traffic volume (in bytes) per subscriber per day. (b) CDF of normalized traffic over the percentage of subscriber per day.

of the daily network traffic. *This points to a significant imbalance of network usage among subscribers with few subscribers hogging the much of the network resource.* Later, we will pay specific attention to mobility and network activity of these subscribers.

2.3.2 Subscriber Mobility

In our data set, we do not have access to precise location of subscribers. Our captured data set also does not have signal strength related information (as we capture packets at the IP layer in the core network) to help in radio localization. However, the signaling packets we capture provide enough information for us to track base station and the cell sector the mobile is associated to at all time instants. This provides us with a rich data set to study subscriber mobility based on the timestamped sequence of the base station he/she is connected to. We have this data in all times instants regardless of whether the subscriber is actually communicating. In this aspect, our data set is far richer than that used in some related literature [61, 138].

2.3.2.1 Base Stations Visited

Figure 3 shows the CDF of the number of distinct base stations visited by each subscriber in a day. Note that the distribution is very similar in the

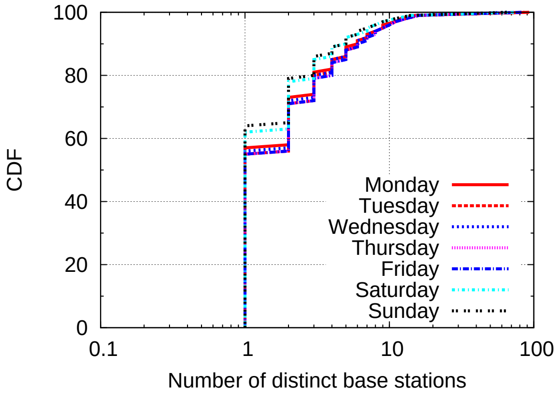
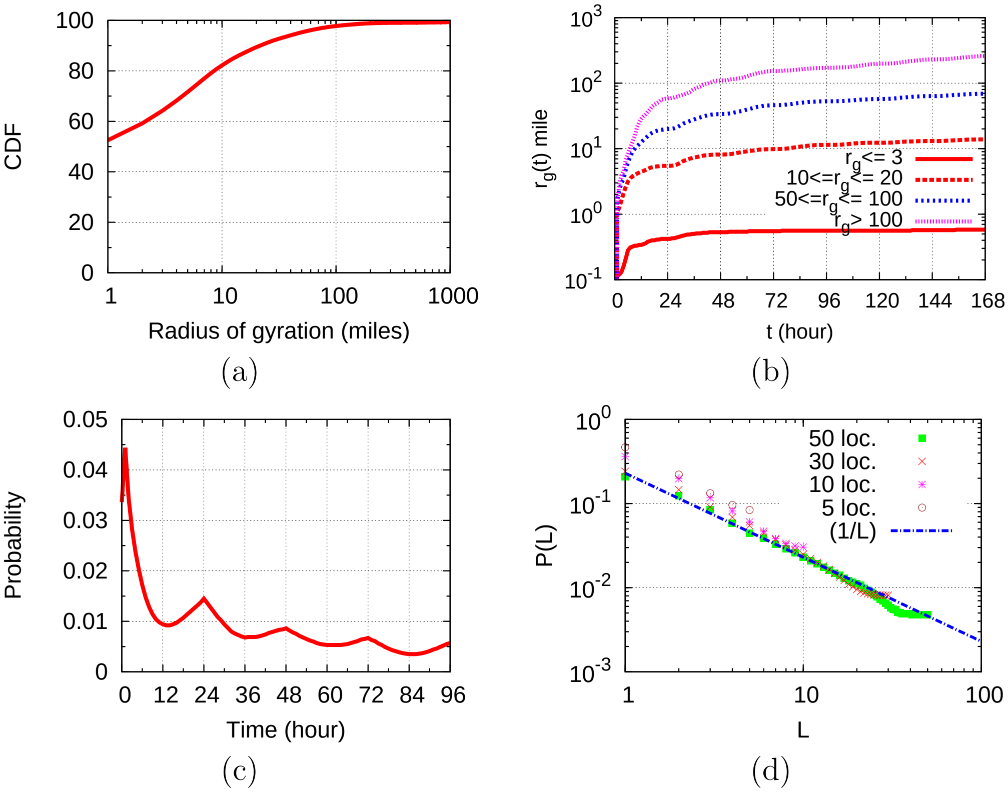


Figure 3: CDF of the number of distinct base stations visited by a subscriber in each day.

weekdays, while the distribution in the weekends is somewhat different. Note the tendency of a lesser degree of mobility on weekends. *Overall, the mobility is low in terms of the number of distinct base stations visited.* Roughly, 60% of the users are mostly stationary (i.e., constrained within a cell) and over 95% of the users travel across less than 10 base stations in a day. On the other hand, the highest number of distinct base stations visited by a user in a day is 93. However, such highly mobile users who visit more than 50 distinct base stations in a day are very few, less than 0.01% of daily users. To understand the mobility of subscribers further, we study the extent of the distance they travel next.

2.3.2.2 Radius of Gyration

The above data only captures the number of base stations visited, but not the physical extent of travel. To capture physical distance traveled we use a concept called the *radius of gyration* [61]. The radius of gyration is the linear size occupied by a subscriber's trajectory. It is computed by averaging the displacement of the recorded locations of the subscriber from a central point. The central point is the center of mass of the entire trajectory. Note that this captures how widely the subscribers move as opposed to the actual distance traveled. For example, traveling in a circle continuously visiting the same



(a) (b) (c) (d)

Figure 4: (a) CDF of radius of gyration (r_g). (b) Radius of gyration versus duration of computation for subscribers categorized into 4 groups according to their final r_g at the end of the seven-day period. (c) Probability distribution of time to returning to the same location. (d) A Zipf distribution showing the probability of finding a subscriber at different locations that are ranked on the basis of their visit frequencies. The subscribers are categorized in terms of how many distinct locations they visit during the seven-day period.

sequence of base stations does not increase the radius of gyration but a long distance travel on a straight line does. Radius of gyration has traditionally been used to study human mobility, as in a recent influential study [61]. The radius of gyration [61] is defined as,

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{r}_i - \overrightarrow{r}_{cm})^2}, \qquad (1)$$

where $\overrightarrow{r}_i$ represents the $i = 1, 2, \ldots, n$ locations recorded for a given user describing his/her trajectory. Recall that the locations are simply the locations of the base stations to which the mobile is connected. $\overrightarrow{r}_{cm} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{r}_i$ is the center of mass point of the user's trajectory.

Figure 4(a) shows the CDF of r_g, where r_g is calculated for each subscriber for the entire 7 day time period. We see approximately 53% of subscribers are practically static and almost 98% of subscribers have radius of gyration less than 100 miles. This reasserts the *low level of mobility for the majority of subscribers.* The probability distribution function of subscriber mobility represented by the radius of gyration can be well approximated with a truncated power-law :

$$P(r_g) = (r_g + r_g^0)^{-\beta_r} \exp(-r_g/\kappa), \tag{2}$$

with $r_g^0 = 2.8$ mile, $\beta_r = 1.7$ and $\kappa = 170$ mile. We note that a similar qualitative trend was observed in [61].

Note that the radius of gyration computation requires use of certain duration of time (t) during which the subscriber trajectory is used for the computation. It is expected that the longer the duration t the larger is the radius of gyration $r_g(t)$. A saturation would indicate that some sort of boundary of the movement area has been reached. To study this, we plot 'average' $r_g(t)$ with increasing t until the entire seven-day period (168 hours) is exhausted [61]. Subscribers are categorized into four different groups based on their final r_g value at the end of the seven-day period. See Figure 4(b). Note that the radius of gyration on average comes to a saturation point relatively quickly, in just a few days. Also, users with larger radius of gyration need longer time to saturate.

We further investigate the reason for the quick saturation of the radius of gyration by measuring the 'return probability' for each subscriber as the probability that a subscriber returns after t hours to the same position [61]. Figure 4 (c) shows the distribution having relative peaks at 24th, 48th and 72nd hours. *It indicates the periodic nature of human mobility with a 24-hour period and tendency of returning to the same location periodically.* This is also the inherent reason for radius of gyration saturating after a few days.

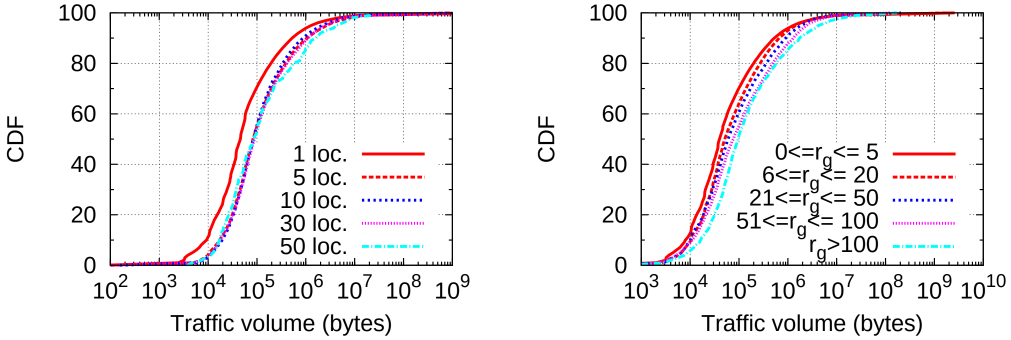
To understand how predictable the subscriber location is, we rank each location a subscriber visits on the basis of the number of times he/she is found there [61]. For example, a location with rank $L = 1$ indicates the most-visited location of the selected subscriber. For each subscriber we create the list of locations where he/she is found in the ascending order of the rank. Figure 4 (d) is the Zipf distribution showing the probability distribution of the visit frequency of locations ranked L. The figure shows the results for four categories of subscribers that visit 5, 10, 30 or 50 distinct locations. It also shows that the distribution can be well approximated by $\sim \frac{1}{L}$ irrespective of the category. Note also that people spend roughly 30% of their time in their top two preferred locations. This clearly shows that *even when subscribers move between multiple locations, they can be found in their 'favorite' location with high probability.*

2.3.3 Relating Subscriber Mobility and Traffic

Now, a natural question is to relate the subscriber mobility and the volume of traffic they generate. We categorize subscribers based on the two mobility metrics used in the previous section: (i) number of locations (base stations) visited and (ii) radius of gyration. This simply categorizes subscribers based on their degree of mobility. For each category of subscribers, we plot the CDF of traffic volume generated per day in Figure 5. A careful reader will note that while the plot lines appear similar, due to the log-scale of the horizontal axis, there is actually significant difference in traffic volume for different categories. *The trend is that more mobile subscribers generate more traffic, with the median traffic generated by subscribers in the highest mobility category being roughly twice that of the subscribers in the lowest mobility category.* This correlation of mobility and traffic has implications in resource planning and spectrum management. While our current results describe only aggregated behavior, our future work will consider finer grain behavior based on timings of movement and timings of traffic generated.

2.3.4 Subscriber Temporal Activity

We describe the temporal activity of subscribers by the number of days in a week or number of hours in a day that they generate traffic. This addresses



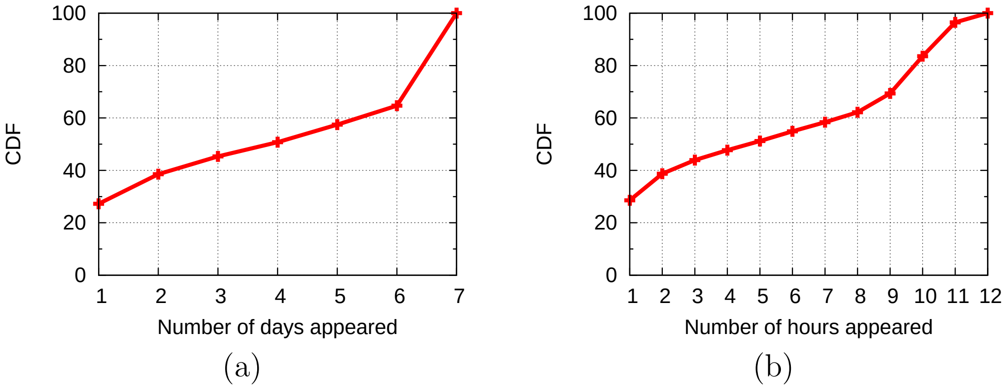
(a) (b)

Figure 5: (a) CDF of traffic generated per day by subscribers of different category based on number of locations (base stations) visited in a day. (b) CDF of traffic generated per day by subscribers of different category based on radius of gyration.

basic questions such as whether the subscribers generate traffic frequently or only occasionally. Figure 6(a) shows the CDF of the number of days subscribers generate traffic. We see that about 34% of the subscribers generate traffic on all 7 days of a week. It is interesting to note that about 45% of total number of subscribers generate traffic only on three or less number of days in a week. To understand the hourly activity of subscribers, we plot the distribution of hours among peak hours (8 AM to 8 PM) in a work day (i.e., Mon-Fri) the subscribers generate traffic. Figure 6(b) shows about 28% of subscribers generate traffic only in a single hour among this set of peak hours. A typical subscriber (median) is active in 4 different hours during the peak hours in a day. The high level conclusion here is that *a large fraction of subscribers generate traffic only in few days a week and only in a few hours within the day.*

To understand the temporal activity of subscribers at a much finer granularity, we study the distribution of 'airtime' used by each subscriber. This term requires some explanation. In the commonly used 3G standards (3GPP or 3GPP2), a subscriber requests and is in turn allocated a radio channel[3] whenever it has data to send. The allocated radio channel is revoked by the

[3] We use the term 'radio channel' to refer to any radio resource allocated to the mobile such as code, frequency or time slot.



(a)

(b)

Figure 6: (a) CDF of number of days in a week subscribers generate traffic. (b) CDF of number of hours among peak hours (8 AM to 8 PM) subscribers generate traffic.

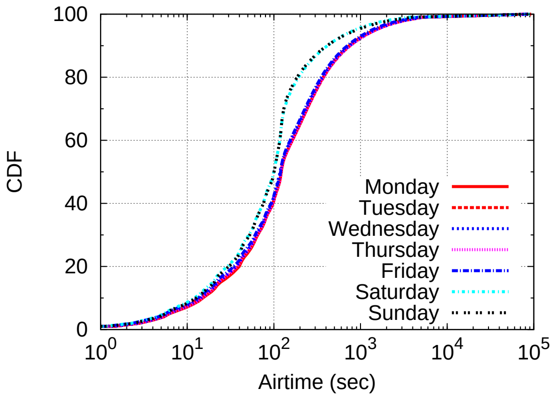
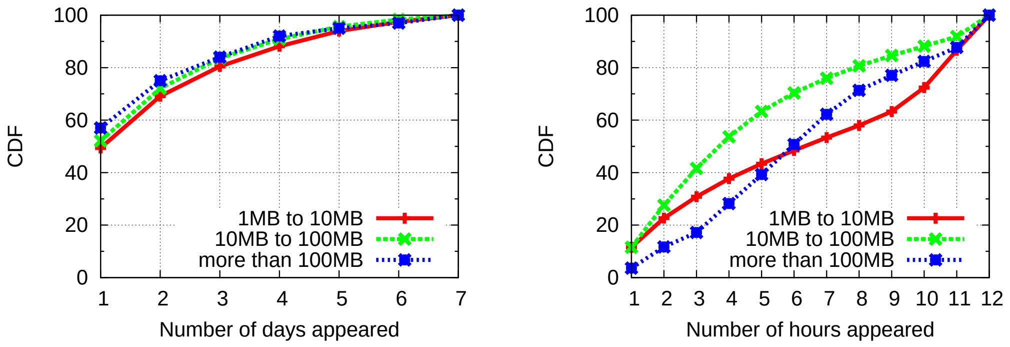


Figure 7: CDF of airtime among subscribers.

network when the subscriber is dormant for certain period known as the dormancy period (typically about 10 seconds) [39] that is configurable for different networks. A subscriber can go between active (with a channel allocated) and dormant state multiple times within a single mobile IP session. We refer the amount of time a subscriber holds onto a radio channel (regardless of whether it actually communicates) as the *airtime*. Effectively, the airtime gives us the amount of time a subscriber uses radio and spectrum resources.

Figure 7 shows the CDF of airtime among all subscribers. We see a significant variation in the amount of airtime used by different subscribers.



(a) (b)

Figure 8: (a) CDF of occurrence for the heavy users in days in a week. (b) CDF of occurrence for the heavy users in hours among peak hours.

The median usage is about 100 sec. in a day. Interestingly, there are few subscribers (less than 1%) that use almost 24 hours of airtime in a day. About 90% of subscribers use less than 1000 sec. of airtime. The median is about 100 sec. Weekend usage is typically lower compared to weekday usage. *In general, we see that a typical subscriber occupies the radio channel only for a short duration in the entire day. This is consistent with our previous observation that the median traffic volume per subscriber per day is not significant while there are a small number of 'heavy hitters' that consume a significant amount of network resource.* Such statistics can help providers develop effective pricing structures.

2.3.5 Relating Subscriber Activity and Traffic

In this section, we draw relation between the traffic generated by subscribers and how frequently they appear in the trace. We particularly focus on the 'heavy users' as they are the ones that transmit bulk of the traffic. Here, the heavy users are the subset of subscribers that are within the top 5000 in at least one day in the week based on the traffic volume. Recall from Section 2.3.1 that about 1% of subscribers send about 60% of traffic. The number 5000 forms roughly 1% of the number of subscribers that generate traffic in a typical day.

Figure 8 (a) shows the number of days these heavy users generate traffic. Interestingly, we see that almost 50-60% of the heavy users generate traffic

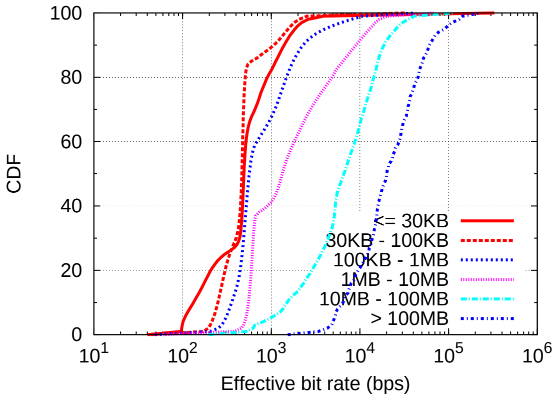


Figure 9: CDF of effective bit rate for subscribers categorized by traffic generated per day.

only on one day in the entire week. This result is different from the percentage of subscribers (about 28%) generating traffic only on one day considering all subscribers as shown in 2.3.4. *This shows that most heavy users are not habitual, but actually quite sporadic.* In Figure 8 (b), we show the distribution of hourly activity of heavy user during peak hours (8 AM to 8 PM). This plot shows that a typical heavy user appear in 4 to 6 different hours during the peak hours in the days they generate traffic. This distribution is not significantly different from the distribution of the entire set of subscribers.

It is also interesting to look at how efficiently subscribers use radio resources, and whether there is any difference between the low and high volume users. To do this, we define a metric called 'effective bit rate'. This is the ratio between the amount of traffic generated by subscribers to the airtime (time actually occupying the radio channel irrespective of traffic generated) used by them. This metric tells us how efficiently the allocated radio channel is used for sending traffic. Figure 9 shows the CDF of effective bit rate with subscribers categorized based on the amount of daily traffic they generate. We can clearly see that *subscribers generating less traffic have progressively poorer effective bit rate.* This may be due to the applications used by subscribers not fully utilizing the allocated channel bandwidth. Even the effective bit rate of a typical high volume subscriber ($\geq$ 100 MB) is approximately 20 Kbps which is much less compared to the maximum nominal bit rate that could be

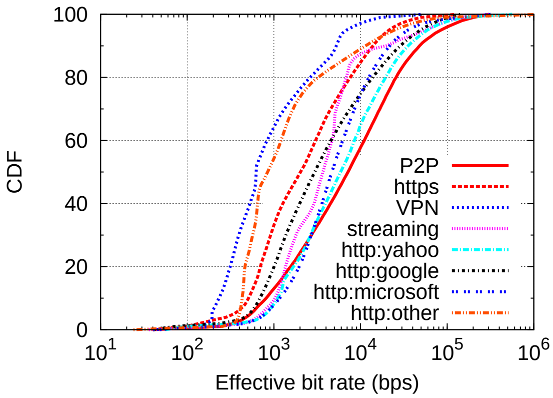
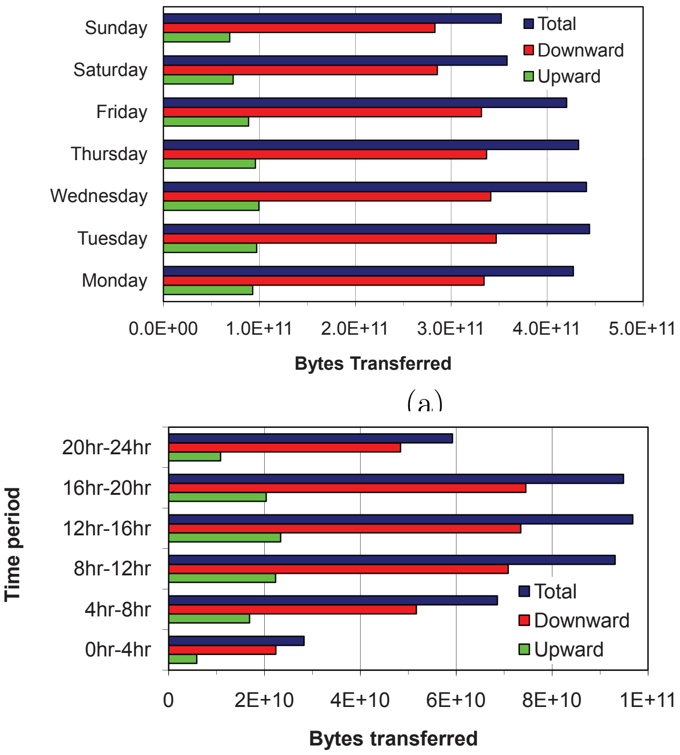


Figure 10: Effective bit rate for popular applications.

supported. For a low volume subscriber, it is roughly 0.5 Kbps.

To investigate the reason for poor efficiency, we identify the most popular applications (that account for 75% of total daily traffic among all subscribers) and study their channel efficiencies. See Figure 10. Only TCP based applications are considered so that the flow start and stop instants can be clearly identified. During the lifetime of each flow the number of bits transmitted and the total airtime consumed are used to compute the effective bit rate. Port numbers in the packet headers are used to identify the application type. For http we also track the server IP addresses to identify the sites visited. Statistics for a few popular sites (google, microsoft and yahoo) are shown separately.

Note that applications like VPN, https (used for secure connection) and http (for sites other than the popular ones such as google, microsoft and yahoo) have the poorest efficiency, while P2P and http for certain popular sites (yahoo) have the best efficiency. The median difference between VPN and P2P is over an order of magnitude (note the log nature of the horizontal axis). Broadly, it appears that *the enterprise applications generate much less traffic compared to other applications for the same airtime consumed.* The likely reason for this is that such applications tend to use the network sporadically (e.g., frequent use of keep-alive messages in VPN) and/or typically are not high throughput applications. Considering the nature of the dormancy period



(a)

(b)

Figure 11: (a) Aggregate load on the network on each day of the week. (b) Breakdown of total load in a single day in 4 hour periods.

in 3G networks it it easy to see that channel usage will be inefficient in such applications. On the other hand, high throughput applications like P2P downloads or http browsing on certain popular sites tend to send more data during their allocated airtime. Overall again, all applications have significantly poorer effective bit rate compared to nominal bit rates of the underlying physical layer technology, implying significant scope of protocol improvements across layers. More will be discussed on this in Section 2.5.

2.4 Base Station Traffic Dynamics

In this section we turn our attention to the network behavior as a whole or in terms of network components (base stations) instead of focusing on subscribers.

2.4.1 Aggregate Load

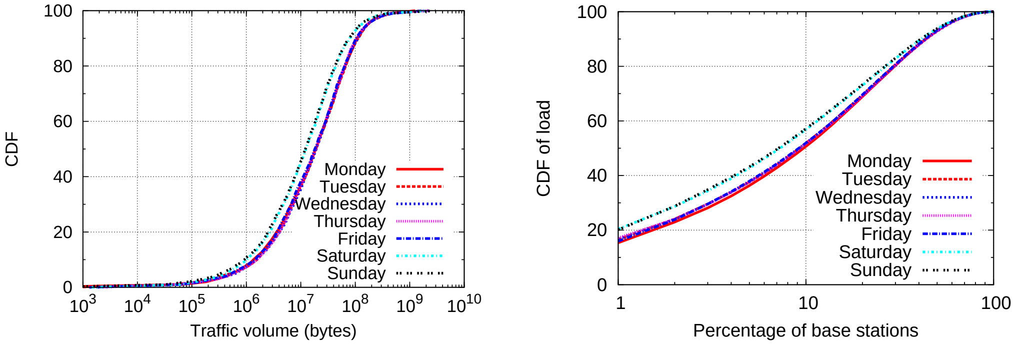
First, we characterize the aggregate load in the entire network considered. Figure 11(a) presents the total traffic split into upload and download for each day of the week. As expected, weekends see a lesser load. Also, downloads dominate relative to uploads with more than 75% of daily load coming from download traffic. We also break down the traffic load on the network in a single day into 4 hour periods, as shown in Figure 11(b). We can see that the load on the network is relatively low in the early morning hours, and roughly similar during the day and the evening.

2.4.2 Base Station Load Distribution

Next, we analyze the volume of daily traffic load for each base station. Figure 12(a) shows the CDF of the daily load (in bytes) of each base station for each day. It shows that roughly about 80% of the base stations are loaded in the range of 1-100MB per day and 10% of the base stations are highly loaded (more than 100MB per day). Figure 12(b) shows the CDF of daily base station loads normalized by the total network load. It shows that *10% of the base stations experience roughly about 50-60% of the aggregate traffic load.* In both cases, weekend behavior is slightly different than weekday behavior. The load imbalance seems more pronounced in weekends. This great imbalance of the base station loads indicates that a more careful cell planning is possibly needed. Network providers may use smaller cells or microcells at the hotspots to even out the imbalance.

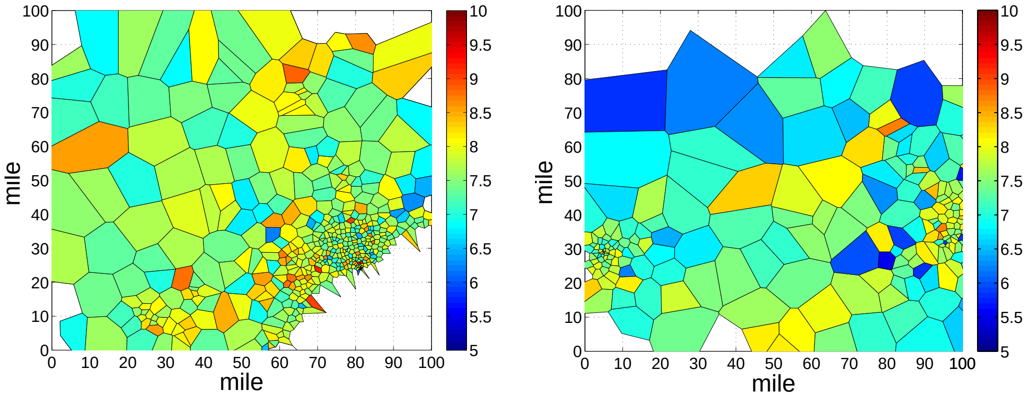
2.4.3 Spatial Characteristics

Our goal here is to identify whether the network load is spatially correlated. Such investigation can potentially help the provider to allocate resources appropriately. This can also be helpful in predicting the load of a spatially



(a) (b)

Figure 12: (a) CDF of actual daily traffic (in bytes) per base station. (b) CDF of normalized traffic per base station.



(a) Region 1 (b) Region 2

Figure 13: Total load of each cell in a typical day in two geographically separated regions. The partition in terms of Voronoi cells defines the coverage of each base station. The color bar on the right hand side of each figure indicate to the total load per cell in bytes in $\log_{10}$ scale.

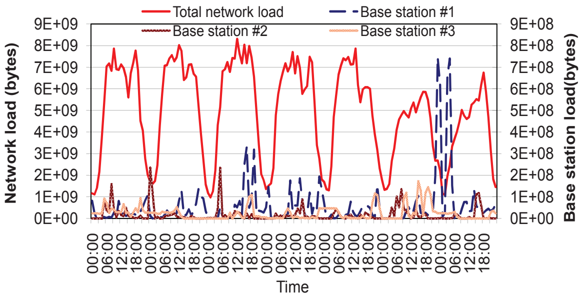
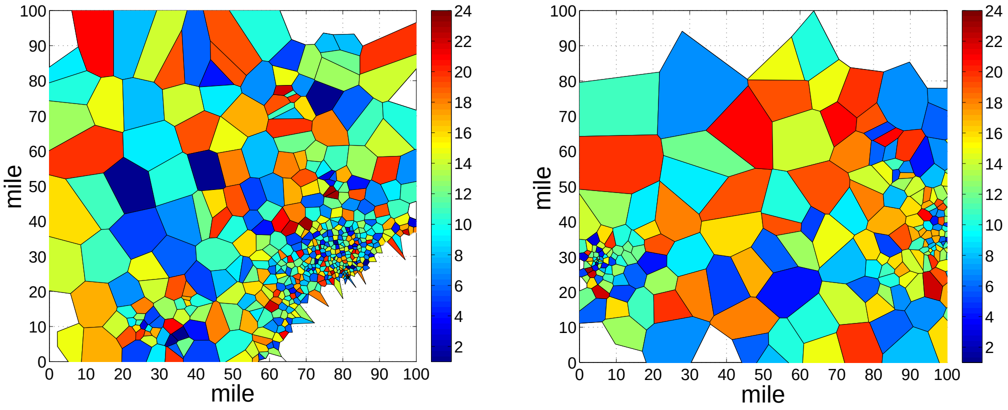


Figure 14: Aggregated network load in each hour and hourly load of three top loaded base stations. Note that they use two different scales.

separated region given the load of another region. We do preliminary tests and data exploration to investigate the spatial charateristics of network load. A deeper analysis for understanding the spatial relationship is presented in Chapter 3.

We investigate the spatial charateristics of network load using Voronoi cells. Each Voronoi cell corresponds to the geographic region of each base station's coverage. Figure 13 shows the aggregate load in bytes for each cell in a typical day in $\log_{10}$ scale for two geographically separated regions in the studied network. Each region is 100 mile × 100 mile and includes major city centers as well as suburban areas. Note higher density of Voronoi cells in certain areas (city centers) signifying some degree of cell planning. We can readily see again that *the cells are not uniformly loaded in space. The load differentials can extend several orders of magnitude.* There does appear to be some degree of negative correlation between the Voronoi cell size and load. This is expected as large Voronoi cells mean sparsely located base stations, implying sparser population density. Though no significant spatial correlation between adjacent cells is observed via visual inspection of similar plots for all days, different result comes up when spatial relation is invesigated in Chapter 3 using strong statistical techniques.



(a) Region 1

(b) Region 2

Figure 15: Peak hour of each cell in a single day in two geographically separated regions.

2.4.4 Temporal Characteristics

2.4.4.1 Load

We summarize the hourly load of each base station for the whole 7 day period. Figure 14 shows the hourly aggregate load of the entire network and then top three highly loaded base stations. The aggregate network load exhibits a nice periodic behavior with relatively high loads during the day and the lowest load during midnight. On the contrary, individual base station loads do not show that much periodicity. Also, the load curve varies significantly among individual base stations with their peaks occurring at different times of the day.

We then investigate how the network load varies temporally as well as spatially. To do this, we determine the peak hour of each base station in the day. The peak hour of a base station is the hour in which the given base station has the highest load among its own hourly loads of the day. Figure 15 shows the peak hour of each cell for the same two geographic regions described in Section 3.2 for a typical day. It shows that base stations have widely different peak hours that without further analysis appear somewhat random. Once again note the lack of spatial correlation.

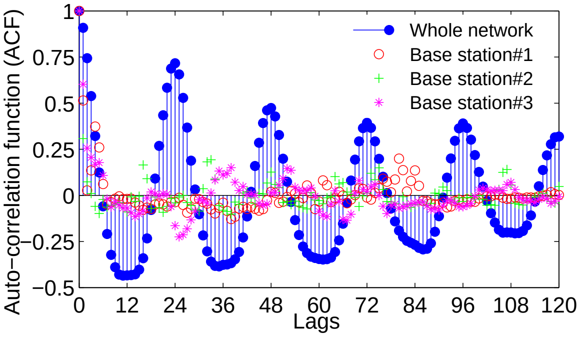


Figure 16: Auto-correlation Function on the network load in time series.

2.4.4.2 Auto-correlation

For rigorous analysis of the periodic behavior describing the network load we evaluate temporal correlation for a load metric. This will enable us understand the underlying trends and seasonal variations better. We represent the hourly aggregate load as time series for the whole network and also for top three base stations used in Section 2.4.4.1. Each time series thus has 168 data points for the 7 day period. Figure 16 shows the auto-correlation function (e.g., cross-correlation of the time series with itself) of these time series at different lags. Note that the *plot shows a high degree of temporal correlation.* Again the high peaks occur at 24 hour intervals and low peaks at 12 hour intervals. This is consistent with diurnal human activity patterns. Note that the positive peaks are very pronounced relative to the negative peaks and also the slow decreasing trend of the peaks with increasing lag. The high degree of correlation of network load at the same time of day can have tremendous implication in network resource management techniques. On the other hand, the individual base station loads do not show good temporal correlation (neither positive not negative) and the periodicity is also missing.

2.5 Summary of Observations and Practical Implications

We now summarize our key observations and identify important practical implications on network planning and protocol design in cellular data networks.

2.5.1 Key Observations

1) Traffic Load: There is a significant traffic imbalance both from individual subscriber's and base station's view point. Few subscribers and also few base stations carry a significant fraction of the total load. Less than 10% of subscribers generate 90% of the load, while 10% of base stations carry 50-60% of the load. The subscribers appear to be sporadic users of the network, the heavy users being more so. A typical heavy user only appears occasionally, but generates a large amount of traffic.
2) Mobility: A large fraction of subscribers have limited mobility (roughly half of them being practically static moving within just one mile). The mobility also exhibits periodic behavior with high probability of returning to the same location at the same time of the day. Overall, the mobility is highly predictable. Interestingly, the more mobile subscribers tend to generate more traffic.
3) Efficiency: Effective bit rate is poor due to the intermittency of data transfers and channel dormancy effects. Efficiency is poorer for low volume users relative to high volume users. This could be tied to the types of applications they use. For example, enterprise applications appear to have much poorer effective bit rate relative to P2P.
4) Correlations: Aggregate network load exhibits excellent periodic behavior and temporal correlation, but individual base stations do not exhibit such properties in any significant extent.

2.5.2 Implications

1) Subscriber Pricing and Usage Pattern: An unlimited data plan with flat rate pricing is not efficient both from the carrier's perspective as well as the majority of subscribers' perspective. The CDF shown in Figure 2 can be used as a guidance to create 'tiered' rate plans. The idea of tiered rate plans are becoming popular [3] to provide different pricing options based on data usage. One of our future works is to devise optimal pricing schemes based on subscriber usage and available network capacity. Also, sporadic network use by high volume subscribers can create poor experience for other subscribers if such usage occurs during peak periods. This can be alleviated by providing high volume subscribers some incentives (e.g., lower pricing during off-peak hours) to move their load to other times.

2)'Wireless-Friendly' Protocol Design: The highly predictable nature of the mobility pattern can be exploited by innovative cloud-based content delivery applications. The idea is to cache the content of particular interest to a subscriber close to the edge of the network where the subscriber can be found with high probability [47]. This reduces the latency in accessing the content to a large extent. Location based services and targeted ad-services can exploit such highly predictable mobility pattern to optimize their performance. Further, it is clear that the network protocols and applications designed for general wired Internet usage are not very 'wireless-friendly,' using valuable channel air time very poorly. This inefficiency is much higher in enterprise applications. Innovative protocols that make use of the wireless channel more efficiently need to designed. We note that some recent research targeting energy usage addresses a similar issue (see, e.g., [89]).

3) Spectrum Allocation and Energy Savings: The high degree of variability in base station loads has important implication on spectrum allocation and energy saving schemes in the network. New energy saving schemes such as adaptively turning on/off certain carriers or radios in base stations based on the load experienced need to be developed. In Section 2.4.4, we noted that the peak hours of different cells vary a lot, which advocates dynamic allocation of spectrum resources to highly loaded cells during their peak hours. One of our future work will be to model the demand characteristics on different cells in cellular data networks based on measurements for a long period of time and feed the model as inputs to dynamic spectrum allocation algorithms such as [131].

2.6 Related Work

There have been field measurement studies on 3G networks mainly focusing on the performance of data traffic, but only from the point of view of individual client devices. Representative works in this space are measurement studies on commercial WCDMA 3G/UMTS networks in [110], performance evaluation of GPRS and UMTS networks in [119], various forms of TCP performance evaluation in [88], [95] and [83], and cross layer studies in [139]. In addition, Joyce *et al.* [75] have presented single cell and network capacity measurements using a commercial network in UK. Yao *et al.* [141] have evaluated bandwidth predictability for HSDPA networks. Tan *et al.* [132] have studied the capacity of 3G networks in terms of throughput, latency, video and voice call handling ability. The authors in [36] have evaluated multimedia streaming through measurements taken from real networks (GSM, GPRS and UMTS). Performance of push-to-talk applications have been evaluated in [34] on 3G networks. The above studies do not use the global view of the network as a whole and a broader analysis of the subscriber behaviors are missing.

Such global views have been pursued only in a limited number of papers. The authors in [71] have carried out spectrum measurements in 2G and 3G bands during the 2006 Soccer World Cup in two German cities. They have shown that the change of spectrum usage is related to specific events. In [121], a measurement-based spectrum modeling approach has been developed using spatial statistics and random fields. The authors in [63] have shown the distribution of voice call duration analyzing the call logs from a cellular GSM provider. The authors in [138] have presented a large scale measurement analysis to characterize the primary usage in cellular voice network. In [81] the browsing behavior of mobile users in a large scale 3G data network has been analyzed. In contrast to these papers, our focus in this chapter is purely on data traffic behavior in the context of resource usage.

Finally, studying human mobility from cellular network data is an important component of this chapter, as mobility directly impacts resource usage. Much of our analysis has been motivated by Barabasi and co-authors' influential work on this topic [61, 30]. They have studied human mobility patterns based on the voice call records over a six-month period of 100,000 anonymized mobile phone users. They have concluded that human trajectories show a high degree of temporal and spatial regularity. In [65] an analysis of user mobility patterns is presented based on data traffic traces from a major regional CDMA2000 cellular network. The overall mobility was found to be limited. Pathirana *et al.* [102] have presented a technique to predict the trajectory of a user in a variant of GSM network. Authors in [30] have investigated the human dynamics and social interactions, and focused on the occurrence of anomalous events. None of these works, however, directly relate the users' mobility to network access behavior and network usage patterns.

2.7 Conclusions

In our knowledge, our work presented in this chapter is the first major study in measurement analysis of subscriber and network behavior in a large scale 3G data network. We have made several important observations related to traffic load, mobility and resource efficiency. We have indicated the implications of these observations in pricing, protocol design and resource management. Our future work will target (i) analysis for longer periods of time as well as in finer grain, (ii) addressing the topics highlighted in the discussions about implications in Section 2.5.

Chapter 3

Spatial Characteristics of Traffic in Cellular Data Networks

3.1 Introduction

In the previous chapter, we have studied the traffic dynamics from the subscriber and network perspectives as well as its impact on spectrum allocation using a large scale data traffic collected *at the core* of a nation-wide 3G network [105]. Our focus in this chapter is to continue analysis on the same data set, but focus on the spatial properties and causal relationships in the network. We specifically focus on 'spatially significant behavior' in terms of the resource usage on the network infrastructure (i.e., cells or base stations (BS)). Our goal is to provide answers to important questions regarding i) how, or if at all, radio resource usage at base stations are spatially correlated, ii) how base stations can be clustered based on the similarity of their resource usage patterns, and iii) whether causal influence exists in the network in that a base station influences the load on other neighboring base stations. These questions are important from the network providers' perspectives specifically in the context of resource (including spectrum) management and planning. Our hope is that our analysis will lead to a better understanding of such behaviors prompting new resource planning, spectrum allocation and network design techniques.

The reminder of the chapter is organized as follows. Section 3.2 focuses on the spatial correlations among base stations. We present an optimal clustering of the base stations based on their resource usage in Section 3.3. In Section 3.4, we investigate the underlying causal structure of the network. Section 3.5 describes the related work, and Section 3.6 concludes the chapter.

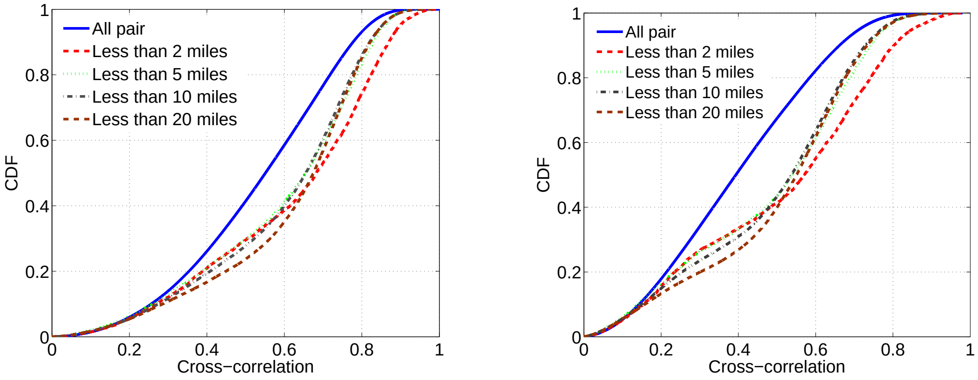
3.2 Spatial Correlation

We study different techniques to understand the spatial characteristic in cellular network. The results indicate that there is significant spatial correlation in the network. We also study two metrics representing network resource usage to use in our analysis.

Traffic load in terms of bytes is the most used metric to describe resource usage in a network. In 3G cellular networks, another metric, 'airtime', can provide a more realistic indication of spectrum usage. In the commonly used 3G standards (3GPP) [19] or 3GPP2) [18]), a subscriber requests and is in turn allocated a radio channel whenever it has data to send. The allocated radio channel is revoked by the network when the subscriber is dormant for a certain period, modeled using the so-called 'inactivity timer' (typically about 10 seconds) [39]. The inactivity timer is configurable for different networks [19, 18]. A subscriber can go between active and dormant state multiple times within a single mobile IP session. We refer the amount of time a subscriber holds onto a radio channel (regardless of whether it actually communicates) as the 'airtime.' Effectively, the airtime gives us the actual amount of time a subscriber uses radio and spectrum resources.

From this simple understanding, traffic load and airtime should be highly correlated. However, our analysis (not reported here) has shown that this is not the case always. A recent study analyzed the relationship between load and airtime at least indirectly [112]. It shows that a significant portion (up to 45.3%) of the channel time is wasted idling. The fundamental reason for this inefficiency is the use of a common inactivity timer value regardless of the nature of traffic [112].

Given this background, we use airtime for our analysis as it is more directly related to the radio resource usage for the current generation networks.



(a) One hour interval

(b) 10 minute interval

Figure 17: CDF of cross-correlation between pairs of base stations (all pairs as well as pairs within different ranges) based on airtime for different summarizatio granularity.

Further exposition of traffic load vs. airtime question is left for a different work. An interested reader can also refer to [112] for existing studies in this regard. We will use the time series of airtime (sec) for two summarization granularities: 1 hour and 10 mins, for each base station for the entire 7 days period. The goal of using two different granularity is to check whether the spatial characteristics depend on the summarization interval. Because of the similar nature of the resulting plots, we will sometimes present the 1 hour data only in the plots.

We will now use the concept of cross-correlation to investigate the nature of spatial correlation of resource usage in the network. Cross-correlation is a standard statistical method of estimating the degree to which two time series are correlated [8, 28]. We compute the cross-correlation of zero lag between various pairs of base stations using the time series of airtime. Figure 17 shows the CDF of cross-correlation for all pairs of base stations as well as pairs of base stations within different ranges for both 1 hour and 10 mins granularities. Note that cross-correlation, in general, between pairs of base stations is relatively high with the 1 hour interval showing a somewhat higher cross-correlation (median around 0.55 for 1 hour interval and 0.4 for 10 min interval). Also, when categorized into groups of base stations that are within different distances from each other, *closer base stations show significantly higher cross-correlation.* For example, for base stations that are within 2 miles from each other the median cross-correlation for 1 hour interval is around 0.7.

Given that there is a noticeable degree of spatial significance in the cross-correlation study, we study the spatial correlation property using another angle with the help of the 'Moran's I' statistic [98]. Moran's I is a popularly used measure of spatial autocorrelation. It measures how correlated a spatial phenomenon is along space similarly as temporal autocorrelation measures correlation along time. Several earlier works (e.g., [121]) use Moran's I to investigate spatial behavior. A concept of distance is used to indicate proximity and is used as 'weights' in the formula. Moran's I is defined as:

$$I = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(x_i - \overline{x})(x_j - \overline{x})}{\sum_i (x_i - \overline{x})^2}, \tag{3}$$

where x is the random variable studied, $\overline{x}$ being the sample mean, x_i's are the observations. w_{ij} is the weight associated with each pair (x_i, x_j) and N is the number of observations. Here, the random variable x is the hourly load on a base station. As common with Moran's I studies, we use binary weights: $w_{ij} = 1$, when the base stations are in close proximity (a threshold of 2 miles is used), else $w_{ij} = 0$. We then plot the Moran's I metric for hourly loads of all base stations in the network on a temporal scale. Figure 18 shows the Moran's I indices for each hour using our hourly data. The plot has been smoothened to remove noise by using a sliding window averaging with a window size of 4 hours. The plot shows high values (varying from 0.2 to 0.6) indicating significant spatial correlation. However, the periodic behavior with a diurnal cycle is interesting. It appears that while temporal usage patterns of base stations may be very different and might even miss periodicity (Section 2.4.4.2), *there is a general tendency for proximate base station loads to be more correlated when the loads are high.*

Next, to investigate the temproral aspect, we want to check whether the correlation varies on the time of the day. To do that we split the day in 6 time periods. Figure 19 shows the CDFs of cross-correlation for pair of base stations within 2 miles based on hourly airtime for each time period. It also shows the CDF considering the whole time series. Note that the correlation is

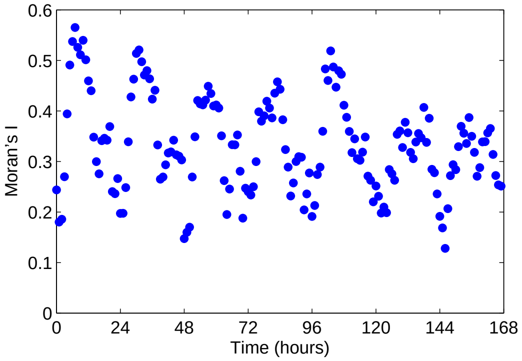


Figure 18: Moran's I values plotted on a temporal scale based on airtime on all base stations.

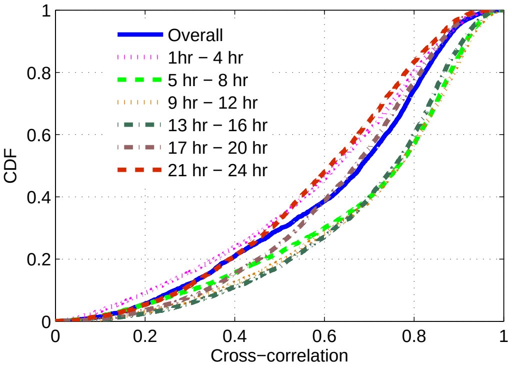


Figure 19: CDF of cross-correlation of pairs of base stations based on airtime, categorized into different time periods in the day. Only base station pairs within 2 miles are considered.

higher than the overall correlation during the day time and lower during the night time. This is consistent to the Moras's I values.

3.3 Clustering Base Stations

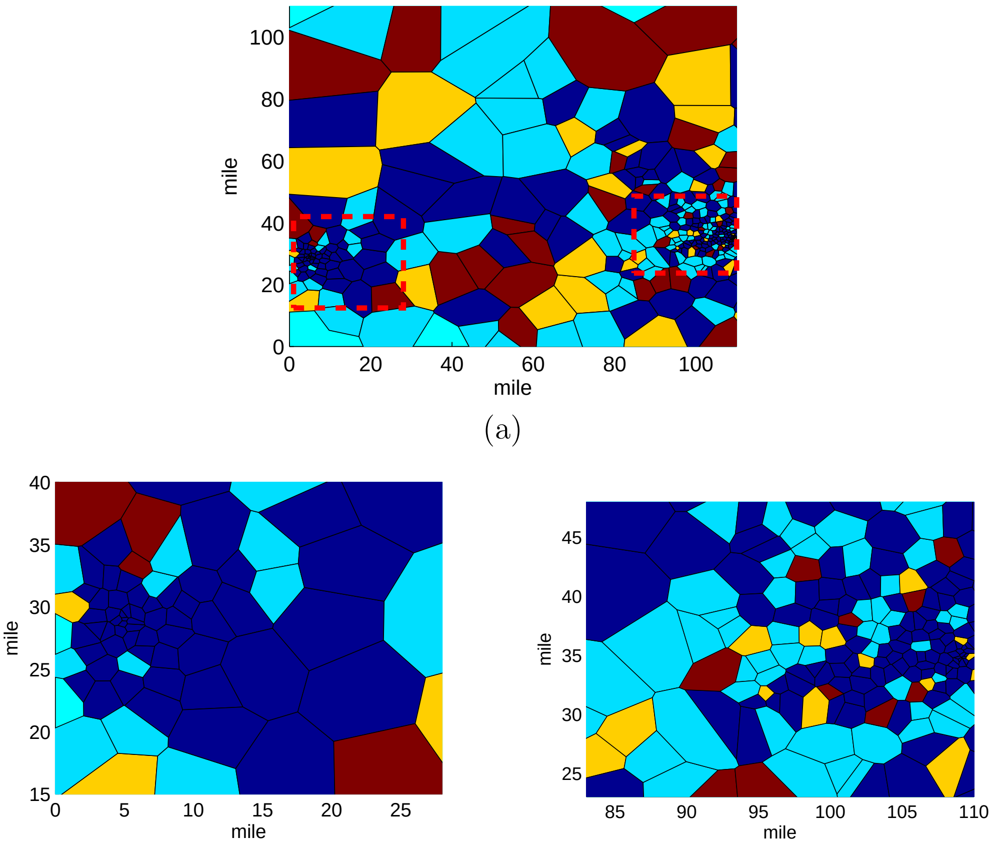
One can think of the pairwise cross-correlation as a similarity measure between base stations. It will be interesting to find out whether we can cluster base stations based on this measure and how such clusters look. The base stations within each cluster then exhibit similar behavior in terms of resource usage. Understanding the nature of such clusters can help the network provider in resource planning, as the provider now can think in terms of clusters or groups instead of individual base stations. The spatial nature of such clusters would be also interesting. For example, if the clusters form large connected components, then it demonstrates spatial significance. This has significant implication in terms of spectrum allocation. This, for example, shows that one can find large geographic regions, as opposed to base stations, that have correlated resource usage behavior.

We use 'spectral clustering' for clustering base stations. Spectral clustering is a powerful technique to partition points (in our case, base stations) into disjoint clusters such that points in the same cluster having a high degree of similarity (i.e., correlation) and points in different clusters having low degree of similarity [99, 143, 21]. The clustering algorithm works on the basis of a 'similarity matrix.' In our case, the similarity matrix is constructed using the pair-wise cross-correlation values for all pairs of base stations, thus forming a matrix.

We use the spectral clustering technique as presented in [99, 143]. The limitation of this technique is that the number of desired clusters needs to be specified. To compute the 'optimum' number of clusters, we follow the technique proposed in [143], where the algorithm self-tunes itself.

We get 4 clusters as the optimum for hourly data and 5 clusters for 10 min interval data. As the results for both the intervals are somewhat similar, we only show the results for 1 hour interval. Figure 20 shows the clustering output for a sample geographic region of size 110 mile $\times$ 110 mile. The map is partitioned into Voronoi cells. Each Voronoi cell approximates the geographic region of one base station's coverage. The color of each Voronoi cell indicates its cluster. The Voronoi cells widely vary in size – denser in downtowns/city centers and sparser in suburbs. We have also provided zoomed-in versions of two of the denser areas in the map.

Visual inspection of these colored maps reveals some degree of spatial connectedness of the clusters. Particularly, the 'blue' and 'cyan' clusters exhibit significant connectedness. Also, note that the 'blue' cluster is quite prevalent



(a)

(b)

(c)

Figure 20: (a) Spectral clustering of base stations in a 110 mile× 110 mile region shown by coloring of the corresponding Voronoi cells. Four different colors represent four clusters. (b) and (c) are the zoomed-in versions of the densely deployed base stations in the two areas indicated by red dotted rectangles in (a) on the left and right side, respectively.

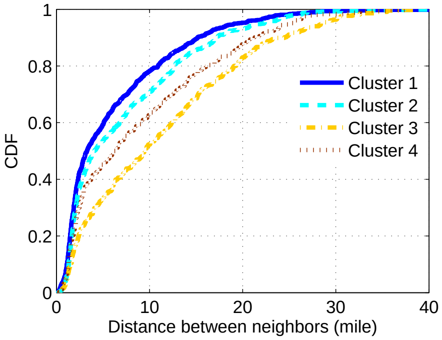


Figure 21: CDF of the distance between two neighboring base stations in each cluster.

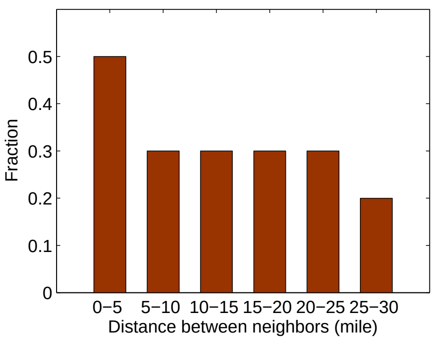


Figure 22: Fraction of neighbor pairs that are in the same cluster, categorized on distance.

in the dense (downtown) areas, followed by the 'cyan' cluster. The sparser areas, on the other hand, have contributions from all clusters, though 'cyan' appears somewhat more prevalent. Figure 21 shows the CDF of the distances between the neighboring base stations in each cluster. The same cluster color is used as before for easy readability. Note that the 'blue' cluster is the most densely packed (small cells). The 'brown' and 'yellow' clusters are mostly concentrated in the suburbs and thus less densely packed (larger cells). This indicates that there is some relationship between cell size and the cluster it belongs to. There is a tendency of cells of similar sizes to cluster together.

Finally, to further analyze the spatial connectedness of the clusters, we investigate the probability of a neighboring pair being in the same cluster. Overall, about 40% of the neighbor pairs are grouped in the same cluster. We also calculate this probability for neighbor pairs at different distances. Figure 22 shows that the neighbors that are geographically closer have a higher tendency to be in the same cluster (upto almost 50%).

3.4 Causality

From correlations, we turn to functional influences in this section. An important metric to understand the underlying functional connectivity in the network is the ‘causal influence’ among the base stations. To keep the computational requirements reasonable, we do this investigation among the neighboring base stations only. The causality relationship among the neighboring base stations can be helpful in predicting the base stations’ loads, and thus allocate the spectrum accordingly in advance. While there are many avenues to pursue this, we use the notion of Granger causality [62], a statistical concept used to measure causality between a pair of time series.

3.4.1 Granger Causality

Granger causality (G-causality) determines whether one time series is useful in forecasting another [62]. According to G-causality, one stochastic variable X_2 ‘Granger-causes’ another stochastic variable X_1 if the information in the past of X_2 helps predict the future of X_1 with a better accuracy than is possible when considering only information in the past of X_1 alone [62, 128]. In other words, there is a Granger causality from X_2 to X_1, if X_2 provides statistically significant information about the future value of X_1. Such causality relation is not symmetric, meaning that ‘X_2 Granger-causes X_1’ does not necessarily imply ‘X_1 Granger-causes X_2’.

Granger causality was originally used in the field of economics to study the relationship between different economic variables such as GDP, oil price, stock market price, unemployment rate and so on [48, 69]. Recently there has been a growing interest in the field of neuroscience for using G-causality to identify causal interactions in neural data (see, e.g., [127, 129]). Use of Granger causality in communication network measurement and analysis is, however, rare. The lone example we have found is a recent study using Granger causality to understand the relationship between building occupants’ energy usage and their IP traffic [82].

We now present the formal definition of Granger causality. Suppose, we have two time series $X_1(t)$ and $X_2(t)$ both of length T. We can describe the two time series using a bivariate autoregressive model [128]:

$$X_1(t) = \sum_{i=1}^{p} A_{11,i} X_1(t-i) + \sum_{i=1}^{p} A_{12,i} X_2(t-i) + \varepsilon_1(t). \tag{4}$$

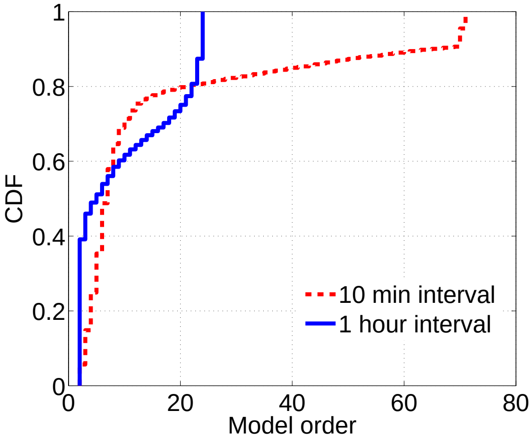


Figure 23: CDF of model order for each pair of base stations.

$$X_2(t) = \sum_{i=1}^{p} A_{21,i}X_1(t-i) + \sum_{i=1}^{p} A_{22,i}X_2(t-i) + \varepsilon_2(t). \tag{5}$$

Here, $p < T$ is the maximum number of lagged or past observations of X_2 (or X_1) used to predict the current value of X_1 (or X_2) at time t. It is called the model order and is provided as a parameter to the model. There are different criteria to determine the appropriate model order, p, so that the data can be represented correctly. Among them, Bayesian Information Criterion (BIC) [126] or the Akaike Information Criterion (AIC) [20] are mostly used. The matrix, $A = \{A_{mn,i}\}$ contains the model coefficients and ε_1 and ε_2 are the prediction errors or residuals. By definition, $X_2(X_1)$ Granger-causes $X_1(X_2)$, if all the coefficients $A_{12}(A_{21})$ are non-zero (in other words, if the variance of error term $\varepsilon_1(\varepsilon_2)$ is significantly reduced by the inclusion of $X_2(X_1)$ in the first (second) equation). It is important to check whether the causality is statistically significant or not. This can be done using the F-test [128]. To become statistically significant the F-statistic value should be greater than a critical value of the F-distribution for some desired significance threshold, between 0 and 1. A significance threshold closer to zero indicates a stricter test.

In our context, the time series for airtime consumed for a pair of neighboring (in the Voronoi sense) base stations describe the behavior of the two

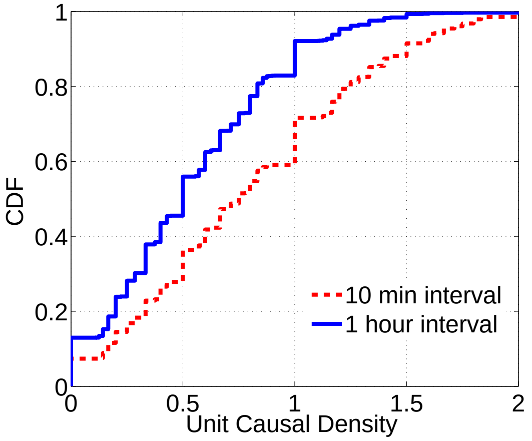


Figure 24: CDF of unit causal density of each base station.

variables X_1 and X_2. We have used the 'Granger Causal Connectivity Analysis' toolbox [128] for MATLAB for our analysis. Akaike Information Criterion (AIC) is used to find the model order p. For the statistical significance test we use 0.05 as the critical value. We test the causality for every neighboring pair of base stations in both directions. In our analysis, 32% of the neighbor pairs show significant causality at least in one direction for 1 hour interval data. This number increases to 40% when airtime is summarized to 10 min interval. *One can conclude that roughly for one third of the neighboring base station pairs there is casuality at least in one direction that is statistically significant.* Figure 23 shows the distribution of model orders for each pair of base stations. Note that model order is generally low (median is 5-7) for both the intervals.

To understand the causal properties of the network as a whole, we define a *causality graph* using the pair-wise causal relations [82]. The Granger causality graph is a directed graph $G = (V, E)$ where V is the set of vertices, E is the set of edges.Each base station in the network becomes a node in the graph. There is an edge from node a to b in the graph, that is $(a, b) \in E$, if the corresponding base stations are neighbors in the voronoi construction and there is significant Granger causality from a to b. The causal graph allows us to explore an interesting set of causal properties [128] that we will describe now.

3.4.2 Causal Density

The *causal density* of the dynamics of a system is a global measure of causal interactivity [128]. It is a single index showing the mean causality over the whole network. High values of causal density indicate that the network elements are globally coordinated in their activity. It is defined as the average G-causality over all the pairs of base stations considered. Causal density bounded in [0,1] gives the average count of significant Granger causality over the whole network. The causal density can be defined using the causality graph as follows:

$$\text{Causal density} = \frac{\sum_{a \in V} \sum_{b \in V-\{a\}} I[(b,a) \in E]}{\sum_{a \in V} |N_a|} \tag{6}$$

where N_a defines the set of neighbors of base station corresponding to node a in the voronoi sense and I is the indicator function. In our analysis we get causal density equal to 0.322 and 0.4 for 1 hour and 10 min interval data, respectively. This indicates existence of statistically significant G-causality on average.

A similar term, *unit causal density* can be defined to find the interaction locally for each base station [128]. This indicates how a base station is causally involved with its surrounding base stations. It is the sum of causal interactions around a base station normalized by its number of neighbors. A node with high unit causal density can be viewed as *causal hub*. Figure 24 shows the distribution of unit causal density. The median of causal density is 0.5 (0.7) and around 20% (40%) of the base stations have causal density greater than 1 for 1 hour (10 min) interval, which indicates *significant G-causality with at least 50% of neighbors in either direction.*

3.4.3 Causal Flow and Causal Path

It is interesting to explore which base stations produce more causal influence on its neighbors and which are the sink base stations, i.e., are mostly influenced by its neighbors. The concept of *causal flow* is used to discover this. Causal flow of a base station is the difference of causal interaction by it to its neighbors and causal interaction pointed to it by its neighbors [128]. In the context of

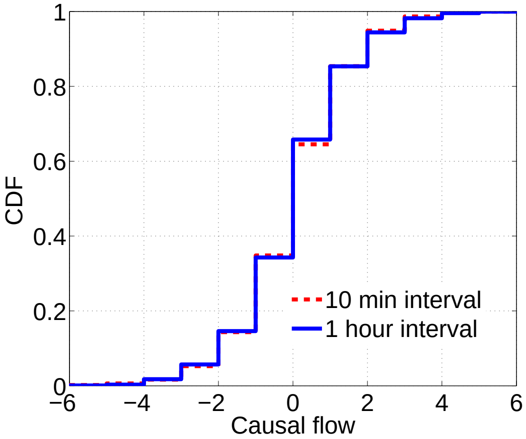


Figure 25: CDF of causal flow.

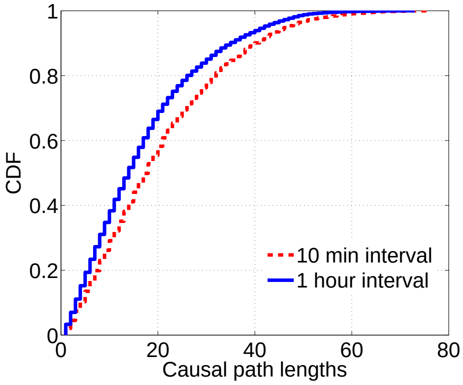
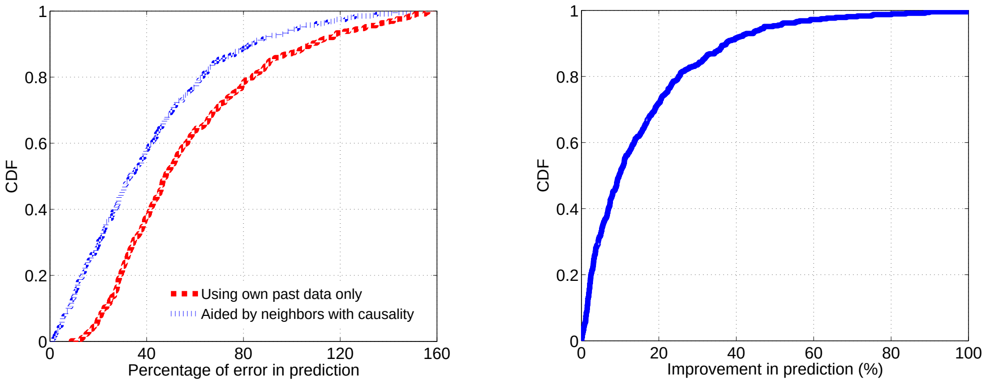


Figure 26: CDF of path lengths in the causal graph.

the causality graph, the causal flow of a node, a is the difference between its out-degree (number of edges from node a) and in-degree (number of edges into node a). A node with highly positive causal flow can be viewed as *causal source* and highly negative causal flow can be viewed as *causal sink*. Figure 25 shows the CDF of causal flows of the base stations in our network. It shows that about 30% of the base stations have large causal flow – either positive (≥ 2) or negative (≤ -2). This observation is similar for both the intervals.

Next we investigate how the influence from one base station propagates



(a)

(b)

Figure 27: (a) CDF of average absolute error percentage in prediction and (b) CDF of improvement in prediction.

or flows in the network. The paths in the causality graph indicate these flows. This somehow indicates the order in which the forecasting and resource allocation should be done for the base stations in the network. We define the *causal path* as a path in the causality graph. We enumerate all such causal paths in the causality graph. Figure 26 shows the CDF of causal path length in the causality graph for both intervals. It shows the presence of very long paths in the graph. The median is around 15 and the 90-percentile path length is 37.

In our future work, we plan to investigate whether the causal flow properties as well as the causal paths have any 'out-of-band' property. We are interested in identifying the causal sources and sinks on the map and correlating them with various forms of GIS data. We also like to investigate whether causal paths follow major roadways or neighborhoods with special properties.

3.4.4 Load Prediction using Causality Relations

Recall that a base station's load can be predicted better using the past information of other base stations that 'Granger-cause' the first one. Here, we show how knowledge of causality improves prediction. To do the prediction, we use the Auto-regressive Moving Average (ARMA) [4] model. We only use the one-hour interval summarization data. First, for each base station we create the model using its own load of the first 4 weekdays and predict the load of each hour of the fifth weekday of our data set. Next, we do the same using the past load of all the neighboring base stations with causal relationship along with its own history. Figure 27(a) shows the CDF of average absolute error percentage in prediction of each base station using both the techniques, while Figure 27(b) shows the CDF of improvement in prediction (the difference of average absolute error) of each base station. The plots show significant improvement in prediction when causal influences are taken into account.

3.5 Related Work

Relatively few papers have analyzed cellular network characteristics using a large scale measurement as this. The authors in [63] have shown the distribution of voice call durations analyzing the call logs from a cellular GSM provider. In a recent work [25], the authors have developed a tool to generate synthetic mobile network traffic using different data sets and models providing partial information about mobility and calling patterns. The authors in [30] have investigated some aspects of human dynamics and social interactions using large scale mobile phone records. The authors in [112] have characterized the settings of operational state machine that guides the radio resource allocation policy in a UMTS network. They have used actual cellular data traces for the investigation. The authors in [74] have analyzed customer tickets collected from a large cellular network to identify potential network problems. The authors in [81] have grouped users and browsing profile simultaneously using real mobile network data collected from a large 3G cellular service provider. The authors in [138] have presented a large scale measurement analysis to characterize the primary usage in cellular voice network. They have investigated the spatial correlation in the network but in a limited scope. In our earlier work [105] presented in Chapter 2 we have used the same data set as in this work and analyzed individual subscriber behaviors, subscriber mobility, and base station traffic dynamics at length.

3.6 Observations and Conclusion

In our knowledge, our work presented in this chapter is the first major study in measurement analysis of a large-scale 3G cellular data network with specific focus on spatial correlation and causality properties. We have made several important observations that have implications in network resource managament:

1. There is a significant amount of spatial correlation for base stations that are in close proximity. These correlations are time sensitive. They increase during high traffic times (midday) and fall during low traffic times (midnight).

2. Spectral clustering based on cross-correlation shows that cells of similar sizes and in the same neighborhoods have a tendency to be clustered together. Also, the number of clusters was found to be small.

3. There is a statistically significant causal structure in the network affecting roughly one-third of the base stations. Causal paths tend to be long, indicating long chains of influence in the network.

The above observations can help develop future analysis and forecasting tools to better provision the network, and for better spectrum and radio resource management and planning. For example, spatially significant correlation in the network load indicates that for many monitoring purposes it may be sufficient to sample loads sparsely across time and space. Since there are only few clusters, these techniques can be made quite attractive. This will in effect reduce the network monitoring burden on the part of the operator. Existence of causal influence between neighboring base stations indicates that load forecasting techniques need to use past loads of neighbors for better prediction. Such forecasting may be useful in various resource management decisions, including spectrum management and energy conservation. The model order is typically small, meaning that one does not need to go too much into the past and thus archiving burden is not significant. Further, the existence of long causal paths is interesting and needs to be examined carefully with respect to available out of band information, such as nature of neighborhoods.

Chapter 4

Prioritizing Traffic flows and Opportunistic Scheduling

4.1 Introduction

Considering the situation of exponential increase in traffic volume and the analysis presented in the previous chapters, it is a challenge for the service providers to accommodate the traffic as increasing network capacity involves major capital investment in terms of new spectrally efficient technology, additional spectrum and/or additional base stations. As the network resource is limited and can not be increased readily, it is important for the service providers to come up with a plan to tackle the traffic out-burst.

As reported in previous studies [25, 105], the load under a base station in a cellular data network fluctuates during the day, following a diurnal pattern, very high during the peak period (mid day) and typically low during the off-peak period (late night). The difference of traffic volume between peak and off-peak period is also very high. Due to this high variance in load, base station resources are under-utilized for a significant period during the day. Dynamic resource allocation can address this situation. But this does not quite solve the issue of capacity limitation, as during the peak period all base stations must allocate all available resources to accommodate the traffic. We take the approach of addressing this issue in the higher layer by shifting some traffic load from peak to off-peak periods in an opportunistic fashion. The goal is to reduce the peak-to-average ratio of traffic load in the network – ideally flattening the load curve as much as possible. This enables the provider to accommodate more users in the network without investing in capacity improvement. The basic idea is not unlike recent efforts in developing smart electric grids [125, 97] where there is an interest in reducing peak load by shifting load towards off-peak hours when electricity is cheaper. But in our case, the temporal shift of the load also prompts the opportunity of spatial movement considering the mobility of the subscriber. This should reduce the variance of load under a base station and also lower the peak of the load curve while serving the same total load. This evidently allows the service provider to accommodate more users in the network without any direct investment on capacity.

4.1.1 Approach

In this chapter, we consider a model where a fraction of network flows identified by the subscribers can be delayed. Flows of this category are treated with low priority. Interactive applications and most of the short-lived flows like mail reading or http browsing may not fall into this category. Possible examples of low priority flows are media download/upload, P2P flows etc., that can tolerate a reasonable amount of delay without hurting the user experience any significantly. The subscriber can specify a deadline by which he/she wants the service to be provided and network makes a best-effort to find appropriate spot in time and space to fulfill the request. The rest of the traffic in the network is treated as high priority and they are served immediately. This model allows the network to move around the low priority traffic both in space and time, and schedule them with the availability of spare resource of the currently associated base station of the corresponding subscriber.

The above approach provides two clear benefits:

i. Moving low priority traffic from a congested space-time point to another that has the capacity to carry the traffic; this allows the high priority traffic to be served with better performance.

ii. Reducing peak resource requirements by removing the load from the peak period to off-peak period; and thus indirectly allowing the network to make room for more users.

Our focus in this chapter is to evaluate the model described above and analyze the feasibility of two simple approaches to schedule the low priority traffic: (i) greedy scheduling approach and (ii) modeling-based approach. We evaluate the waiting time of the low priority traffic and also analyze the effect on high priority traffic. We also investigate how much the model can reduce the resource requirement in the network. The evaluation is done using a trace-driven simulation on a large-scale data traffic collected at the core of a nation-wide 3G network for our analysis. The data set spans one week in 2007 and consists of all data traffic associated with the entire subscriber base (in the order of hundreds of thousands) in a nation-wide network with thousands of base stations. All generated data packet headers (but not including user payloads) and various signaling and accounting packets are captured, archived and later post processed using a tool we have developed to gather all the flow level information. This is the same data set we used in our earlier works [105, 106] presented in Chapter 2 and 3.

The rest of the chapter is organized as follows. We describe our model and scheduling strategy in Section 4.2. We present our analysis using the greedy scheduling-based approach in Section 4.3 and describe the modeling-based approach in Section 4.4. We describe the implications both from the perspective of network provider and subscriber in Section 4.5. We discuss related work in Section 4.6 and conclude in Section 4.7.

4.2 Overall Approach

4.2.1 Model Description

We use an abstract model of the base station behavior to help us analyze opportunistic scheduling. A number of base stations covers a geographic region. A subscriber moves in the network and associates to a single base station at any time instant. When a subscriber creates a flow, the associated base station allocates radio resource (channel) for that flow. Here flow means TCP or UDP

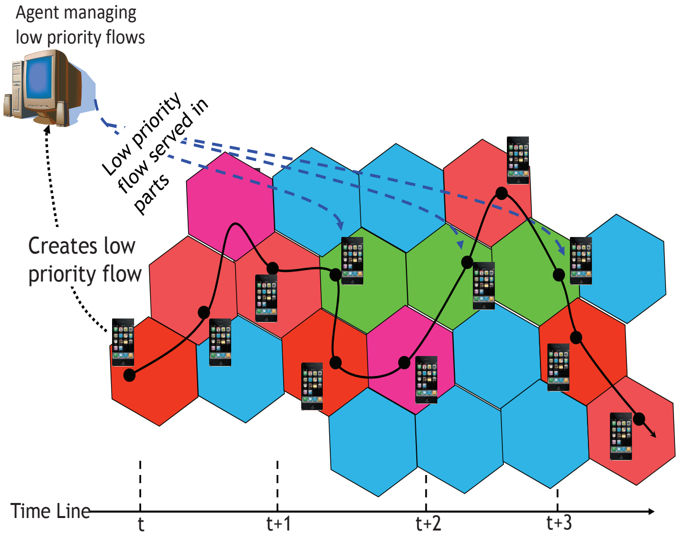


Figure 28: Overall idea of opportunistic scheduling of low priority flows. The color in each cell indicates the congestion level in that cell. Red means highly congested and green means no congestion. The trajectory of a subscriber is shown. The low priority flow is served when the congestion level at the cell is low.

flow (upload or download). If the subscriber moves from one cell to another, hand off takes place and all the ongoing flows are now served by the new base station. In the current cellular network design, all regular flows are treated equally (i.e., ‘high priority’ in our terminology), meaning that they need to be served immediately. Base stations need to be equipped with enough resource to accommodate all such flows, especially in the peak period. A flow arriving in a congested situation either gets dropped or served with a poor performance.

In our model, we introduce a second category of flow priority. These flows have ‘low priority’ and can be delayed and scheduled opportunistically based on the resource availability. Figure 28 shows the overall idea. A subscriber creating a flow tags it with low priority and informs an ‘agent’ running higher in the hierarchy inside the core network. This agent also tracks individual base station loads and is responsible for scheduling the low priority flows. The agent uses its knowledge of the base station loads and also the estimated mobility of corresponding subscribers for the actual scheduling decisions. Note that low priority flows may need to wait for being scheduled by the agent and even after the start, it may need to be defered/suspended and resumed later for multiple times until it is completely served. This means that low priority flows may be served in chunks, as demonstrated in Figure 28. The subscriber can also specify a time window for a low priority flow within which the flow needs to be completed. The agent considers this time window as a deadline for that low priority flow and tries to schedule the flow accordingly.

There are many ways to implement such an agent-based scheduling at a higher layer. For example, there could be a counterpart of this agent running on the mobile device as well. This counterpart coordinates scheduling with the agent in the core network. The control information can be exchanged as higher layer packets (with high priority). While at the high level, the core idea is similar to QoS scheduling and such scheduling could potentially be done at the radio layer – depending on the actual technology used, our unique approach solely performs the scheduling at the higher layer (at flow level) at a coarse granularity. This makes the approach completely agnostic to underlying radio technology. This also enables easy integration of application-specific information with the scheduling. For example, photo upload on a social network could be non-urgent and can be scheduled at a later time, while emails must be downloaded immediately.

4.2.2 Approach

We assume that a fraction of the flows created by the subscribers is marked as low priority flows. We consider two different approaches for scheduling the low priority flows.

Greedy Scheduling Approach is a simple approach where the agent monitors the load of the base station at a suitable granularity where the corresponding subscriber of the low priority flow is associated with and starts the flow whenever there is any spare capacity available. While the idea itself is straightforward keeping track of each of the base station's load for a large number of low priority flows may create an extra overhead of control message traffic in the network. We discuss about this approach in more detail along with the results of our simulation run in Section 4.3. This greedy approach demonstrates the maximum potential of improvement via opportunistic scheduling.

Modeling Based Scheduling Approach schedules the low priority flows using a predictive model of base station loads and subscriber mobility. This makes it more practical as load measurements can happen at a coarser granularity and continuous real time load information is not needed. It models the mobility pattern of each subscriber in the network to predict their locations. It also models the load on each base station in the network and determines the opportunistic time spots of each base station where low priority flows can be scheduled. Using these two models, low priority flows are scheduled using a network flow-based problem formulation. While practical, this approach can suffer from modeling error. We discuss about this approach in more detail in Section 5.2.

We investigate how these approaches can help us in reducing the resource requirements in the network. To evaluate this, we assign a lower capacity to each base station in our simulation run and estimate the effect on both high and low priority flows for the same traffic as in our trace. The goal here is to schedule low priority flows so that high priority flows remain relatively unaffected and low priority flows suffer only reasonable delays.

4.2.3 Data Set

Our data set provides a range of information for each flow created by subscribers including the start time, flow duration in seconds and number of bytes transferred. It also provides the information of the corresponding subscriber and the base station where the flow is initiated. It keeps track of the mobility information of each subscriber regardless of flow creation. Thus, we have knowledge of all handoffs.

To perform any scheduling, we need to have some idea of the capacity of the base stations. However, this information is not directly available from the data set. We model the capacity, C_j, of a base station j as the maximum aggregate throughput of the base station observed during the span of the data trace. While this may not be perfectly accurate, this indeed provides a lower bound on the capacity. Thus, we will at best underestimate performance.

We also model the flows and their resource requirement. Our data set keeps track of the total number of bytes transferred and duration for each flow. From this information, we calculate the *average throughput*, T_i^{avg} for each flow i. Along with that our data set also provides the *maximum throughput*, T_{ij}^{max} achieved for each flow, i during its life-time under a base station, j. We consider this as an indication of the channel quality for that flow under that particular base station. Our flow-model suggests that each flow, scheduled under a base station, is served with a fraction of its maximum throughput. That is, a flow i scheduled under base station j is served with a throughput of σT_{ij}^{max}, where $0 < \sigma \leq 1$. At any instance of time, all flows under a base station j are served with the same fraction that is determined using this formulation: If $\sum_i T_{ij}^{max} \leq C_j$, then $\sigma = 1$; else σ is such that $\sum_i \sigma T_{ij}^{max} = C_j$.

The value of σ depends on the availability of the resources, specifically on the number of active flows in the base station. Since our trace does not have any packet level information, it is assumed that each flow is served with a constant throughput (rate) that depends on σ. The rate (or, σ) is changed at discrete points of time – at flow arrival or departure. A flow departs when all the bits specified in the trace are served.

To demonstrate the accuracy of the above capacity and flow rate models, we do a round of validation test via a simulation run. We use the data set as is without any priority enforcement of the flows and with full capacity of the base stations as modeled above. The goal is to simulate the flows with the arrival time and transmitted number of bytes as specified in the trace and investigate the deviation of the flow duration in the simulation with respect

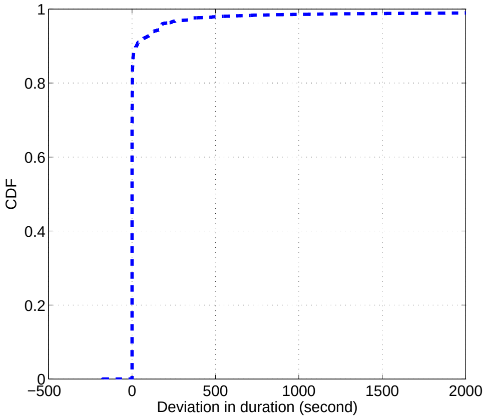


Figure 29: CDF of the deviation of flow duration in the simulation from that recorded the trace of each flow. Here, all the flows are treated with same priority.

to the recorded flow duration in the date set for each flow. Figure 29 plots the distribution of this deviation. More than 80% of the flows follow the same timeline as in the trace and among the rest of the flows, most of them deviate only marginally. This indicates the effectiveness of the model we will use in the simulation.

Low priority flows are introduced under a base station based only on either of the scheduling approaches discussed in next two sections. Only the spare capacity after serving all the high priority flows with $\sigma = 1$ (as discussed above) is distributed among all the currently assigned low priority flows. If there is no spare capacity available, the low priority flows must wait. More on this will be discussed for each approach in the respective sections.

Our data set keeps track of the total number of bytes transferred and duration for each flow. From this information, we calculate the *average throughput*, T_i^{avg} for each flow i. Along with that our data set also provides the *maximum throughput*, T_i^{max} achieved for each flow, i during its life-time under a base station, j. We consider this as an indication of channel quality for that flow under that particular base station. Our flow-model suggests that each flow, scheduled under a base station, is served with a fraction of its maximum throughput.

That is, a flow i scheduled under base station j is served with a throughput of σT_{ij}^{max}, where $0 < \sigma \leq 1$. At an instance of time, all the flows under a base station j are served with same fraction which is obtained using this formulation: $\sum_i \sigma T_{ij}^{max} \leq C_j$, $0 < \sigma \leq 1$. The value of σ needs to be changed based on the availability of the resource, specifically at the arrival and departture of any flow under the base station. We assume that when a flow is scheduled with some resource, it is served constantly with the specified throughput until that rate is changed or the size of the data transmission for that flow is over the total number of bytes as specified in the trace. This may not be realistic as in real protocol physical resource such as channels which are assigned to mobile devices can be time shared among multiple flows and acquisition of physical resources should depend on the packet generation behavior of the flow.

To demonstrate the effectiveness of our capacity and flow model, we do a sanity checking by a simulation run. We use the data set without any priority enforcement of the flows and with full capacity of the base stations observed from the trace. The goal is to simulate the flows with the arrival time and transmitted bytes as specified in the trace and investigate the variation of flow duration in the simulation with respect to the flow duration in the date set for each flow. Figure 29 plots the distribution of the variation of flow duration. More than 80% of the flows follow the same timeline as in the trace and among the rest of the flows, most of them are deviated marginally. This indicates the effectiveness of the model of our simulation.

Low priority flows are introduced under a base station based only on either of the scheduling approaches discussed in next two sections. Only the spare capacity after the high priority flows are distributed among all the currently assigned low priority flows. More on this will be discussed for each approach in the respective sections.

4.3 Greedy Scheduling Approach

In this section, we describe the greedy scheduling approach. We also develop a trace driven simulator to evaluate the approach using our data set described before. Our goal is to quantify the benefit that this model provides in terms

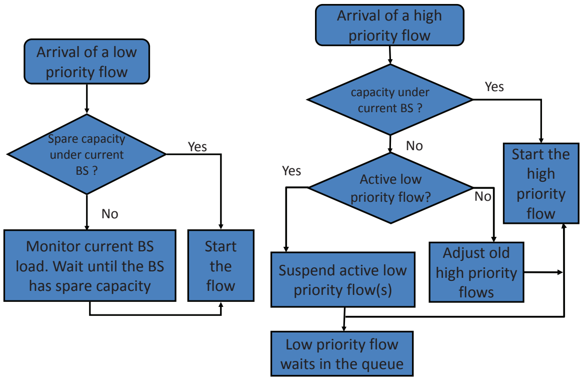

Arrival of a low priority flow
Spare capacity under current BS ?
Yes
No
Monitor current BS load. Wait until the BS has spare capacity
Start the flow
Arrival of a high priority flow
capacity under current BS ?
Yes
No
Start the high priority flow
Yes
Active low priority flow?
No
Suspend active low priority flow(s)
Adjust old high priority flows
Low priority flow waits in the queue

Figure 30: Flow chart describing the greedy scheduling approach.

of reducing resource requirement using the greedy approach.

4.3.1 Approach

The greedy scheduling approach is described below. Also, see the flow chart in Figure 44.

- High priority flows are treated in the regular fashion. They start immediately after arrival. All the active high priority flows under a base station share the total capacity so that each of these flows is served with its own required capacity based on the availability.

- Upon arrival of a low priority flow from a subscriber, the agent checks whether there is any spare capacity under the base station where the subscriber is currently associated. If there is no capacity available, the flow is stored in a queue where all such low priority flows wait to be served. Otherwise, the flow starts immediately with the capacity available.

- An active low priority flow is deferred in case of an arrival of an high priority flow under the same base station having no spare capacity. The deferred low priority flow is stored in the queue with its current status.

- For each of the low priority flows waiting in the queue, the agent always checks for any spare capacity under the base station where the corresponding subscriber is associated with. Whenever the agent finds an opportunity under a base station, it starts a low priority flow from the waiting list with the capacity available.

- When a subscriber having active flows hands off from one base station to another, all the active high priority flows (if any) of that subscriber are first accommodated under the new base station. It may require to defer a number of active low priority flows under the new base station to accommodate the migrating high priority flows. If the mobile subscriber has any low priority flow being served by the old base station, the agent makes a decision about that flow based on the available capacity of the new base station.

We have developed an event-driven queueing simulator to study the impact of opportunistic scheduling. To classify flows into high and low priority we take the following approach. We assume that short-lived flows are of immediate need and cannot be delayed (e.g., http browsing or email reading). On the other hand, the subscriber could be incentivized to delay long-lived flows (e.g., large download or P2P traffic). We consider flows longer than 1500 sec as long-lived flows, where the overall average flow duration among all the flows is 150 sec. In our data set, around 12% of flows are such long-lived flows. A random subset of such flows (about 8% of all flows in the network) is chosen as low priority for our simulation.

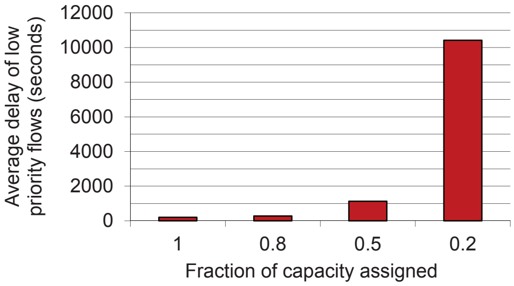
The throughput of the individual high priority flows varies depending on the load. As mentioned in the previous section, each high priority flow is assigned a fraction (σ) of its maximum throughput. In the simulation, a low priority flow is started under a base station only if all the current high priority flows are being served with their maximum throughput (i.e., $\sigma = 1$) and the base station still has spare capacity. The spare capacity of the base station is distributed among the active low priority flows under that base station. The number of low priority flows under a base station is incremented as long as each of the active low priority flows under that base station achieves at least its average throughput, T_i^{avg}. Arrival of a new high priority flow under a base station may need to suspend zero or more low priority flows depending on the capacity situation. In the case when there is no active low priority flow under the base station, the new flow is accommodated only by adjusting the throughput of other high priority flows, if required. On the other case, a number of active low priority flows under the base station is suspended to start the new high priority flow with its maximum throughput.

Our goal of this evaluation is to investigate how the model and the greedy approach can reduce the resource requirement in the network. To do this, we study what would happen to the incoming flows if the base station capacities were reduced. We do the simulation study for capacities such as 20%, 50%, 80% of the actual capacity of the base stations as determined from the trace and also provide the 100% results as the base case. The idea is to study the impact on the flows with reduced capacity base stations. If the impact is acceptable, e.g., low priority flows are not delayed substantially and only few high priority flows are impacted, this would indicate that more subscribers could be accommodated with the provisioned capacity.

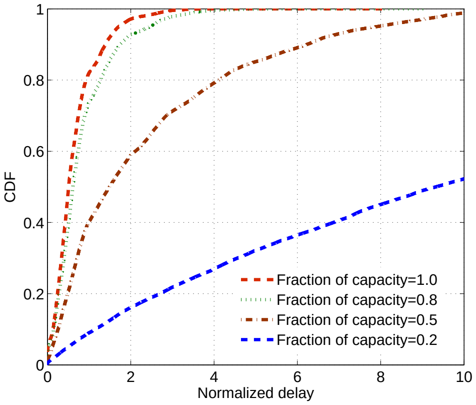
4.3.2 Simulation Results

With flow prioritization and opportunistic scheduling, it is possible that high priority flows end early relative to their actual end times in the trace. This is because they are expected to receive higher throughputs during service. Low priority flows on the other hand are likely to be deferred, possibly multiple times, and thus would end late relative to its actual end time in the trace. We use the term 'delay' for a low priority flow to indicate the difference between its end times in the simulation run and the actual trace (end time in simulation minus end time in trace). We use the term 'gain' for the high priority flows to indicate the same thing, but in the opposite direction (end time in trace minus end time in simulation).

Figure 31(a) shows the effect of greedy scheduling on the low priority flows for different (reduced) capacity assignments of the base stations. Note that the average delay of low priority flows is 1200 sec (20 min) when the capacity

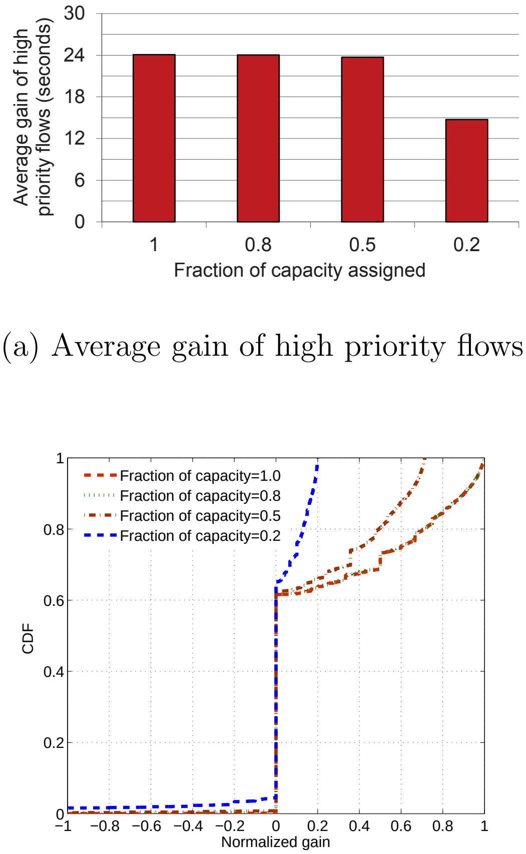


(a) Average delay of low priority flows.



(b) CDF of normalized delay of low priority flows.

Figure 31: Effect on the low priority flows in the greedy scheduling approach. 75% of the long-lived flows (8% of all flows) are assumed low-priority.



(a) Average gain of high priority flows.

(b) CDF of normalized gain of high priority flows.

Figure 32: Effect on the high priority flows in the greedy scheduling approach. 75% of the long-lived flows (8% of all flows) are assumed low-priority. The rest are high priority.

of base stations is made half of the actual. This is comparable to the original flow duration of long-lived flows in the data set as evident in Figure 31(b) that shows the delay of each flow normalized by its flow duration specified in the data set. On the other hand, Figure 32 shows the gain of the high priority flows in actual and normalized fashion. Note that more than half of the flows are unimpacted and and over one third of the flows show gain in varying degrees depending on the capacity of the base stations. A negligibly small fraction of high priority flows are negatively impacted for capacities 100%, 80% and 50%. This fraction is only noticeable (about 5%) for the 20% capacity case.

4.3.3 Critique of Greedy Scheduling

While the greedy scheduling approach is straightforward, the approach as described requires the agent to monitor the load on the base stations on a continuous basis, looking for scheduling opportunities. This naturally requires a significant amount of control information to be passed around among the base stations and the agent. This could be a significant overhead on the network, especially during the peak periods. Managing all the low priority flows in the network by a single agent may also be a scalability issue. Also, a low priority flow may suffer from a large number of suspend/resume operation incurring an extra processing overhead on the network. This can potentially introduce thrashing. Much of these issues can, however, be addressed via well-known techniques, such as choosing more granular measurement/scheduling intervals to reduce control overhead and choosing load thresholds to make scheduling decisions for low priority flows to reduce thrashing. But these can also negatively impact the performance advantage.

However, the greedy scheduling evaluation does indeed point out the significant capacity advantage that can be obtained just by lower the scheduling priority of a small number of flows.

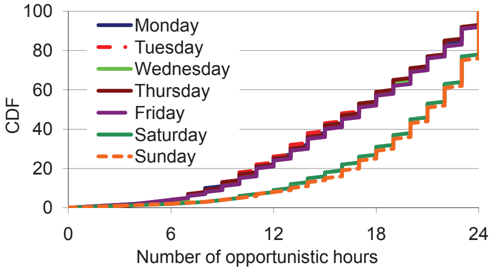


Figure 33: CDF of the number of opportunistic hours in a base station in a day.

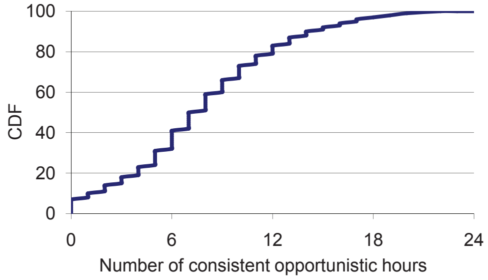


Figure 34: CDF of the number of 'consistent' opportunistic hours in a base station in the weekdays.

4.4 Modeling Based Scheduling Approach

We propose a modeling based approach to address the practical limitations of implementing the greedy approach that requires continuous load monitoring. The modeling based approach relies on the hypothesis that human mobility and network load are predictable and thus models for them can be created using historical trace data and off-line analysis. These models are useful in scheduling low-priority flows. This strategy completely eliminates the need for continuous real time monitoring. To establish the usefulness of this approach, we first evaluate how much predictability exists in the load and mobility that can be gainfully exploited.

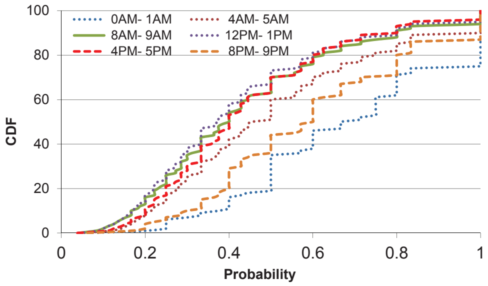


Figure 35: CDF of probability of a subscriber being in the most likely location.

4.4.1 Profiling Base Station Load

We first determine how frequently periods of opportunity arise when the base station load is 'low.' Low is defined as 25% of the capacity, where the capacity is defined as the maximum load the base station has seen in the entire period of the trace as before. We do this study in the granularity of an hour. The Figure 33 shows the CDF of the number of these opportunistic hours of a base station in each day. Note that a typical base station has at least 16 such 'opportunistic hours' in a day and the behaviour is similar among all the weekdays. Weekends, as can be expected, provide more number of opportunistic hours.

The next question is: Is the set of opportunistic hours of a base station 'consistent' (i.e., same hour of day across days)? Figure 34 shows the CDF of the number of consistent opportunistic hours of each base station among the 5 weekdays. We kept the weekends out of this as the nature of load in weekends is different from the weekdays. We see that a typical base station has 7 opportunistic hours that are consistent among days. This analysis of base station load indicates that a plenty of scheduling opportunities exists for low priority flows and much of it is predictable.

4.4.2 Profiling Subscriber Mobility

We now explore whether similar predictability exists in subscriber mobility. We model the subscriber trajectories to find out the probability of a subscriber being at a specific location (that is, in a specific cell) at a given time instance. To do this, we split each day into time periods (hour in our analysis). For each period, we measure the total time a subscriber spends in different cells. Specifically, if the length of the time period is l_t, then for each time period t_i of the day, the subscriber is observed for a total of $5 \cdot l_t$ time as profiling is done using the 5 weekdays in our data set. Suppose, the subscriber is seen at location j for duration δ_{ijk} during the the time period t_i on day k. We calculate the probability of the subscriber being in location j during time period t_i as the ratio of $\sum_{k=1}^{5} \delta_{ijk}$ and $5 \cdot l_t$. The distribution is created for each subscriber for each time period.

Figure 35 shows the CDF of the probability of a subscriber being in the most likely location at different time periods. Here we consider 1 hour time period and a few selected periods are shown. The location with the highest probability (see above) during a time period is chosen as the subscriber's most likely location at that time period. Note that a typical subscriber is found in his most likely location with probability 0.4-0.6 (median values). Note that this probability increases during the off-peak period. This analysis indicates that a subscriber's location can be predicted with a reasonable degree of accuracy.

4.4.3 Scheduling Low Priority Flows

We formulate the problem of scheduling the low priority flows as a *network flow* problem [41] using the profiles created for each base station and subscriber. Here, for scheduling convenience we also assume that each low priority flow also has a deadline by which it needs to be finished. We construct a network graph as shown in Figure 36 as follows:

- Each subscriber having at least an unserved low priority flow is represented as a node u_i. Each of these nodes is connected to nodes, marked as f_j, representing low priority flows created by the corresponding subscriber. The weight on this directed edge, denoted as d_{ij} is the estimated

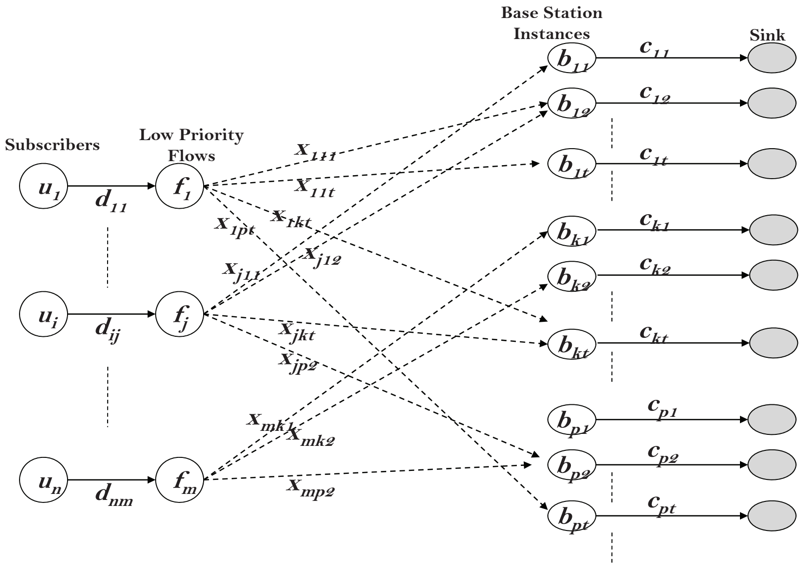


Figure 36: Network graph used to solve the scheduling problem using the modeling based approach.

number of bytes to serve the low priority flow.[1]

- We create an instance b_{kt} for each base station b_k at time period t. Each of these nodes is connected to a sink with a directed edge with weight c_{kt} denoting the spare capacity available under base station b_k at time period t. This is obtained from the profiles created for each base station based on historical data or via any other statistical estimation process.

- Node f_j representing a low priority flow created by subscriber u_i is connected to different base station instances based on the estimated mobility of the subscriber and the deadline of the flow. This means that f_j is connected to b_{kt} if the subscriber u_i is likely to be under base station b_k with reasonably high probability at time period t (t is within

[1] We assume that such estimation can be done using application layer information or prior profiling.

the specified deadline of the flow). The weight of this edge, denoted as x_{jkt} is a function of the throughput achieved and the duration of the subscriber's stay under base station b_k in period t. This is an estimation of the number of bytes the subscriber u_i can transmit during his/her stay under base station b_k in period t. This is also modeled from the historical data.

The scheduling agent constructs this graph periodically with all the low priority flows waiting with their current states and determine the schedule by solving this as a special case of the network flow problem where there are multiple sources and multiple sinks [15]. The subscriber nodes u_i act as source nodes. This formulation makes sure that base stations do not get overloaded and low priority flows are scheduled with in the specified deadline. As the scheduling is done by the agent periodically, it can cover up any modeling and estimation error in subscriber mobility and base station loads. The mobile device with the subscriber informs the agent about its current location (i.e., associated base station) whenever there is a handoff. Based on the location of the subscriber and the computed schedule, the agent starts any low priority flow that might be waiting.

4.4.4 Evaluation

We evaluate the above modeling based approach in a similar manner as we have done it for the greedy scheduling approach: assigning a lower capacity to the base stations and analyzing the effects on both high and low priority flows to demonstrate the reduction in resource requirement. Before going into the real evaluation, we model the load of each base station in the network to predict the spare capacity at each time period (one hour in our case). As our evaluation includes assigning lower capacities to the base stations, we also need to model the base station load for each of the lower capacity assignment. For each such capacity assignment we simulate the network with all flows in the data set and model the spare capacity in each time period for each base station. We also model the mobility of each subscriber by calculating the probability of the subscriber being under a base station at a specific hour. For the purpose of this modeling, we only use the data set of 5 weekdays from our week-long data. Week-end data is not deemed statistically meaningful as there are only two days and their nature substantially differs from the weekdays.

For a meaningful evaluation, we will need a long term trace. Since the trace is relatively short (only 5 weekdays), for evaluation purposes we synthetically augment the trace using established statistical techniques. The augmented data considers all the subscribers and base stations from the original trace data for the weekdays. The data generation is based on the probability of a subscriber creating a flow under a base station within a time period. Specifically, in the augmented data set – in a given time period within a day – a base station for a subscriber is chosen randomly based on the probability of the subscriber's association with this base station in the original data set at the same time period. It is assumed that the subscriber will be associated with this chosen base station for that entire time period. The subscriber creates flows during this time period according to the probability distribution of the number of flows the subscriber creates during this period. This determines the number of flows the subscriber creates. The actual flows are then selected randomly from the pool of flows for this subscriber for the specific time period regardless of location.

Just like in the earlier case, we identify a fraction of the long-lived flows (8% of all the flows in the synthetic data) as the low priority flows. For each of these flows, a deadline is picked randomly from an window of 1 to 4 hours beyond the arrival time of the flow (with random numbers redrawn if the chosen deadline is earlier than the original flow end time). Note that the additional time provided to the flow is comparable to the original flow duration as we are only considering long-lived flows as low priority. We apply our approach to schedule the low priority flows using the models created from our original data set. At the beginning of each hour, we determine a schedule of all low priority flows that are waiting.[2] Note that any low priority flow arriving in the middle of an hour, will only be scheduled at the beginning of the next hour. This situation can be improved by choosing a smaller scheduling

[2] The time period of one hour is chosen to make the computing process tractable. The scheduling interval can be arbitrary.

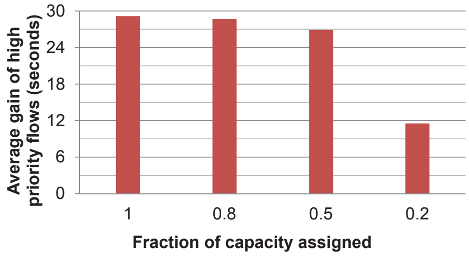


Figure 37: Gain of high priority flows for the modeling based approach.

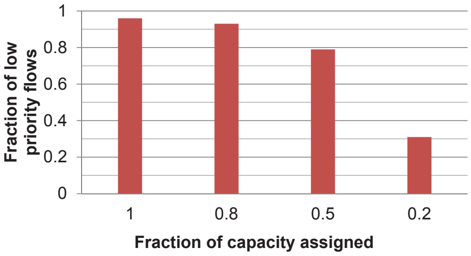
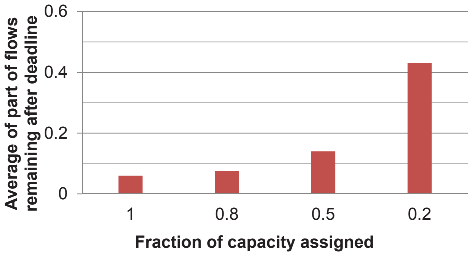


Figure 38: Fraction of low priority flows finishing within deadline.

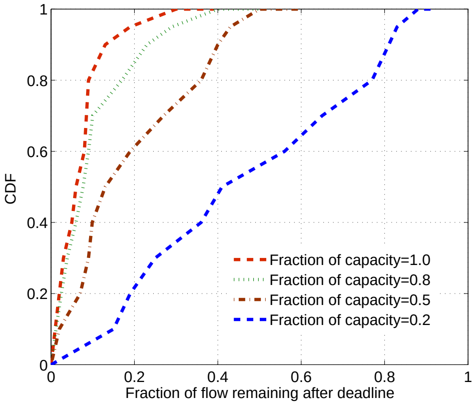
interval. After the schedule computation, the agent will start a low priority flow according to the schedule at the location of the corresponding subscriber only if the base station's current real load is lower or equal to the predicted load of that base station at that time instance.[3] At each scheduling event, we consider all the low priority flows: either scheduled or newly arrived. This helps the approach to overcome any modeling error.

Figure 37 shows the average gain of the high priority flows for different capacity assignments of the base stations. The average gain of high priority flows is around 27 seconds when the capacity of base stations is made half.

[3] Note that this does require checking on the monitored load in real time. But this needs to be done only when a scheduled low priority flow is about to be started. If load estimations are sufficiently accurate, such checking may be avoidable.



(a) Average fraction of flow size remained after the deadline.



(b) CDF of fraction of flow size remaining after the deadline.

Figure 39: Fraction flow-size of lows priority flows remaining after deadline.

This is similar to what we have observed for greedy approach. This is understandable as in both cases the high priority flows are benefiting in similar fashion with more available resources. As the low priority flows are scheduled by the deadlines, the delay of the low priority flows may not be interesting to analyze. On the hand, it may be interesting to see what fraction of low priority flows gets finished by the deadline. Figure 38 shows about 80% of the low priority flows are finished by the deadline specified for the case when base station capacity is made half of its original capacity. This shows the potential of the scheduling approach. We also investigate what fraction of each flow served after the deadline. For each of the low priority flows that cannot meet the deadline, we calculate the fraction of flow size in terms of number of bytes served after the deadline. Figure 39 shows the average fraction of flows remining after the deadline. Note that the remaining portion is not significant (about 15%) even when the capacity is made half.

Note that even with the assignment of full capacity, that is, with fraction of capacity = 1, a small fraction of low priority flows can not meet the deadline. Our investigation suggests that this is due to the modeling and prediction error. Moreover, the deadlines of the low priority flows are picked randomly and is not correlated to original flow duration.

4.4.5 Critique

The modeling based approach is more practical as it does not require any real-time load measurement Instead it relies on historical data to derive load and mobility estimates. It considers the global network-wide scenario as opposed to the previous greedy approach where each base station is treated in an isolated fashion. Many wireless providers do collect subscriber/base station specific load information in various forms for network monitoring. Thus, off-line use of such data to create profiles as used in the above evaluation is entirely plausible. Scalability of global scheduling can still be an issue. But the network can always be partitioned in smaller parts and scheduling can be done in each of these parts independently to address scalability issues.

4.5 Discussion

We now summarize our key observations and identify important practical implications both from the perspectives of the network provider and the subscriber:

1) Better Service: Normally subscribers may experience poor service during the peak periods due to network congestion. Flows can be dropped or served with very poor rate due to congestion. A frustrated subscriber can try multiple times to initiate communication possibly leading to more congestion. Existing networks do not have any built-in mechanism for service differentiation and treats all flows equally. The proposed mechanisms provide a higher-layer, agent based mechanism to provide service differentiation based on a simple prioritization of flows. We have shown that existing load can be served even with half the capacity with only modest delays on the low priority flows and little or no negative impact on high priority flows, and sometimes with some positive impact.

2) Subscriber Pricing and Incentivization: Providers are moving away from unlimited data plans and replacing them with tiered plans as cellular data networks are becoming more popular. This is evidently focused towards managing the network load better. With the opportunistic scheduling, the providers could provide incentives to subscribers to tag (automatically via apps, or via a profile driven approach, or even manually) flows as low priority. A possible incentive could be that low priority flows are not metered to count as a part of total data usage by the subscriber. This provides a semblance of unlimited data plan to the subscriber and may attract more customers to the provider's network.

3) Reducing Resource Requirement: Our analysis with both the approaches shows that the resource requirement of base stations can be reduced significantly considering only a small fraction of flows with low priority. We believe that this can be reduced even more if the fraction of low priority flows increases. The service provider can utilize the spare capacity to accommodate more high priority flows, in other words, more new subscribers in the network.

4.6 Related Work

Our work in this chapter has some level of similarity with the broad topic of quality of service scheduling and load balancing, as we propose to move low priority flows both spatially and temporally. This general idea has been widely used where wireless resources are redistributed in form of channel assignment rather than traffic [44, 55, 91]. A large body of work on scheduling approaches on link layer is also available [136, 90, 49, 43]. In contrast, our work deals with the load shifting problem at a higher layer and at the flow level. We focus on scheduling of flows, specially low priority flows either in parts or in whole ignoring the low level issues such as power, interference, radio resources, packet level scheduling. Similar load shifting studies have been done in other contexts, such as power savings (see, e.g., [109, 59]).

There are different pieces of work dealing with the priority scheduling in wireless networks. The authors in [140] have proposed a technique to set priority among the source stations in ad hoc network. The authors in [76] have devised an distributed priority scheduling in packet level for the nodes in ad hoc networks. The authors in [80] have modeled the arrival of flows in a base station as a queueing model with priority set between its own flows and flows arrived because of handoff. Our work is different from these set of works as we set priorities on the flows by using application layer information and deal with the opportunistic scheduling of low priority flows. A similar work, but in a different context has been done targeting TCP, where the authors have developed a variation of the regular TCP, called 'TCP-low priority,' in order to to utilize excess network bandwidth distributedly as compared to the fair-share bandwidth in regular TCP [86].

4.7 Conclusion

In this chapter, we have explored an avenue to reduce the peak load in cellular data networks. The idea is to treat certain flows as low priority and delay scheduling such flows if the base station has reached its capacity limits. Low priority flows are to be scheduled opportunistically based on the available capacity. The main goal of this model is to move traffic from the peak periods to off-peak periods that potentially reduces the average-to-peak ratio of load under base stations. We have presented two approaches to schedule the low priority flows. The first one is a straightforward greedy approach, but needs continuous monitoring of base station load in order to determine scheduling opportunities. The second one is a modeling based approach where models are created to predict subscriber location (base station) and base station loads based on historical data. This approach reduces the need for monitoring, but can potentially suffer from inaccurate estimates. Our analysis indicates that the capacity requirements at the base station can be reduced significantly – by as much as a factor of two – with only modest added delays on the low priority flows. If low priority flows are those that are long-lived and delay-tolerant such delays would be perfectly acceptable to the applications, but would be beneficial for addressing the data overloads in the base stations. Further, this will help the providers to accommodate more subscribers without augmenting the network capacity. Our future work will also involve incentive and pricing schemes to make this realistic. The future work will also consider the actual design of the agent-based system that can perform the opportunistic scheduling proposed in this study.

Chapter 5

Learning Probabilistic Models with Applications to Resource Management

5.1 Introduction

With the growing popularity of broadband capable mobile devices, an increasing share of the world's end-user network traffic is being carried by broadband wireless networks. Managing this explosion in traffic volume and ever growing expansion of cellular network has now become a challenge for the service providers. It needs constant monitoring of the network via large-scale measurement and analysis. The measurement and analysis themselves are complicated as large volumes of data from large geographically distributed networks need to be collected, possibly at various layers of protocol stack. Scalable techniques to collect such data is challenging with ever increasing traffic volumes and data rates. From early days of Internet measurements, 'sampling' has been widely used to reduce the data collection efforts for measurement data [144, 107, 38, 70, 52]. While many of these techniques are also relevant for cellular data, the nature of traffic in a large-scale, geographically distributed wireless network is much more nuanced. For example, it is intimately related to base station locations, wireless propagation, wireless resources/capacity, subscriber mobility and characteristics of the end devices. These provide a degree of richness where interesting modeling problems can be posed and solved to address various challenges.

The case in point is building of a spatial model where we seek probabilistic relationships among the base station loads across space. Using statistical machine learning tools we build such a model using measurement data and demonstrate how this model is useful in 'spatial sampling,' – i.e., describing the loads on *all* base stations given the measured[1] loads only on a small subset of base stations. Obviously, for any form of model-based data reduction approach such as this there is a trade-off between accuracy and complexity of data collection. It is not always clear what type of accuracy is appropriate unless it is tied to an application. Thus, we also study the applicability of this model for two applications – (i) energy savings [59] and (ii) opportunistic scheduling [103].

5.1.1 Modeling Conditional Dependencies

The general modeling approach uses statistical machine learning tools. We assume that the base station loads are time sequences of multi-variate Gaussian random variables. We do ensure that the Gaussian assumption is reasonable in our context. Then we estimate the 'inverse covariance matrix' of this multi-variate distribution using the given training data of base station load measurements. The inverse covariance matrix essentially describes the model. Zero matrix elements denote 'conditional independence' among the related variables denoting lack of influence. A sparser matrix provides a model that is more readily interpretable and can possibly explain various forms of influence among the variables (base station loads) useful to the network designer. They also tend to have less variance in the sense that different training data is less likely to change the model substantially. To this end, machine learning literature has a strong focus on building relatively sparse models that also adequately explain the data [67].

[1] We use the terms such as 'measured,' 'observed,' 'monitored,' and 'sampled' interchangeably.

This estimation of the inverse covariance matrix is computationally involved for large dimensional data (the number of variables is equal to the number of base stations that could be large) [145]. Various computational problems also arise in the estimation process when the number of samples is not adequate [130]. With some experiments with various other techniques, we have found that the technique based on *Least Absolute Shrinkage and Selection Operator* or 'Lasso' [134] is well-suited for our modeling problem. Lasso is commonly used in connection with linear regression for 'regularization.' At a high level, regularization prevents model overfitting for the given training data by introducing a penalty function to encourage creation of sparser models, i.e., models that describe the dependent variables adequately but using a smaller number of independent variables. Prior work has shown that Lasso effectively estimates a relatively sparse inverse covariance matrix [100, 135].

5.1.2 Estimating Base Station Loads and Applications

Once the model is learnt, it is used to estimate the base stations' loads. The idea is that the load of only a subset of the base stations will be monitored, the model will estimate the load of the rest of the base stations. The overall accuracy of the prediction depends on the number of observed base stations as well as the individual base stations that are monitored. We propose a simple greedy heuristic to choose the subset of base stations that should be monitored to improve the accuracy of the prediction. We also investigate whether base stations with certain property should be sampled to improve the overall accuracy of the prediction. Finally, we analyze how much accuracy is important. To this end, we study two different resource management applications — (i) energy savings [59] and (ii) opportunistic scheduling [103]. Both applications normally require monitoring of base station loads. Thus, studying impact on the applications when the loads are partly estimated and only partly measured is a good way to understand how much estimation error is acceptable. Our study shows very minor impact even when the loads on a relatively small subset of base stations are actually measured.

The rest of this chapter is organized as follows. The modeling approach is described in Section 5.2. The data set and the learning steps are presented in Section 5.3. We demonstrate the load estimation based on the model and present an approach to select the set of base stations to sample in Section 5.4. We present the effect of accuracy of the prediction from the application's point of view in Section 5.5. The related works and conclusions are presented in Sections 5.6 and 5.7, respectively.

5.2 Modeling Approach

5.2.1 Basics

The loads on the set Γ of n base stations are assumed to be modeled via a multi-variate Gaussian-distributed random variable

$$X = (X_1, \ldots, X_n) \sim \mathcal{N}(\mu, \Sigma),$$

where μ is the mean vector and Σ is the covariance matrix. The broad goal is to estimate the 'inverse covariance matrix' Σ^{-1}, given a set of p observations of this multi-variate random variable (training data). The inverse covariance matrix is a powerful modeling tool. One can learn the conditional independence structure of the distribution from this inverse covariance matrix. Assume $\Gamma = \{1, \ldots, n\}$ is the index set denoting the base stations. If $\Sigma^{-1}_{ij} = 0$, then the load variables X_i and X_j are conditionally independent, given the other variables $X_{\Gamma\setminus\{i,j\}}$. Additionally, a load variable X_i, given the set of variables X_{N_i}, such that $N_i \subseteq \Gamma \setminus \{i\}$, $j \in N_i$ and $\Sigma^{-1}_{ij} \neq 0$, is conditionally independent of rest of the variables [58].

In our work, we would like to estimate the load of all base stations in Γ given the load of a subset, say S. We would do this simply by exploiting the structure of their inter-dependence expressed via the Σ^{-1} matrix. The estimate for base station i, given the load X_S of base stations in S, $i \notin S$, can be computed as the conditional mean [16]:

$$\mu_{i|S}(X_S) = \mu_i - \Sigma_{ii}\Sigma^{-1}_{iS}(X_S - \mu_S). \tag{7}$$

Note that the covariance relationship is required to calculate the conditional mean. The following steps are used.

1. The model is learnt by estimating the inverse covariance matrix using the data set of p observations (training data). The 'principle of persimony' in machine learning suggests use of the simplest models (i.e., using the minimum number of variables that adequately explain the observations) [40]. This reduces model complexity as well as expected model variance. In this context, if the inverse covariance matrix is relatively sparse, it provides a more 'interpretable' model as conditional dependence of a variable is now on fewer number of other variables. Statistical machine learning literature has considered this issue vigorously and several methods have been developed. We use the concept of the *Least Absolute Shrinkage and Selection Operator (Lasso)* [134] to learn the model. More will be said about the Lasso in Section 5.2.2.

2. Once the covariance matrix is learnt, it can be used estimate the load of all base stations using only observations from subset, S, by using Equation (7). The subset size, $|S|$, plays an important role in the accuracy of the estimation. It is also important to select S carefully, as some base stations may play a greater role in the estimation that the others. We will present our approach of selecting base stations in Section 5.4.

5.2.2 The Lasso

In statistics and machine learning, linear regression analysis focuses on modeling linear relationships between a dependent variable and one or more independent variables or predictors given a set of observations. Ordinary least square (OLS) estimates have been very popular in linear regression. They are obtained by straightforward minimization of the residual squared error. However, OLS suffers from poor prediction accuracy (typically produces large variance even though bias is low) and inability to produce a sparse model (i.e., all variables are included in the model) [67].

A significant research effort has gone in the machine learning community to improve on the OLS estimate. The 'Least Absolute Shrinkage and Selection Operator' or the 'Lasso' [134] has emerged as a robust and computationally efficient technique that can provide good balance between bias and variance while producing a sparse model. The Lasso works by jointly minimizing the empirical error and imposed penalty. The penalty is formulated using a 'regularization' term that uses a parametrized L_1 penalty function that can lead to sparse solutions. However, obtaining such models through classical model selection methods usually involves heavy combinatorial search. The Lasso, however, is computationally efficient even when a large number of variables and observations are involved [145, 124], such as in our case. Lasso can also be very effective in situations where the number of predictors or variables are much larger than the observations. This has been shown in the case of genomic data [130, 60]. This is important as models may need to be created with only short term observation data available.

Assume the standard representation of linear regression:

$$Y_p = \mathbf{X_p}\beta^p + \boldsymbol{\varepsilon_p}, \tag{8}$$

where Y_p is the $p \times 1$ response matrix of the observed variable and $\mathbf{X_p} = (X_1^p, X_2^p, \cdots, X_n^p)$ is the $p \times n$ design matrix where X_i^p is the set of p samples of ith predictor variable. $\boldsymbol{\varepsilon_p} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_p)^T$ is a column vector of i.i.d. random variables with mean 0 representing measurement noise. β^p is the column vector of model coefficients that to be estimated via regression. The Lasso estimates β^p as

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \|Y_p - \mathbf{X_p}\beta\|_2^2 + \lambda\|\beta\|_1, \tag{9}$$

where $\|.\|_1$ and $\|.\|_2$ stand for the L_1 and L_2 norms of a vector. The parameter λ controls the amount of regularization applied to the estimate. It is easy to see that using $\lambda = 0$ reverses the Lasso problem to OLS. On the other hand, a very large λ completely shrinks $\hat{\beta}$ to 0 thus leading to the empty or null model. In general, moderate values of λ cause shrinkage of the solutions towards 0, and some coefficients may end up being exactly 0.

In an n-dimensional multivariate case, the model can be created by fitting a Lasso model to each variable, modeled as response, using the others as predictors. Denote the model coefficients now as an $n \times n$ matrix $\hat{\boldsymbol{\beta}}$, where each row $\hat{\beta}^i$ denotes the vector of coefficients used for modeling the i-th variable X_i. It can be shown that the elements of $\hat{\boldsymbol{\beta}}$ are determined by the inverse covariance matrix Σ^{-1} [96]. Specifically, $\hat{\beta}^i_j = -\hat{\Sigma}^{-1}_{i,j}/\hat{\Sigma}^{-1}_{i,i}$, $i \neq j$. ($\hat{\beta}^i_j = 0$, for $i = j$.) This provides a mechanism to estimate $\hat{\Sigma}^{-1}$ given estimate of $\hat{\boldsymbol{\beta}}$.

5.2.3 Regularization

The model provided by the Lasso includes the regularization parameter, λ. It is important to choose the right value of λ. We use a standard validation approach to determine λ using the 'likelihood' estimate. Given a model and some observed outcomes, the likelihood L, measures the probability of generating those observed outcomes by that model. As standard practice, we use the concept of log-likelihood [11]. Given a multivariate Normal distribution $\mathcal{N}(\mu, \Sigma)$ and a set of observed samples $\mathbf{X}$, the log-likelihood is determined by:

$$\ln(L) = \frac{1}{2}[-\ln|\Sigma| - n\ln(2\pi)] - \frac{1}{2p}\sum_{l=1}^{p}(X^l - \mu)^T\Sigma^{-1}(X^l - \mu). \tag{10}$$

Here, p denotes the number of samples from each variable, n is the number of variables and X^l denotes the vector of l-th sample from all variables. The idea is to do a cross-validation for each λ by using a small part of observations set aside for this purpose (and not used in model building). The model corresponding to the λ that maximizes the log-likelihood is to be chosen.

5.3 Data Processing and Learning

We use the same data set used in our earlier work [105, 106] presented in previous chapters. The data set consists of flow-level data (TCP or UDP) collected at the core of a nation wide cellular network for a one week period. We use a set of 400 base stations covering a 75×84 miles geographic region that include both a busy downtown and surrounding sub-urban areas. The available data spans only 1 week. Since we have observed in our earlier work [105, 106] that network behavior is significantly different in the weekends compared to the weekdays, we focus on weekdays only in this work. (Use of weekends will limit us to much fewer samples).

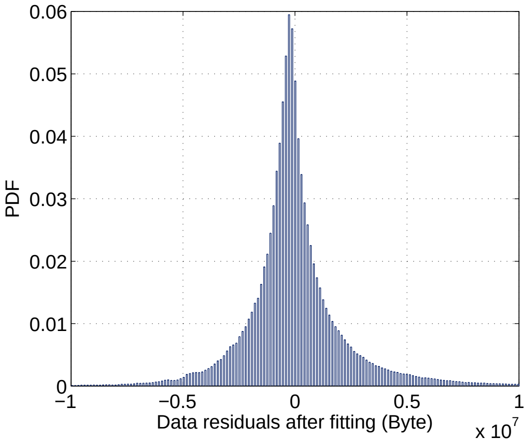
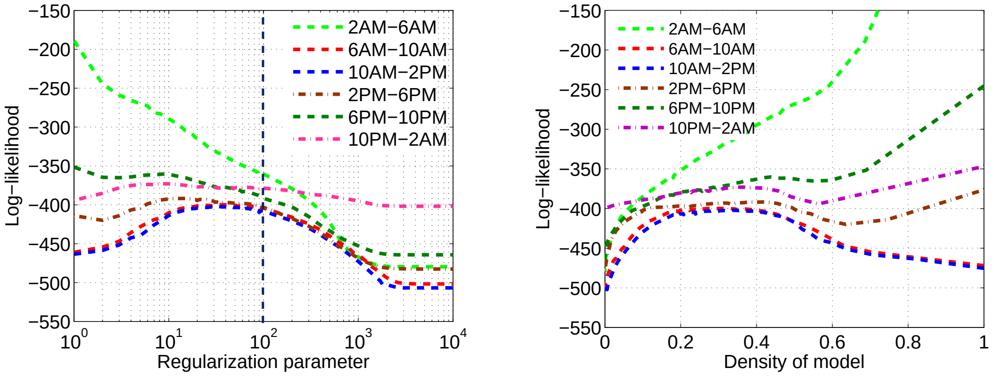


Figure 40: Distribution of the residual loads after the fitting for all base stations combined.

5.3.1 Preprocessing

The load variables that are input to the model are simply the volume of data seen by each base station at given intervals. A 2 hour interval is used for this study. Thus each base station load is a simple time series $\mathcal{Y}_i$ of length $12 \times 5 = 60$ covering the 5 day period. However, 60 samples are considered too few to construct a robust model of dimension 400. Thus, a bootstrapping method is used to generate more samples that has the same statistics. To generate such samples, the load is again summarized as above, but now at a much smaller – 10 mins interval. A new 2 hour load sample is formed by using 12 samples randomly chosen from these 10 mins interval samples and then summed up. Only those 10 min interval samples are used to draw from that are within the same 2 hour period. In other words, 10 min samples from an 8-10AM period are used to generate a synthetic sample of load in that period only. This method provides a long time series of load variables, Y_i, for every base station i every 2 hour period for all weekdays for an assumed 1 year period. These bootstrapped samples are used only to learn the model. The real data set $\boldsymbol{\mathcal{Y}} = \{\mathcal{Y}_i\}$ is still used for evaluation.

Prior work has observed that traffic loads in cellular networks have strong



(a)

(b)

Figure 41: (a) Log-likelihood of models obtained by Lasso for different values of the regularization parameter λ. Separate models are created for different parts of the day. (b) Log-likelihood plotted versus model density.

diurnal variations [105], peaking during mid day and having a lean period during the middle of the night. This pattern obviously is not symptomatic of i.i.d. Gaussian assumptions that the modeling approach is based on. To address this, we fit a load curve to each base station with a time dependent function ${f_i}^{(t)}$ and then calculate the residuals as $R_i = \{{R_i}^{(t)} = {Y_i}^{(t)} - {f_i}^{(t)}\}$.

Figure 40 shows the PDF of the residuals of all the base stations combined. It shows the expected ball-shaped form and can be assumed Gaussian. We have also confirmed this using the Jarque-Bera hypothesis test [10] of normality. The residual data of each base station, R_i, is now normalized so that the mean and the standard deviation of each base station load sample become 0 and 1, respectively. This normalized data set (say, $\mathbf{X} = \{X_i\}$) is now used to learn the model.

One more step remains. It is unclear at the outset whether a single model representing all times in a day is appropriate. This is simply because the influence structures in the model could vary. We found some preliminary evidence of such temporal dependencies in our earlier work, albeit in other contexts [106]. Note also that the modeling approach we are considering does not capture any temporal aspect in spite of the use of time series as an input. In particular, the order of the multi-variate samples $X^{(t_1)}$ and $X^{(t_2)}$ are immaterial

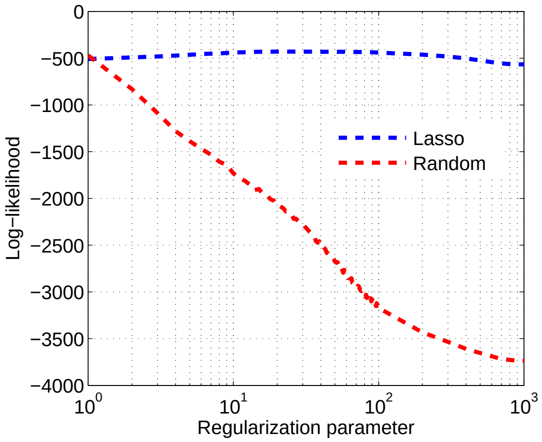


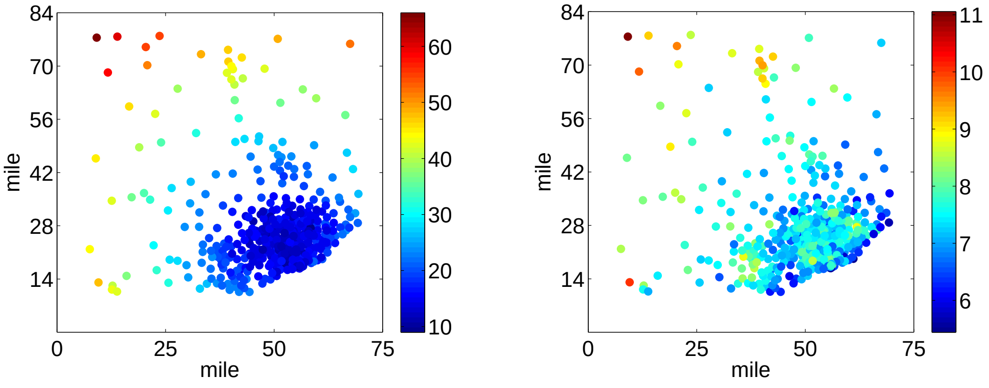
Figure 42: Comparison of the model obtained using Lasso technique to randomly constructed model. Only the 10AM-2PM period is considered here.

to the model. To ensure that errors do not creep in due to temporal changes in the model, we create separate models for each 4 hour period of the day.[2] Thus, essentially we will learn 6 separate models instead of one single model. We split the data set $\mathbf{X}$ accordingly.

5.3.2 Learning

Now, we learn the model separately for each 4 hour period during the day. This step essentially estimates the regression coefficients $\hat{\beta}$ per Equation 9 for each base station i using the rest of the base stations as predictors (see the last para in Section 5.2.2). Computational procedures as described in [17] are used. We do this for a range of values of the regularization parameter λ and then perform a cross-validation as described in Section 5.2.3. Figure 41(a) shows the log-likelihood for different λ values for different 4 hour periods of day.

[2] The 4 hour assumption is somewhat arbitrary at this point. The assumption is that the model does not deviate significantly during this 4 hour.



(a) (b)

Figure 43: Average edge length in the model graph for each base station in the network. The base station location is shown in the geographic space and the edge length is color coded. (a) Edge length is distance in miles between the pair of base stations. (b) Edge length is in number of hops in the shortest path between the pair of base stations in the Voronoi graph of the base station locations.

To choose the right value of λ we study the tradeoff between model accuracy (represented by log-likelihood) and model complexity. We capture complexity by a density measure, defined as the number of non-zero terms in the inverse covariance matrix divided by the matrix size. Decreasing λ will obviously increase density, but should generally improve the log-likelihood. Our goal is to empirically look for a λ value that provides a low density but with still a small enough value of log-likelihood. Figure 41(b) shows the log-likelihood versus density plot. The subplots (a) and (b) together show that beyond $\lambda \approx 10^2$, reduction of λ unnecessarily increases density with little improvement in log-likelihood. Thus, this value is used for further analysis. The density for this choice λ is quite small, about 0.15.

As an added validation, we also want to establish that the Lasso is doing something 'useful,' in particular, it is not simply choosing an arbitrary set of coefficients to make non-zero. To establish this, we compare the Lasso-based technique with a random model. The random model simply assumes that each base station is conditionally dependent on only a random subset of base stations and thus makes only the respective entries in the inverse covariance matrix non-zero. For a given choice of λ the model densities are kept equal, i.e., the size of the random subset is same as the number of non-zero estimates as determined by Lasso. The value of non-zero estimates in the random model is determined by standard covariance estimation [7]. Figure 42 shows the effectiveness of Lasso technique with respect to the random model in terms of log-likelihood. Here we have shown only one time period, 10AM-2PM for brevity. Note that the log-likelihood of the randomly constructed model goes down very fast with the increase of λ compared to the model obtained using Lasso. This demonstrates the effectiveness of Lasso in choosing the set of predictors and the coefficients.

5.3.3 Model Analysis

Here, we analyze some properties of the model we just learnt. For a better understanding, the model can be represented as a graph $G = (V, E)$ where each node, $i \in V$ represents a base station and there is an edge $(i, j) \in E$ if and only if $\Sigma^{-1}_{ij} \neq 0$. This is a standard representation in the field of probabilistic graphical models [96].

To understand the structure of conditional dependency, we calculate the average edge length for each node. We use two definitions of edge length for each edge – (i) physical distance (in miles) between the two base stations, (ii) distance in number of hops in the shortest path between the two base stations in the Voronoi graph[3] of base station locations. This can be more appropriate sometimes if the spatial density of the base stations vary widely. The hop-wise shortest path distance captures distance in the physical neighborhood graph. Figures 43(a) and (b) show the locations of the base stations in the map and color codes its average edge length using the above definitions. Note that densely populated region (downtown area) has a relatively smaller edge length relative to the sparsely populated region (rural/suburban) using either definition. This observation is in line with our conclusions in prior work that

[3] Two base stations are adjacent in the Voronoi graph if their corresponding Voronoi cells have a common contiguous boundary.

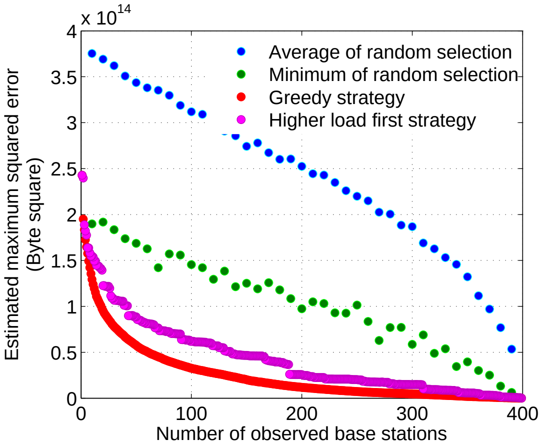


Figure 44: Errors for various base station selection strategies. Expected maximum squared error is shown for the greedy strategy, average and minimum squared errors for the random selection and expected maximum squared error for the load-based strategy.

spatially significant influence structures exist in cellular network traffic [105, 106].

5.4 Estimating Base Station Loads

In this section we use the models learnt in the previous section to estimate the load of the base stations. Recall that our goal is spatial sampling, i.e., measure loads on a subset of the base stations to estimate the rest. We present an approach to select the subset of base stations that should be sampled and evaluate the error in predicting the load of the rest of the base stations.

Before we go forward, note that the models we have learnt so far are in terms of the normalized residuals ($\mathbf{X}$). See Section 5.3.1. Simple transformations will produce it for actual residuals (R_i's) and then the actual loads (Y_i's). Thus, assuming that the distribution of the normalized residuals is given by $\mathcal{N}(0, \Sigma^{x(\imath)})$ where $\imath$ denotes the time period. The distribution $\mathcal{N}(\mu^{y(\imath)}, \Sigma^{y(\imath)})$

for the period $\imath$ for the real data is obtained by doing the following transformations.

$$\Sigma^{y(\imath)} = A \cdot \Sigma^{x(\imath)} \cdot A^T, \tag{11}$$

$$\mu^{y(\imath)} = \mu^{r(\imath)} + \mu^{f(\imath)}. \tag{12}$$

Here A is a diagonal matrix with the standard deviations of the residual data, $\mu^{r(\imath)}$ is the mean vector of the residual data $\{R_i\}$ and $\mu^{f(\imath)}$ is the mean vector of the fitting function. We will now use this distribution to select which base station should be sampled and evaluate the prediction error for the rest of the base stations.

5.4.1 Selecting Base Stations to Sample

Given samples $\mathbf{X_S}$ from the base stations in set $S \subset \Gamma$, we want to estimate the load of all other base stations in $\Gamma \backslash S$ as conditional means using Equation (7). Now, the question is which base stations to select for S. A smaller set means less monitoring complexity at the risk of higher estimation errors.[4] To evaluate our approach, for each selection of S, we calculate the estimated maximum squared error, $Err_{\text{max}} = E[max_{i \notin S}(Y_i - \mu^{y(\imath)}_{i|S}(Y_S))^2]$, over all base stations i to be estimated. Here Y_i is the load of base station i. Our goal is to determine, for each selection of $|S|$, the actual S that minimizes the above Err_{max}.

The approach we undertake uses simulation and a brute force search. We reduce the search complexity using a greedy selection heuristics. For the simulation, a set of samples for each base station is generated using the distribution (obtained using equation (11) and (12)). The greedy heuristic works as follows. Initially, S is an empty set. For a given value of $|S|$, the heuristic gradually expands S by inserting one new base station in S at a time until the limit of $|S|$ is reached. In each iteration, the heuristic picks every other base station that is not already in S and estimates the maximum error Err_{max} if the new base station added to S. The inclusion of the base station that minimizes the estimated error is actually inserted in S.

[4] The set S may not be the same for different time periods as different models are learnt for different time periods.

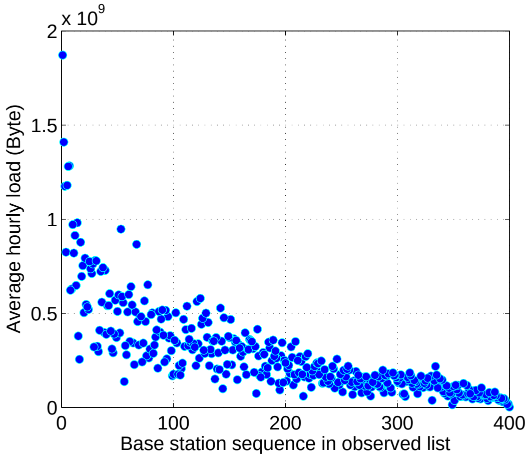


Figure 45: Scatterplot of the rank of a base station in greedy heuristic and its average load.

Figure 44 shows the maximum error for this greedy approach for different sizes of S. For brevity, we show the plot for only one period, 10AM-2PM. We also investigate how the random selection of the base stations perform compared to the greedy heuristic. Here, base stations are selected randomly for S. For each size of S, we repeat the process of random selection 500 times. Figure 44 also shows the average and minimum expected squared error for different size of S. Note that greedy approach is significantly better than the random selection. Moreover, the estimated error reduces sharply with the increase of the size of S.

We also investigate whether there is any property that makes a base station more likely to be picked in the set S to be monitored. To do that we consider the sequence in which base stations are inserted into the set S in the greedy heuristic approach. We call this the 'node rank'. The base station that is selected first has node rank $= 1$ and other base stations have higher ranks. Figure 45 shows the node rank of each base station and its average load. Note that there is a tendency of choosing heavily loaded base stations first. To delve more into this, we calculate the estimated maximum squared error as before where the set S is selected based on the average load of the base stations. We

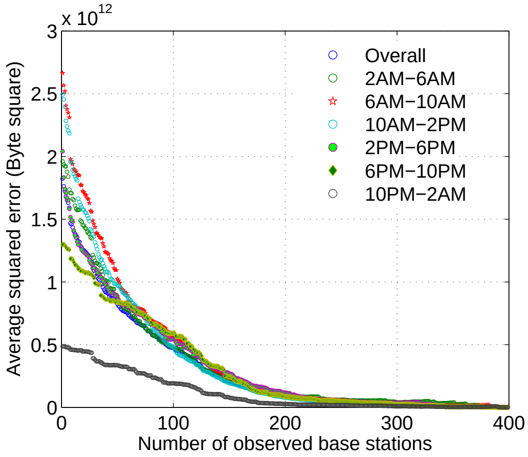


Figure 46: Average squared error in estimation considering the original data set.

plot the results back in Figure 44. Note that the error is only incrementally worse than what we observed for greedy approach for different sizes of S. This is of significant practical relevance, as operators can simply choose the monitor the k most highly loaded base stations if they can afford to monitor only k base stations.

5.4.2 Overall Accuracy of Estimation

So far, we use the samples generated from the distribution to select the base stations to observe. We now evaluate the accuracy of the estimation. To do that we use the original data set $\boldsymbol{\mathcal{Y}}$ and base station selection sequence by the greedy approach. We calculate the squared error as $\frac{1}{T}\sum_{t=1}^{T}[\frac{1}{n}\sum_{i=1}^{n}(\mathcal{Y}_i^t - \mu_{i|S}^t(\mathcal{Y}_S^t))^2]$. Here $\mu_{i|S}^t(\mathcal{Y}_S^t)$ denotes the conditional mean of base station i given the observations of base stations in S for the time instance t. Different models and different sequence of observed base stations are used for the data of different parts of the day. Figure 46 shows the overall error for different sizes of S. The errors at different time periods are also shown. Errors are generally higher during the day while they are much lower during the night (10PM-2AM). Generally speaking, the errors become very small when

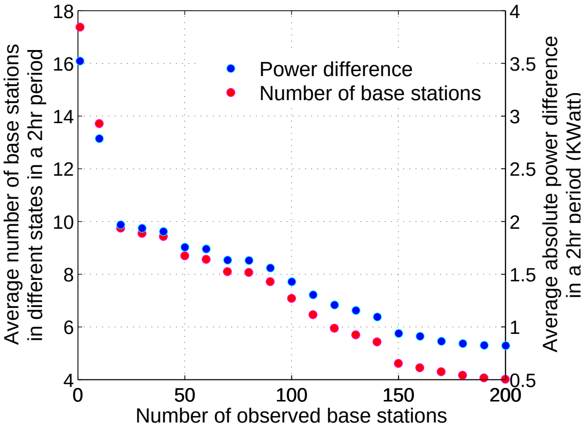


Figure 47: Comparison of the outcomes of the energy saving algorithm [59] when using (i) original data set and (ii) original data only for a set of base stations and estimations for the rest.

at least 50% of the base stations are sampled. This is very encouraging as this shows the possibility of cutting down monitoring effort by at least half. We will show in the next section that practical applications can work with even lesser sampling.

5.5 Impact of Estimation Accuracy on Applications

In the proceeding sections we have developed and analyzed a probabilistic model of the traffic load on cellular base stations that enables the cellular provider to sample the load only on a subset of base stations to estimate the load on the others. The goal is to reduce the complexity of the monitoring infrastructure. As expected, the estimation error increases when the sampled subset is small. See Figure 46. However, the tradeoff presented in this figure does not directly say what subset size is appropriate, or in other words, what type of estimation errors are tolerable. Obviously, there is no general answer to this question as this is intimately related to the applications where the load estimations are actually used. In this section, we use two resource/traffic management applications that rely on load measurements and analyze the impact on their performances when some of the actual measurements are replaced by estimations. This sheds light on the appropriate size of the sampled subset. For example, some applications may be relatively robust to estimation errors, where a smaller subset could be appropriate. Others may have entirely different characteristics.

5.5.1 Application 1: Energy Savings

The first application we study uses load information to save energy expenditure at the base stations [59]. Energy is saved by turning off a subset of base stations and adjusting the transmission power of the other base stations so that there is no coverage hole. The constraint here is that no base station gets overloaded. We use the algorithm presented in [59] where power is allocated adaptively, depending on the users' demand. Given the load information and the number of subscribers under each base station, the algorithm decides which base stations to turn off and adjusts the transmission power of other active base stations so that all subscribers' demands in terms of throughput requirements are met. Since the subscriber locations are not available in the data set, the evaluation uses the assumption that the locations are distributed randomly in the originally associated base station's coverage area. The channel quality and capacity are calculated based on the SINR model and Shanon's Law. These assumptions are not central to the technique, but are made in our work as lower layer information is not available in our data set.

The algorithm is run twice – (i) once with the original data set $\boldsymbol{\mathcal{Y}} = \{\mathcal{Y}_i\}$, the five-day long load information of each base station for each two hour period, and (ii) again assuming that only a set of S base stations uses original data and the rest use estimates based on the described approach. Different values of $|S|$ is used for the evaluation. For each value, the actual selection of S uses the greedy approach discussed in the previous section. Based on the load inputs – either actual or estimated as may be the case – the algorithm makes a decision, at the granularity of 2 hours, whether the base stations should be on or off, and if on what would be the transmit power level of the base station. The differences in the algorithm's outcome will depend on estimation errors. Assuming that the ideal outcome is the one obtained with the original data, the difference in the outcome with the estimated data shows the impact of the estimation error.

We represent the difference in the two outcomes using the average number of base stations that are in different states (on or off) in each 2 hour period Figure 47 (using the left Y-axis) show this for different size of the selected set S. Note that out of 400 base stations, only 18 base stations are in different states when just one base station is sampled whereas 8 base stations are in different states when 50 base stations, i.e., one-fourth of the total number of base stations are sampled. This difference becomes miniscule (only 4) when half of all base stations are sampled. This demonstrates the accuracy of our prediction for different size of observed base stations in terms of the decision making from the application perspective. We also calculate the total absolute power difference of base station power assignments for each 2 hour period and Figure 47 (using the right Y-axis) shows the average. As the power required to keep a base station on (about 200W) is significantly higher than the power requirements for transmission, absolute power difference has a similar nature as number of base stations in different states. Based on this analysis, the provider now can decide about the tolerable level of error in power saving and monitor the appropriate set of base stations to achieve the required level of accuracy.

5.5.2 Application 2: Opportunistic Traffic Scheduling

Now, we consider another application, opportunistic traffic scheduling in the context of cellular data network [103]. The idea here is to reduce the peak-traffic load and reduce the peak-to-average load ratios in cellular data networks by using a higher-layer priority-based scheduling. Assume that all flows (TCP or UDP) to or from the mobile device are tagged either high or low priority. This tagging can be done manually or automatically based on a profile, depending on types of applications generating such flows. If the estimated load at the

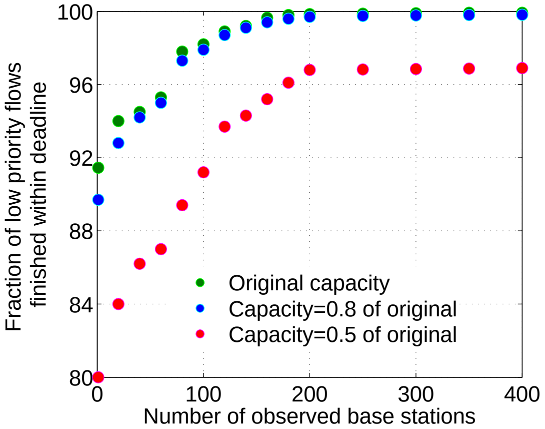


Figure 48: Fraction of low priority flows finishing within deadline in the opportunistic traffic scheduling application [103].

associated base station is high enough (e.g., near capacity) low priority flows are delayed and opportunistically scheduled later when the estimated load is low. A higher layer agent on the infrastructure side can do such scheduling based on load estimates. It uses specific higher layer signalling with its counterpart running on the mobile device to do the actual invocation/suspension of flows. High priority flows, on the other hand, are served immediately as is normal in today's network. In the specific technique we consider [103] a notion of 'deadline' is used for the low-priority flows to aid in scheduling. Note that essentially the technique moves low priority flows temporally and also spatially (if the user moves) from peak to off-peak periods. The general goal is to serve the same load with lesser network capacity (or conversely, serve more load with the same capacity).

One of the approaches to schedule the low priority flows in [103] uses load prediction. In this approach, base stations' loads and subscribers' mobility are predicted using appropriate models. We have shown in [103] how the scheduling problem now essentially boils down to solving a network flow problem. Needless to say, the effectiveness of the approach depends on the accuracy of the predictions. This is because scheduling decisions are taken at course-grain time epochs based on estimates of future loads and scheduling decisions cannot be altered based on real-time measurements in order to keep the complexity of the system manageable.

We now evaluate the effect of prediction accuracy from the perspective of this application. The predicted load of base stations using our approach is provided to the scheduler. Different sets of predicted data are used for different number of sampled base stations. We run a trace driven simulation on our data set for a selected set of low priority flows and scheduling is done based on the predicted data. All relevant details appear in [103]. Since the interest here is only to evaluate the accuracy of load prediction, we assume that mobility information of the subscribers are known and need not be predicted. This is to avoid any error influencing the results due to errors in mobility prediction.

The performance measure here is the fraction of low priority flows that finishes within the specified deadline. Figure 48 shows this information for different number of observed base stations. Multiple plots are shown with artificially reducing the base station capacities. This is to evaluate whether a lower network capacity can still carry the offered load effectively. Note that more low priority flows finish by their deadlines as the accuracy of the prediction improves when more base stations are observed. Note also the impact of reducing the number of observed base stations is very small. For example, for capacities 100% and 80% of the original the error is within a few percent even with just one-quarter of the base stations observed. The error is almost negligible when half of the base stations are observed.

5.6 Related Work

5.6.1 Sampling Network Load

Use of sampling to monitor network behavior has been widely used in the Internet measurement community. See, for example [50] for a review. Much of the focus has been on packet sampling. A straightfoward approach is systematic packet sampling where packets are sampled according to a deterministic function. In some cases systematic sampling can perform better than random sampling [66]. Widely deployed original Cisco NetFlow tool [5] uses systematic sampling. The potential problems of systematic sampling are avoided by using independent, randomly generated triggers in order to select packets [144, 107]. There is also a concept of probabilistic sampling where samples are chosen in accordance to a pre-defined selection probability [146]. Another line of thought is to do adaptive sampling as the sampling rate influences the accuracy of estimation procedure [38, 37]. The authors in [53] have proposed the idea to sample packets based on a hash function computed over the packet content. to reconstruct the trajectory of packets having same hash value. But just by doing packet sampling, accurate flow level information may not be estimated in practice. The authors in [70] have introduced the concept of flow sampling. The authors in [51] and [52] have proposed size-dependent flow sampling where flows of byte size greater than a threshold are sampled with probability 1 and the others are sample with the probability proportional to their size. Similar approach has been used in [54, 84]. We have used the concept of sampling at a higher level in the context of spatially separated entities like base stations. Similar ideas have been used in [46], but in the context of data acquisition of sensor networks.

5.6.2 Cellular Network Data Analysis

Only a limited number of papers have analyzed cellular network characteristics to understand the global view. The authors in [112] have characterized the settings of operational state machine that guides the radio resource allocation policy in a UMTS network. They have used actual cellular data traces for the investigation. The authors in [74] have analyzed customer tickets collected from a large cellular network to identify potential network problems. The authors in [81] have grouped users and browsing profile simultaneously using real mobile network data collected from a large 3G cellular service provider. The authors in [138] have presented a large scale measurement analysis to characterize the primary usage in cellular voice network. They have investigated the spatial correlation in the network but in a limited scope. The authors in [111] analyzed the feasibility of web caching in the context of cellular network using network-wide data. In our earlier work [105] presented in Chapter 2 we have analyzed individual subscriber behaviors, subscriber mobility, and base station traffic dynamics at length using a superset of the data set used in this work. We also have investigated various forms of spatial relationships among base station loads [106] presented in Chapter 3. However, none of these works explicitly used any form of sampling to reduce the monitoring load or data collection effort in the network.

5.7 Conclusions

Building scalable techniques for measurement-driven analysis of large cellular networks is an important problem. This chapter addresses this issue by using probabilistic modeling techniques to capture the nature of conditional dependences in the traffic load on cellular base stations. Using available machine learning tools and trace data collected from a cellular network we have shown how to construct such models with a sparse structure. While such modeling may have many uses, we specifically have used the modeling approach for spatial sampling. The idea is to reduce the measurement burden in the network, with only a subset of base stations monitored and the rest estimated using the model. With trace data from a network of 400 base stations, we have shown the steps taken to build the model and analyzed the properties of the model as an error versus complexity tradeoff. Finally, we have used two resource/traffic management related applications to show the impact of using such modeling for load estimation. This part of the analysis shows that the impact on the applications is relatively minor even when a small fraction of the network is actually monitored.

Statistical machine learning offers many modeling techniques that can be gainfully used to learn structural properties of the network data. The modeling approach we considered is ‘spatial’ and does not directly capture ‘temporal’ properties. To capture possible temporal variations during the day, we took a brute force approach by building separate models for different parts of the day. Our future work will include temporal aspects in the model directly to provide a further degree of robustness.

Chapter 6

Passive Monitoring of WiFi Networks with Applications

6.1 Introduction

In this chapter, we shift our focus from cellular network to WiFi network and present a technique to model and understand the wireless interference between network nodes and links in realistic WiFi network deployments [1]. The goal is to do this in the most unobtrusive fashion possible: (i) *Without installing any monitoring software on the network nodes*: this is motivated by practicality as many APs are often closed devices, and clients may not be always be privy to new software; (ii) *Using a completely passive technique*: This is important as active measurements impact (and are impacted by) network traffic.

To achieve these goals, our approach uses a distributed set of 'sniffers' that capture and record wireless frame traces. We then analyze the trace to understand the interference relations. While this is true that this approach requires additional hardware for measurement, this can be viewed as a form of third-party solution. Such independent third-party solutions for wireless monitoring are not uncommon in industry [1, 2]. The research community has also provided similar approaches. See, for example, DAIR [22, 23], Jigsaw [35] and Wit [94]. While these approaches provide many monitoring solutions,

[1] The early part of the work in this chapter was done in collaboration with Anand Kashyap

they still do not provide fundamental understanding of interference relations between network nodes and links.

Aside from understanding interference relationships, there are other applications of the technique we develop. Certain types of selfish behaviors can be detected via this approach — an example we will demonstrate. A selfish node can gain unfair share of the available bandwidth by manipulating different MAC protocol parameters, such as the clear channel assessment (CCA) threshold, or the backoff window size. This can deliver an unfair bandwidth advantage to a selfish node [108] and can be used to even launch a denial of service attack. A node, for example, can be selfish by raising the CCA threshold. This can effectively disable its carrier sensing and creates more transmission opportunities for the selfish node. This can also cause collisions, and thereby force the other transmitters in the vicinity to perform backoff. While the selfish node itself may also undergo a collision, the backoff period will be shorter as it will not freeze its backoff counter when carrier sensing is disabled. We can detect the selfish carrier-sense behavior using the pair-wise interference relationships discovered by the proposed technique. In our knowledge, this problem has been explored only in one paper [108], that provides a limited solution using a non-passive technique.

6.1.1 Approach

A set of 'sniffers' are deployed to collect traffic traces from a live network. The traffic traces are then merged using existing merging techniques for distributed sniffer traces [142, 94, 35].[2] Then, we use a machine learning based approach to analyze the merged traces to infer sender-side interference relationships. It also determines the receiver-side interferences. See Figure 49. More specifically, the approach determines for each link (or node), which other links (or nodes) it interferes with, as well as the extent or degree of interference.

For detecting selfish behavior, we use the sender-side interference relation to identify *asymmetric behavior* between network nodes. This means that between a given pair of nodes, while one node can sense the transmission of

[2] These techniques also infer and add the packets that are missing from the merged trace.

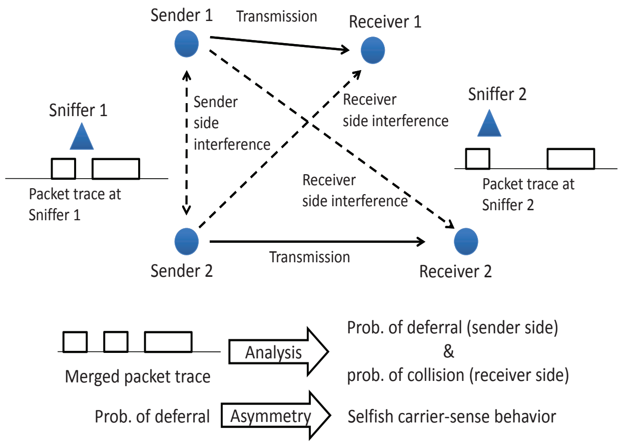


Figure 49: Overview of the approach.

the other node, the converse is not true. *The main idea of our approach is that significant asymmetry in favour of a specific node when witnessed persistently by multiple other nodes is indicative of selfish behaviour.* This is because such asymmetry may be very unusual due to normal wireless channel effects.

Our approach can be used as a 'toolbox' with two important applications: understanding the interference properties, and detecting selfish behavior in an arbitrary WiFi network, regardless of the topology or architecture. System managers can use this tool to perform capacity planning and appropriate radio resource management, such as assignment of channels, transmit power levels or directions when using directional antennas. In addition, this tool can act as a 'police' to detect the malicious user activity and can provide a significant insight about WiFi interference behavior in large installations, potentially influencing future standards design.

Because of its passive nature, our approach is dependent on the sufficiency of the available network traffic. The most important challenge is to make accurate estimation of interference for traffic of unknown and arbitrary nature, especially in presence of low load in the network. Also, accurate identification is very challenging when a selfish node exhibits probabilistic behavior to avoid detection.

We discuss related work in Section 6.2 and the broad approach in Section 6.3. The details of the HMM formulation are covered in Section 6.4. Section 6.5 contains the experimental evaluations for interference relation. Section 6.6 defines the metric to identify selfish nodes and Section 6.7 presents the experimental evaluations for selfish carrier-sensing detection. We will conclude in Section 6.8.

6.2 Related Work

6.2.1 Analyzing Interference

Interference in an 802.11 wireless network can be readily measured by putting saturated traffic on two links simultaneously and measuring the aggregate throughput. The decrease in throughput due to interference from the other transmission indicates the amount of interference. This approach ordinarily needs $O(n^4)$ measurements for an n node network. However, [101] outlines a method to do this with only $O(n^2)$ measurements. More sophisticated approaches do not perform direct measurements as above, but uses certain modeling steps to reduce the number of measurements to $O(n)$. The idea here is to (i) measure Received Signal Strength (RSS) on each link using broadcast beacons, (ii) perform a profiling study describing the deferral and packet capture behavior of the radio interface, (iii) develop a suitable MAC layer model. Together the above can estimate interference between active links and link capacities in presence of interfering traffic. There are different variations of this basic approach presented in [120, 78, 113] which need active measurement. While the requirement of a quiet, interference-free environment to do RSS measurements makes these methods unrealistic in live networks, the method presented in [120] can model interference by doing measurement even in the presence of external interference. However, the profiling needs to be done apriori.

In addition to the above, there are various sundry works on evaluating interference characteristics in an 802.11 network. For example, in [72], the authors investigate the impact of carrier sensing. In [31], the authors develop a model for the physical layer capture. In [42], the authors show that pairwise interference modeling is often not accurate and multiple interferers must be accounted for. In [93], the authors present an inference tool to infer the activity share among a set of conflicting links. In [79], we present our approach of indentifying interference relations, but with limited evaluation.

6.2.2 Detecting MAC-Layer Misbehavior in 802.11

Most of the existing MAC layer misbehavior detection techniques only attempt to detect one type of selfish behavior: backoff manipulation in 802.11. They use different methods, such as game theoretic approach [29], Sequential Probability Ratio Test (SPRT) [115], non-parametric cumulative sum (CUSUM) test [133], coordination from the receiver [87] to identify backoff manipulation or to restrict the sender from being selfish. DOMINO [117] can detect other misbehaviors in addition to backoff manipulation, e.g., sending 'scrambled frames', using smaller DIFS and using oversized NAV. None of these techniques can detect selfish carrier-sense behavior and thus can be complementary to the approach described in this paper.

Manipulation of the carrier-sense behavior is harder to detect. This is because normal fluctuations of wireless channel must be distinguished from manipulated carrier sensing. In our knowledge, only one paper [108] has addressed this issue before our work in [104]. The technique proposed in [108] relies on a strong assumption that the selfish node that has increased its CCA threshold is unlikely to correctly recognize low power transmissions from the AP as legitimate packets. Thus, by sending low power probes, the AP can potentially detect such nodes. This assumption implies that packet reception with power lower than CCA threshold is not possible, as such packets are treated as noise. However, the attacker can avoid detection by simply changing the CCA threshold only when it transmits a packet and reverting back to the normal threshold right after the transmission.[3] Also, depending on how the radio transceiver is designed, packet reception success may not be dependent on the CCA threshold. Also, this technique is not passive.

6.2.3 Use of Distributed Sniffers

Techniques based on using distributed sniffers can be found in a number of measurement studies for the purpose of learning various properties of live network such as congestion [73], protocol behavior in a hotspot setting [122, 35, 94], etc. The DAIR system also uses such an approach for troubleshooting [22] and security [23]. More details on similar related works appear in Section II-B of [79]. In this chapter, we employ a technique similar to [142] to merge individual traces into a unified trace. However, unlike all the previous studies, our focus is on learning the interference relations and detecting selfish carrier-sense behavior in the network.

6.3 Overall Approach

6.3.1 Problem Statement

In 802.11, interference can occur either at the 'sender side' or at the 'receiver side' (or both) [78]. Sender side interference pertains to deferral due to carrier sensing. In this case, one node freezes its backoff counter and waits when it senses the second node's transmission. In case of receiver side interference, overlapped packet transmission causes collisions at the receiver. This requires packet retransmission. In both cases, the sender additionally has to go through a backoff period, when the medium must be sensed idle.[4] The net effect of the interference is reduction of throughput capacity of the network.

Our general goal is to understand the deferral behavior that accounts for the sender side interference. To detect selfish carrier-sense behavior, we need

[3] There may indeed be a latency issue that can slow down the selfish node if such changes are frequent. But we do not consider this to be a fundamental issue.

[4] We are assuming that the reader has an overall idea of the 802.11 MAC protocol. Specific details will be brought up as necessary.

to identify the asymmetry in the deferral behavior. The deferral behavior between two nodes, X and Y is said to be asymmetric if Y defers for X's transmission and X does not defer for Y's, or vice versa. Such asymmetry is possible in wireless networks due to interface heterogeneity. But it is simply unlikely that a node X demonstrates similar asymmetry with many such Ys in the same direction. Our strategy is to flag such nodes as potentially selfish, with degree of selfishness indicated by extent of asymmetries exhibited and the number of such Ys (called 'witnesses').

For modeling convenience, we consider interference between node or link pairs only. Note that it will allow us to capture the 'physical interference' [64] where a given link is interfered collectively by a set of other links, not by a single link alone. This is due to the additive nature of the received power. However, pairwise consideration can still be quite powerful in practice. Also, in reality the probability of having multiple concurrent packet transmission is very small even when there are many active flows in the network. For example, the authors in [94] analyzed a major trace collected during the SIGCOMM 2004 conference and found that only 0.45% of packets actually overlapped in transmission. This limits the usefulness of having a more elaborate higher order model for deconstructing interference relationship. On the other hand, pair-wise relationship can be enough for our method of detecting selfish carrier-sense behaviour. *We do note that this simplification is not fundamental to our basic technique. The technique can be extended, albeit with higher computational cost, to physical interference.*

In wireless networks, interference is better expressed in terms of probabilities because of the inherent fluctuation of the signal power due to fading effects and probabilistic dependency of error rates with SINR (signal to interference plus noise ratio). Prior measurement and modeling studies have elaborated on this aspect [101, 78]. Thus, *in this work we estimate via passive monitoring the non-binary, pairwise interference between any two network nodes or links, in terms of probability of interference.* For any link pair, the probability of interference is given by:

$$p_d + (1 - p_d)p_c, \tag{13}$$

where p_d is the 'probability of deferral' between the senders, and p_c is the ‘probability of collision’ at the receivers if both senders transmit together.[5] See also Figure 49. When considering node pairs only, probability of interference is just p_d, assuming symmetric interference between these two nodes.

If one of the nodes in a node pair shows selfish carrier-sense behavior, the sender-side interference (p_d) should be very asymmetric. Thus, our next goal is to quantify the asymmetry for each pair of nodes in the network. For a given pair of nodes, X and Y, we estimate the probability $P_{\text{def}}(X,Y)$ that node X defers to node Y's transmission. We do this estimation for all node pairs in either direction. As mentioned before, significant asymmetry in this probability indicates possible selfishness. Let us assume that there is asymmetry in favor of X, i.e., $P_{\text{def}}(X,Y) << P_{\text{def}}(Y,X)$. If this is also witnessed by more nodes such as Z, i.e., there exists several $Z \neq Y$ such that $P_{\text{def}}(X,Z) << P_{\text{def}}(Z,X)$ we have more confidence that X is behaving in a selfish manner.

6.3.2 Discussions

To estimate the interference relations between a given pair of nodes, our technique needs to have instances when simultaneous transmissions are attempted by the two nodes. The conjecture here is that if one observes the live network traffic for a long enough period, enough of such instances will be available for each node pair. Our goal is to (i) identify such instances, and (ii) infer the deferral behaviors during such instances. There are several challenges here. First, creating a complete and accurate trace is itself a difficult problem. There are many approaches proposed in literature to create a complete trace. But for our technique, incomplete trace may suffice as long as it is statistically similar to the complete trace. Second, unknown load of the nodes makes it harder to estimate the deferral behavior. In our apprach, we utilize the strategy of analyzing inter-packet times which can provide certain confidence. Third, heuristics can be used to infer the deferral behavior. But straightforward heuristics may have limited power. More details about these challenges appear in [79].

[5] This definition ignores ACKs for modeling and notational convenience as in [101, 78], and is not a limitation. We indeed use unicast traffic with ACKs for evaluation.

6.3.3 Approach

We need to come up with a rigorous statistical modeling approach to determine deferral behavior among network nodes. Our basic approach is as follows. We model the 802.11 MAC layer operations of two sender nodes in the network (say, X, Y) via a Markov chain. The parameters of this chain (essentially the state transition probabilities) are estimated from the observed trace using an approach based on the Hidden Markov Model (HMM) [114]. These parameters in turn can estimate the deferral probabilities. We devote the entire next section describing the HMM-based approach.

6.4 Hidden Markov Model For Sender-side Interactions

A hidden Markov model (HMM) [114] represents a system as a Markov chain with unknown parameters. Here the states of the Markov chain are not directly visible, but some observation symbols influenced by the states are visible. The unknown parameters (such as the state transition probabilities of the Markov chain) can be learnt using different standard methods [114, 45, 24] with the help of the observed sequence of observation symbols. Various machine learning applications such as pattern, speech and handwriting recognition have used HMM technique. We will be using the HMM approach for modeling interactions between a pair of senders in an 802.11 network and inferring sender-side interference relations (deferral behavior) between them.

6.4.1 Markov Chain

Each sender in 802.11 MAC protocol can be modeled as a Markov chain [26, 79] as shown in Figure 50. A sender node, say X, is found in one of the following four states - ‘idle,’ ‘backoff,’ ‘defer,’ and ‘transmit.’ The essence of the 802.11 MAC protocol lies in these four states. We intentionally ignore interframe spacings (e.g., DIFS) to keep the chain simple. In the rest of the paper, we call the 4 states $\mathfrak{I}$, $\mathfrak{B}$, $\mathfrak{D}$, and $\mathfrak{T}$, respectively for the sake of brevity. The

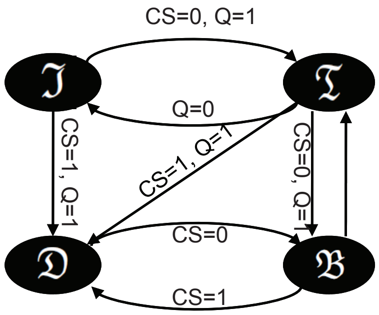


Figure 50: State transition diagram for a single sender. CS = 0 (CS=1) means that the carrier is sensed idle (busy). Q = 0 (Q =1) means that the interface packet queue is empty (non-empty).

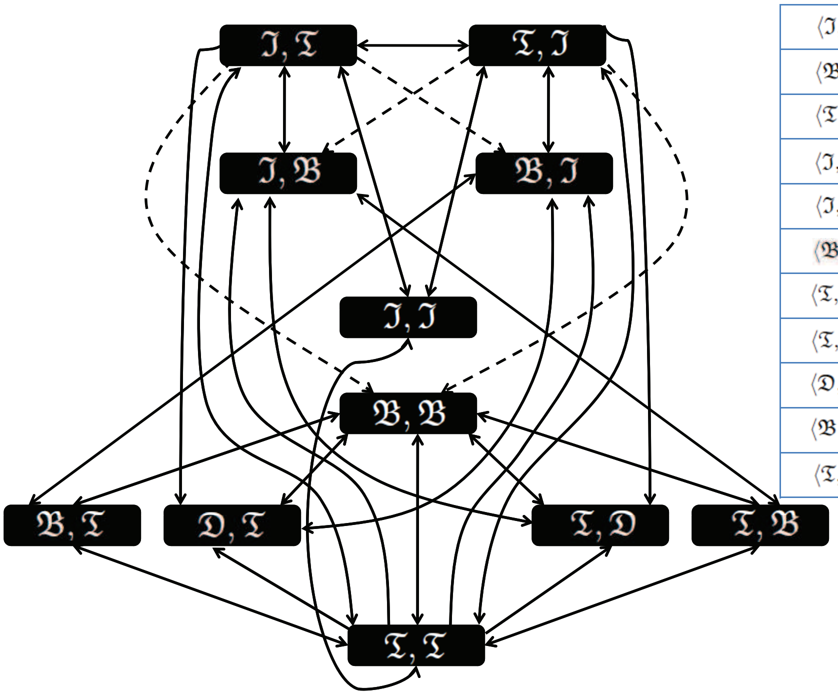
high level description of this chain can be found in [79]. Note that the state transition probability between $\mathfrak{B}$ and $\mathfrak{D}$ of the corresponding sender node is influenced by the states of other nodes (i.e., transmitting or not) in the network, and the deferral probabilities between the sender and these nodes. Similar argument applies for the transition probabilities from $\mathfrak{I}$ to $\mathfrak{D}$ and $\mathfrak{T}$, and transition probabilities from $\mathfrak{T}$ to $\mathfrak{D}$ and $\mathfrak{B}$.

Since the state transitions of the Markov chain for a given sender is impacted by the transmissions from other nodes, a Markov model of a single sender is not enough to get the complete picture of the network behavior. Instead, a combined Markov model needs to be considered. Here, each state is a tuple consisting of states of individual nodes. Such a Markov chain would be intractable as it would lead to a state space explosion with exponential number of states. Since we focus mainly on determining the pairwise interference relationships, and our technique to detect selfish behavior needs only pair-wise deferral behavior, we can restrict ourselves to the consideration of a combined Markov chain for only a pair of nodes, say X and Y. Each state in this Markov chain is a 2-tuple consisting of the states of X and Y. For example, the state where X transmits and Y defers would be $\langle \mathfrak{T}, \mathfrak{D} \rangle$. Out of 16 possible states in theory, 5 states are not legal (e.g., $\langle \mathfrak{D}, \mathfrak{D} \rangle$, $\langle \mathfrak{D}, \mathfrak{B} \rangle$ etc.[6]), leaving 11 possible states. See Figure 51 for the 2-node combined Markov chain. (Only the solid lines indicate valid transitions. The dotted transition lines will be discussed later.)

The state transition probabilities between certain states in this Markov chain are determined by the deferral probabilities between X and Y. For example, transition probabilities from state $\langle \mathfrak{B}, \mathfrak{B} \rangle$ to state $\langle \mathfrak{T}, \mathfrak{D} \rangle$ or $\langle \mathfrak{T}, \mathfrak{B} \rangle$ would depend on deferral probability of Y with respect to X. Let us explain this using an example. Assume that Y carrier senses X (or Y can sense X's transmission) perfectly. Then when X moves from $\mathfrak{B}$ to $\mathfrak{T}$ state (i.e., starts transmitting as soon as the backoff interval is over), Y must also move from $\mathfrak{B}$ to $\mathfrak{D}$ as it defers to X's transmission by freezing its backoff countdown timer. If instead Y never carrier senses X, it will remain in the $\mathfrak{B}$ state. The deferral probability of X and Y depends on the number of instances when either of the nodes moves to $\mathfrak{D}$ state.

Note again that this combined Markov chain is specified for a node pair only, as we are interested in pair-wise interference. This process can be repeated for all pairs to determine the all-pair sender-side interference. We filter out the packets of just the two senders under consideration for analysis, and ignore the other packets. This may misinterpret an active node, deferring for a third node's transmission, as idle, and we may miss an opportunity to interpret the interaction between the particular pair as interfering or non-interfering. But, it is important to note that this does not create any incorrect interpretation. Recent studies [94] show that the number of instances of 3 or more nodes simultaneously being active is much less than that of only a pair of nodes being active. Thus, we should get enough instances of just a pair of nodes being active in a long trace. An alternate but computationally expensive method could try to identify portions of the trace where only the senders in a node pair being considered are active.

[6] Note that this Markov chain assumes only two nodes X and Y interact. Thus, for example, the state $\langle \mathfrak{D}, \mathfrak{D} \rangle$ is not possible as both nodes cannot defer at the same time.



$\mathfrak{I}, \mathfrak{T}$
$\mathfrak{T}, \mathfrak{I}$
$\mathfrak{I}, \mathfrak{B}$
$\mathfrak{B}, \mathfrak{I}$
$\mathfrak{I}, \mathfrak{I}$
$\mathfrak{B}, \mathfrak{B}$
$\mathfrak{B}, \mathfrak{T}$
$\mathfrak{D}, \mathfrak{T}$
$\mathfrak{T}, \mathfrak{D}$
$\mathfrak{T}, \mathfrak{B}$
$\mathfrak{T}, \mathfrak{T}$

$\langle \mathfrak{I}, \mathfrak{I} \rangle$	i_l
$\langle \mathfrak{B}, \mathfrak{I} \rangle$	i_s
$\langle \mathfrak{T}, \mathfrak{I} \rangle$	x_l
$\langle \mathfrak{I}, \mathfrak{B} \rangle$	i_s
$\langle \mathfrak{I}, \mathfrak{T} \rangle$	y_s
$\langle \mathfrak{B}, \mathfrak{B} \rangle$	i_s
$\langle \mathfrak{T}, \mathfrak{D} \rangle$	x_l
$\langle \mathfrak{T}, \mathfrak{B} \rangle$	x_s
$\langle \mathfrak{D}, \mathfrak{T} \rangle$	y_l
$\langle \mathfrak{B}, \mathfrak{T} \rangle$	y_s
$\langle \mathfrak{T}, \mathfrak{T} \rangle$	xy

Figure 51: Markov model of the combined MAC Layer behavior of two nodes (sender side only). Note that some arrows are bidirectional.

6.4.2 Observation Symbols

The state transition probabilities of the combined Markov chain depend on the deferral behavior between the two nodes under consideration. Thus, if we can learn the unknown state transition probabilities, this will in turn provide us the deferral relations. But the states of this Markov chain are not directly visible in the packet trace. Instead a set of observation symbols are visible. There are four possible observation symbols in the trace depending on whether X or Y transmits:

i: neither X, nor Y transmitting.

x: X transmitting.

y: Y transmitting.

xy: both X and Y transmitting.

We thus need to map each of the 11 states in this Markov chain to one of the four observation symbols. This mapping obviously is not unique as more than one state can map to the same observation symbol. For example, both states $\langle \mathfrak{I}, \mathfrak{I} \rangle$ and $\langle \mathfrak{B}, \mathfrak{B} \rangle$ map to the symbol i. Similarly, both $\langle \mathfrak{B}, \mathfrak{T} \rangle$ and $\langle \mathfrak{D}, \mathfrak{T} \rangle$ map to symbol y. The difficulty here is that backoff cannot be distinguished from defer or idle periods. This ambiguity can be reduced by using a heuristic that exploits the time duration of various observation symbols. This is elaborated below.

A backoff interval in 802.11 lasts for an integral number of slots (20 μs in 802.11b) chosen randomly from a window of 0 to 31 slots (for first backoff stage[7]). This knowledge can be used to distinguish between backoff and idle/defer periods. The conjecture here is that defer and idle periods are very unlikely (though not impossible) to be within this bounded interval and also last for an integral number of slots like backoff period. But this strategy requires the clock accuracy within few microsecond, which demands specialized technique.

We thus use a weaker heuristic in this work that does not require strong clock accuracy. We assume that defer/idle periods are always longer than 31 slots and backoffs are always equal or shorter. This, however, introduces errors for very short idle time and small 802.11 frames with airtime less than 31 slots (620 μs for 802.11b[8]). These sources of error make the results presented in the next sections as only a lower bound on the accuracy obtainable by the base technique. We keep this as our future work to remove the timing inaccuracy by using more sophisticated technique.

[7] As a simplification, we develop the model only for the first backoff stage here. This implicitly assumes that retransmissions are rare (which has been true in our experiments). The general approach can be extended to handle multiple backoff stages by observing the number of retransmissions in the trace.

[8] This means TCP packets with payload less than 400 bytes in 802.11b.

Based on the above weaker heuristic, each observation symbol (except xy) can be classified into two types. The symbol i can be either i_s or i_l, corresponding to short (≤ 31 slots) and long (> 31 slots) respectively. According to the heuristic, i_s is most likely output by $\langle \mathfrak{B}, \mathfrak{B} \rangle$ state, while i_l is most likely output by $\langle \mathfrak{I}, \mathfrak{I} \rangle$ state, for example. Similarly, the symbols x and y can be either x_s and x_l, and y_s and y_l, respectively to differentiate among the activities (defer/idle or backoff) of the non-transmitting node during that period. Figure 51 shows the observation symbols for each state.

The heuristic described above helps us to distinguish between backoff and idle/defer periods. However, we still cannot differentiate between idle and defer. For this reason, both the states $\langle \mathfrak{T}, \mathfrak{I} \rangle$ and $\langle \mathfrak{T}, \mathfrak{D} \rangle$ map to the same observation symbol x_l. This implies that the transition from state $\langle \mathfrak{T}, \mathfrak{I} \rangle$ to state $\langle \mathfrak{T}, \mathfrak{D} \rangle$ will not be visible in the merged trace as there is no change in the observation symbol. Thus any transition from state $\langle \mathfrak{T}, \mathfrak{I} \rangle$ to any other state, for example, state $\langle \mathfrak{I}, \mathfrak{B} \rangle$ via state $\langle \mathfrak{T}, \mathfrak{D} \rangle$ will not be correctly interpreted. To overcome this problem, we force transition links from state $\langle \mathfrak{T}, \mathfrak{I} \rangle$ to states which have incoming transition from state $\langle \mathfrak{T}, \mathfrak{D} \rangle$. We refer to these links as virtual links. Similarly, we also add virtual links from state $\langle \mathfrak{I}, \mathfrak{T} \rangle$ symmetrically. Figure 51 shows the virtual links in the model in dotted lines. After we calculate the transition probabilities of the model using the technique described in the following subsection, we remove such virtual links and distribute the probability on each such virtual link to the corresponding sequence of valid transition links. See the appendix for a detailed elaboration of this technique.

Each packet in the merged packet trace is timestamped with the arrival time at the sniffer along with other information including the id of the sender, size of the packet, and the rate at which it was transmitted. We parse this information in the trace to obtain the sequence of observation symbols for the two senders under consideration. Based on this sequence, we use the following technique to learn the state transition probabilities of the Markov chain, that in turn will provide the probability of interference between the senders.

6.4.3 Formal Specification and Learning

We now formally describe the HMM using standard notations [114]. The HMM consists of the following:

- Set S of N states, where $N = 11$. S is given by:
  $S = \{S_i\} = \{\langle\mathfrak{I},\mathfrak{I}\rangle, \langle\mathfrak{B},\mathfrak{I}\rangle, \langle\mathfrak{T},\mathfrak{I}\rangle, \langle\mathfrak{I},\mathfrak{B}\rangle, \langle\mathfrak{I},\mathfrak{T}\rangle,$
  $\langle\mathfrak{B},\mathfrak{B}\rangle, \langle\mathfrak{T},\mathfrak{D}\rangle, \langle\mathfrak{T},\mathfrak{B}\rangle, \langle\mathfrak{D},\mathfrak{T}\rangle, \langle\mathfrak{B},\mathfrak{T}\rangle, \langle\mathfrak{T},\mathfrak{T}\rangle\}$.

- Set V of M observation symbols, where $M = 7$. V is given by: $V = \{i_s, i_l, x_s, x_l, y_s, y_l, xy\}$.

- Matrix A of state transition probabilities, indicated by $A = [a_{ij}]$, where a_{ij} is the transition probability from state S_i to S_j. This matrix is unknown at the outset and will be determined. Note that some state transitions are invalid and such a_{ij} is set to 0. Such transitions are not shown in Figure 51.

- Matrix B of observation symbol probabilities, indicated by $B = [b_{jk}]$, where b_{jk} is the probability that the observation symbol is v_k for state S_j. In our case, observation symbols are deterministic for each state. However, they are not unique. The mapping from states to symbols are shown in a table within Figure 51.

- Vector π of the initial state distribution, indicated by $\pi = [\pi_i]$, where π_i is the probability of initial state being S_i. We use $\pi_i = 1/N$ for all $i, 1 \leq i \leq N$.

The above specification defines the HMM, $\lambda = (A, B, \pi)$. The packet trace provides the observation sequence $O = O_1, O_2, \cdots O_T$, where each observation $O_t \in V$, and T is the number of observations in the sequence.

Given the above HMM λ and the observation sequence O, our goal is to learn the model parameters $\lambda = (A, B, \pi)$ that maximize $P(O|\lambda)$. This is a difficult problem, and there is no optimal algorithm for it. We can, however, use the expectation-modification (EM) algorithm, which is an iterative method to determine λ, such that $P(O|\lambda)$ is locally maximized. The EM algorithm alternates between an expectation (E) step, which computes the model parameters most likely to produce the observation, and a modification (M) step, which computes the maximum likelihood of model parameters across multiple E steps [45]. We use the well-known *Baum-Welch method*, which is a type of EM algorithm, based on the forward-backward algorithm developed by Baum *et. al.* [24]. The method ensures that in every estimation step, we find a model which is more likely to produce the observation. Thus, if we estimate the parameters of the model λ to get $\overline{\lambda}$, then $P(O|\lambda) \geq P(O|\overline{\lambda})$.

While initializing the state transition probabilities in Baum-Welch method, we asssign equal probability to all the outgoing valid transitions from each state. This ensures that there is no initial bias in the model towards interfering or non-interfering pair of nodes. This also aids in quick convergence of the method. We deal with the problems of numeric inaccuracies because of continued multiplications of certain small fractions by using the scaling technique in the procedure [92].

Let $\Pi = [\Pi_i]$ be the stationary (steady state) distribution of the states. After learning the transition probabilities $A = [a_{ij}]$, $\Pi = [\Pi_i]$ can be determined as $\Pi = \lim_{n \to \infty} \pi A^n$. The convergence is guaranteed as A is a stochastic matrix.

6.4.4 Interference Relations

6.4.4.1 Learning Sender Side Interference

Transitions into any state with a defer component (i.e., states such as $\langle \mathfrak{D}, * \rangle$ and $\langle *, \mathfrak{D} \rangle$) indicate interference. Similarly, transitions into any state of the set $\{\langle \mathfrak{B}, \mathfrak{T} \rangle, \langle \mathfrak{T}, \mathfrak{B} \rangle, \langle \mathfrak{T}, \mathfrak{T} \rangle\}$ indicate absence of interference. Thus the sender side interference can be interpreted as the total probability of transition into the interfering states. If we represent Π_i's as $P(\mathfrak{I}, \mathfrak{I})$, $P(\mathfrak{B}, \mathfrak{I})$ etc, the deferral probability, p_d, is given by,

$$\frac{P(\mathfrak{D}, \mathfrak{T}) + P(\mathfrak{T}, \mathfrak{D})}{P(\mathfrak{D}, \mathfrak{T}) + P(\mathfrak{T}, \mathfrak{D}) + P(\mathfrak{B}, \mathfrak{T}) + P(\mathfrak{T}, \mathfrak{B}) + P(\mathfrak{T}, \mathfrak{T})}. \tag{14}$$

The above expression essentially captures the probability of being in the interfering states when one of the two nodes is transmitting. Here, we are assuming a symmetric link between a node pair. In reality, links may be asymmetric, and the above expression can be easily modified to consider asymmetric deferral probabilities. This is discussed in Section 6.6.1.

6.4.4.2 Learning Receiver Side Interference

The receiver-side interference causes collisions that can be detected relatively easily by tracking retransmissions in the trace.[9] One can identify retransmitted packets by observing the set 'retransmit bit' in the frame header. A retransmitted frame, say R, can be correlated back to the original frame, say P, that has not been received correctly as both these frames carry the same sequence number. Any frame S from a different sender overlapping with P is a potential cause of collision. If P does not overlap with any other frame, the packet loss is due to wireless channel errors rather than collisions [118, 94]. Because of the probabilistic nature of packet capture, sufficient statistics need to be built up to determine receiver-side interference. This is because frames like S and P – even when overlapping – may not always result in a collision. Thus, the receiver-side interference between two links, or in other words, the probability of collision p_c can be determined as the ratio of the collision count and the overlapped-frame count.

6.5 Evaluating Interference Relations

We will now evaluate the effectiveness of our approach to infer interference relations by a series of evaluations. We will use a mix of different scenarios starting from careful micro-benchmarking to using large and congested wireless network traces. For the benefit of the reader, we summarize the various scenarios we will use in Table 1.

[9] For unicast transmissions only. However, unicasts are much more frequent relative to broadcasts in a real network packet trace.

Sec. #	Scenario	Trace type	Trace source	Nature of traffic	Interference evaluation
6.5.1	Micro-benchmarking	Experimental	Collected by authors	UDP broadcast (diff. rates)	Sender-side
6.5.2	NS2 simulations	Simulation	Collected by authors	Short TCP transfers	Sender-side
6.5.3	Departmental WLAN	Experimental	Collected by authors	Long HTTP downloads	Sender- & receiver-side
6.5.4	SIGCOMM 2004 trace	Experimental	Externally obtained	Real traffic mix	Sender- & receiver-side

Table 1: Summary of evaluation scenarios used in the paper to infer interference relations.

6.5.1 Micro-benchmark for Sender-side Interference

We first describe a set of micro-benchmarking experiments. Here two senders transmitting broadcast traffic are used to specifically evaluate the sender-side interference using carefully controlled load. We evaluate for a range of interference scenario by positioning the senders at different locations. We also compare our micro-benchmarking experiments to infer sender-side interference with two other possible methods described below.

6.5.1.1 Comparison Points

A) Profile based method (PROFILE): This technique is specifically based on [120, 78] and needs active measurements. It creates a profile for each device in the network with specific interface card used. Profiling is done by collecting a large number of measurements using a pair of devices to create the correlation between the received signal strength (RSS) and the probability of deferral. This needs to be repeated for all different cards used in a network. Later the profile can be used to estimate the probability of deferral between two nodes by measuring the average RSS values between them and doing a lookup on the profile. As this technique is expected to be quite accurate, we use this as a benchmark.

B) Moving window based method (WINDOW(t)): This is a simple heuristic that may need extensive parameter tuning. In this technique, a moving time window of size t seconds over the combined packet trace is maintained. For each window position, we analyze only the packets inside the window and infer whether the nodes considered interfere or not (see below). Finally, we count the number of window instances where the nodes interfere, and obtain the probability of deferral as a fraction.

Specifically, we use the following approach.

- Only consider windows that have packets from both nodes. (We do not want to consider windows that have mostly one node transmitting and the other silent.)

- Determine the saturation throughput T_{sat}. This is tricky and will depend on the transport protocol and packet sizes used.

- The aggregated throughput T_{obs} of the two nodes in the window being considered is calculated. If $T_{obs} > T_{sat} - \delta_1$, then the window is considered saturated, otherwise the window is considered unsaturated.

- A saturated time window is marked non-interfering if $T_{obs} > T_{sat} + \delta_2$.

- The parameters δ_1 and δ_2 are needed to ride out measurement noises and are tuned.

- Probability of deferral is the fraction of saturated time windows that are marked interfering.

6.5.1.2 Micro-benchmarking with Two Nodes

Our micro-benchmark experiment consists of a setup with two senders and two sniffers.

Each sniffer is co-located with a sender to guarantee that all frames are captured. Both the senders and sniffers have 802.11 radios. All the cards used have Atheros chipsets, and the popular MadWiFi driver is used. We also use a 'beacon' node, whose sole responsibility is to transmit 802.11 beacons at regular intervals to provide a common time base needed for merging the

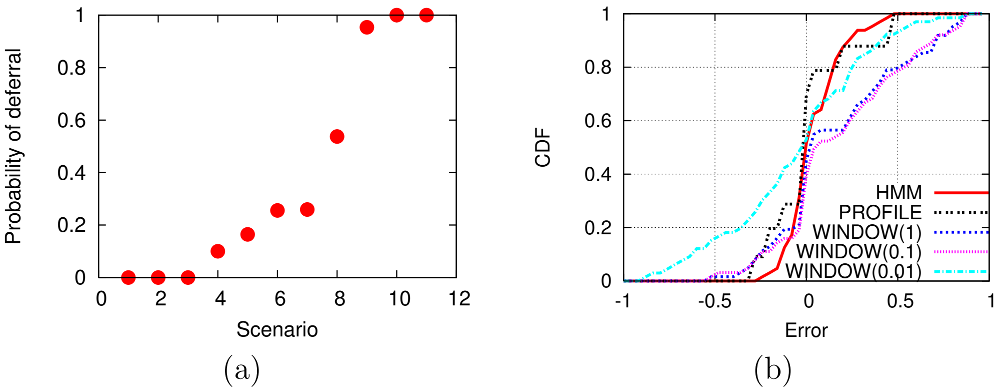


Figure 52: Combined performance results for 11 chosen scenarios for two node experiments.(a) Measured probability of deferral for different scenarios; (b) CDF of error in estimating probability of deferral.

traces. In a normal deployment, these beacons will be supplied by existing APs.

For the experiments, we configure all the four radios in the same channel. The choice of channel is immaterial. We also set the sender radios in 'ad hoc' mode and the sniffer nodes in 'monitor' mode. All experiments are done for 802.11b using the PHY-layer data rate of 11Mbps. A large packet size (1470 bytes) is chosen for the experiments. This is because, with smaller packets, the sniffers cannot capture all packets in our low-cost embedded hardware, likely due to inefficiencies in interrupt processing. Tcpdump is used for packet capture in the sniffers.

We create a range of interference scenarios by positioning one sender-sniffer pair fixed at one location, and moving the other to various locations in the building. For each scenario, we perform the following measurements. First, we measure the actual probability of deferral between the nodes. To do that, we follow the method in [101] briefly described below. We let each sender, configured with saturated UDP traffic, broadcast in isolation for a minute, and measure their throughputs in isolation. We then let them broadcast together with saturated traffic, and measure their throughputs again. The ratio of the sum of throughputs when the senders broadcast together to the sum of throughputs when the senders broadcast in isolation is defined as BIR, or

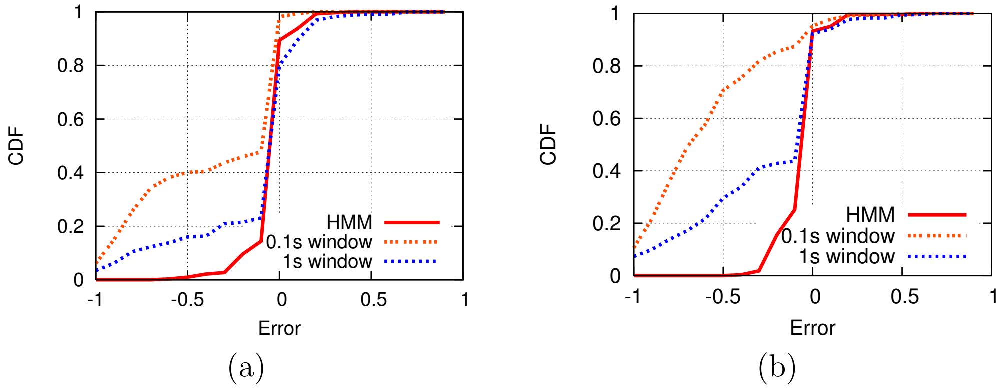


Figure 53: NS2 simulation results showing CDF of error in deferral probability estimates for the (a) sparse and (b) dense networks.

the broadcast interference ratio [101]. Note $0.5 \leq BIR \leq 1$. The 'measured' probability of deferral is estimated as $1/BIR - 1$.

We also collected the RSS measurements at each sender for each scenario when the other sender broadcasts in isolation. This is used to estimate the probability of deferral using the $PROFILE$ method described above. The profiling of each interface card have been independently done using a method similar to [78].

Next, we conduct a series of experiments with varying traffic load in the following fashion for each scenario to evaluate HMM and $WINDOW(t)$ methods. The senders are configured to broadcast UDP packets simultaneously for one minute with 10 different load levels ranging from 0.1 Mbps to 6 Mbps. The PHY-layer bit rate is chosen to be 11 Mbps; thus, 6 Mbps for each node means saturated load. Meanwhile, each sniffer captures all the packets it hears in that duration. The packet trace from each sniffer is merged using the techniques described earlier, and this combined trace is used to estimate the probability of deferral using the HMM and the $WINDOW(t)$ methods. The later is repeated for three different window sizes (t= 0.01s, 0.1s, 1s).

We make such measurements for 11 different locations of the senders, creating 11 different scenarios. The distribution of the measured probability of deferral at different locations is presented in Figure 52(a). For each scenario, 10 different values of offered load are used between 0.1 Mbps and 6 Mbps, thus creating 110 measurements for HMM and the $WINDOW(t)$ methods, and 11 measurements (one for each scenario only) for the $PROFILE$ method. The distribution (CDF) of errors ('estimated' – 'measured' probability of deferral) is plotted for all three methods in Figure 52(b). Note that the HMM approach is quite competitive with the $PROFILE$ method. In fact, it is slightly better overall for the particular distribution of deferral probabilities. The reason for this is that the $PROFILE$ method uses profiles for interface card models, rather than from the specific cards used in the experiments [78], even though it uses RSS measurements on the actual network with the actual cards used. Variations between individual cards can lead to modeling errors.

The root mean square error (RMSE) values are 0.165 and 0.208 for HMM and $PROFILE$, respectively. The RMSE values for $WINDOW(t)$ methods is 0.385, 0.408, and 0.402 for $t = 0.01$s, 0.1s, and 1s respectively. We have noted before, however, that the $PROFILE$ method is impractical for analyzing live network traffic and it also requires access to the network nodes.

Overall, HMM is quite competitive with $PROFILE$, but requires only passive measurements. The experience with the window-based method is quite variable. It is also quite sensitive to choice of window size.

6.5.2 Simulation-based Evaluation

Simulations let us create arbitrary topologies and interference conditions easily. However, the physical layer (including interface behavior for carrier sense and packet capture) implementation is often idealized or unrealistic in simulations. To address this issue, we use an extended version of the ns2 simulator that includes realistic measurement-based models [77]. These models were validated against experimental results showing excellent accuracy [77].

For the sake of completeness, we note that the enhancements in ns2 in [77] are done specifically in the following physical layer components – (i) radio propagation model, (ii) deferral or carrier sense model, and (iii) packet reception model. For (i), models are derived from real measurements in a testbed. For (ii) and (iii), measurement-based profiles of a testbed are created where every value of RSS is mapped to a deferral probability and every value of

SNR is mapped to receive probability, respectively. These profiles make the interference relations between links non-binary.

We consider two scenarios, where 20 nodes are uniformly and randomly distributed in a 200m×200m area and a 100m×100m area. These two scenarios produce different topologies: sparse and dense. We generate traffic by creating one-hop TCP flows on randomly chosen feasible links. Both the inter-arrival time and duration of flows are chosen from an exponential distribution. For the results presented here (See Figure 53), the simulations were run for 180s, the average duration of each flow was 5s, and the average inter-arrival time between flows was varied from 2.5s to 1s, such that the average load in the network varies from 2 to 5 flows.

In Figure 53 we show the CDF of the estimation error (as before) for the probability of deferral between node pairs. CDFs for both HMM and $WINDOW(t)$ methods are presented for the sparse and dense network. The $PROFILE$ approach is not shown here as it would be perfectly accurate in the simulator (as the simulator's deferral model itself uses the same profile model). From the plots note that HMM performs significantly better than the window-based method. Average RMSE value for the HMM method is about 0.1, while the average RMSE value for the better of the two window-based methods is about 0.4. Note again the accuracy of the window-based method is quite sensitive to window sizes.

6.5.3 Complete Evaluation on WLAN

Here, we provide a complete evaluation – both sender and receiver sides. These experiments are done on an active WLAN with 7 APs spread over two floors of the Computer Science department building of Stony Brook University. 7 laptops are used as clients. Each client fetches a large file via HTTP download using a unicast link for about 20 mins. This simulates real network traffic that are sniffed using 9 sniffers (Soekris single board computers with 802.11 miniPCI cards with Atheros chipset and with external USB flash memory to store packet traces). The sniffers are deployed based on convenience, i.e., near a power outlet and in the rooms that we have regular access to. However, an

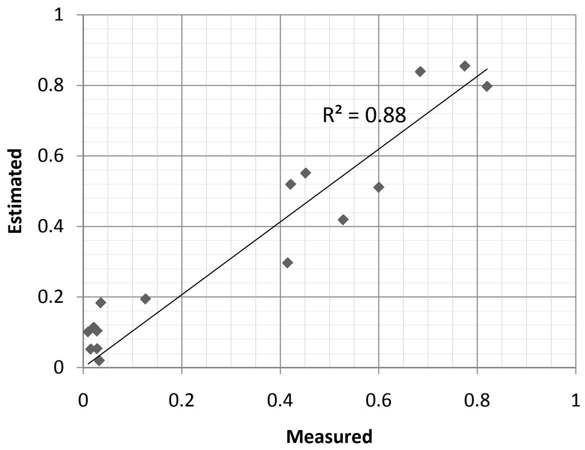
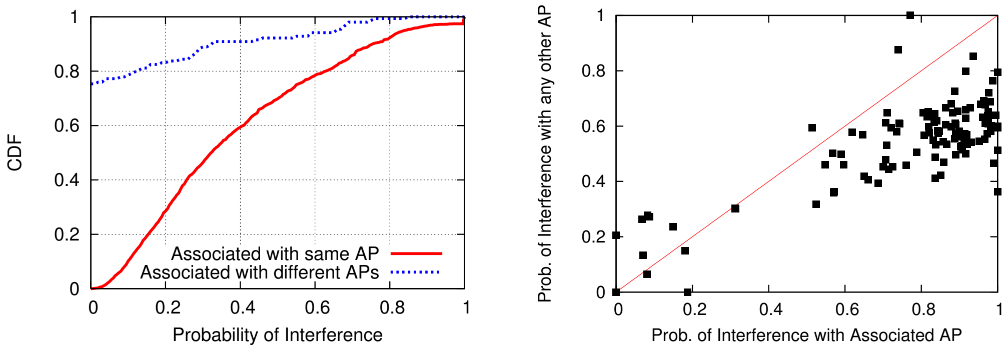


Figure 54: Estimated and measured probabilities of deferral for the 16 test cases with the departmental WLAN.

attempt was made to keep them as close to the APs as possible.

16 client laptop pairs are considered for evaluation. All of these pairs associate with two different APs. Unlike the micro-benchmarking experiments, the default auto-rate control with 802.11b is used. Also, the 802.11 frames are now unicast with ACK. RTS/CTS is disabled. For each pair, the probability of interference between the pair of download links (AP to client) is 'estimated' using equation 13. First, the probability of deferral (p_d) is estimated using the HMM-based method using the merged sniffed traffic traces from all sniffers. Second, the probability of collisions (p_c) is estimated by observing the retransmissions for overlapped packets as described in Section 6.4.4.2. However, in all cases, retransmissions were rare, typically less than 1% of frames were retransmitted. This is consistent with prior experimental observations [94]. Thus, p_c could be safely ignored with p_d alone determining the probability of interference.

For validation, p_d is 'measured' via the BIR method described in the previous subsection. For these measurements, simultaneous saturated UDP traffics on the downlinks are used for about 2 mins. The validation results are shown in Figure 54 as a scatterplot. Note the high degree of predictability of the estimation in this real-life experiment. The straight line is the least square



(a) (b)

Figure 55: Interference analysis of the SIGCOMM 2004 trace: (a) CDF of probability of interference between clients associated with the same AP and different APs; (b) comparison of interference between the associated AP and another AP.

fit with the condition that the line passes through 0. Note that it is very close to the $y = x$ line. The R^2 value for this line is 0.88 showing a good fit.

A careful reader will notice a slight bias at the low end of the deferral probabilities. The HMM method consistently overestimates deferral probability, when the probability is very small. We have also observed this in our micro-benchmarking though it does not show up in the CDF plots. The reason for this is the heuristic we used in our modeling (Section 6.4.2) that defer/idle periods are always assumed longer than 31 slots. When there is little interference, often idle periods could be shorter than backoffs. If they are misclassified as backoffs, the possibility of misclassifying some idle states as defer increases. As discussed in Section 6.4.2, a stronger heuristic using more accurate clocks could address this issue.

6.5.4 Using Large-Scale Wireless Traces

Encouraged by the strong validation results in the departmental WLAN trace analysis, we use the wireless network trace collected at the SIGCOMM 2004 conference [94] for demonstrating powerful capabilities of our tool. The trace was obtained from the CRAWDAD archive [123]. The SIGCOMM 2004 conference was 4 days long and was attended by more than 500 attendees. During busy periods, several simultaneously active flows were not uncommon [94]. The WLAN under consideration in this trace had 5 APs – three on channel 1, one on channel 8 and the other one on channel 11. Five sniffers were used each with three wireless interfaces. Two of them listened on channel 1 and 11, respectively, and the third one listened either on channel 8 or 6 [94]. We consider only channel 1 in this work.

First, we analyze the probability of interference between client-to-AP links where the clients are associated with the same AP. For this analysis, we pick random pairs of clients associated with the same AP and find a 20 min long period when they are both simultaneously active. In Figure 55(a) we plot the CDF of the probability of interference for 1990 such randomly chosen link pairs. This shows that the interference is well-distributed over the entire range showing roughly similar probabilities of (mostly) interfering clients and (mostly) non-interfering clients. This indicates that a significant number of 'hidden' clients associate with the same AP. However, these hidden clients almost never collide. Collision probability is found to be minuscule (less than 0.4%). Thus, probability of interference is again controlled by the deferral probability alone.

Next, we do a similar analysis but for pairs of clients that associate with different APs. This study is exhaustive instead of a random sampling as the number of such pairs is relatively small (154). In Figure 55(a) note that almost 75% of such client pairs do not interfere at all and about 5% interfere strongly. The rest are in between. This indicates that the association control works quite well. This point is further elaborated in Figure 55(b) where we show a comparison of the deferral probability of 120 randomly selected clients with its associated AP and with another random AP. In a good deployment we would normally expect the latter to be small and the former to be much higher than the latter. However, we see that while the interference with the associated AP is higher about 90% of the cases (indicating a good association control), the other AP often presents significant interference. This can indicate, for example, a poor channel assignment.

6.6 Detecting Selfish Behavior

In this section, we demonstrate how the interference relationship can be used to detect selfish carrier-sense behavior and define a metric to quantize the selfishness of a node. We also define the characteristic of an effective witness and introduce two simple heuristics to identify effective witnesses.

6.6.1 Detecting Asymmetric Behavior

To detect selfish carrier-sense behavior, we need to identify asymmetric behavior. This can be detected using the following fashion. The probability that X has a packet to transmit and it defers while Y transmits is given by

$$P_{\text{def}}(X,Y) = \frac{P(\mathfrak{D},\mathfrak{T})}{P(\mathfrak{D},\mathfrak{T}) + P(\mathfrak{B},\mathfrak{T}) + P(\mathfrak{T},\mathfrak{T})}. \tag{15}$$

The opposite probability (i.e., Y has a packet to transmit and it defers while X transmits) is likewise

$$P_{\text{def}}(Y,X) = \frac{P(\mathfrak{T},\mathfrak{D})}{P(\mathfrak{T},\mathfrak{D}) + P(\mathfrak{T},\mathfrak{B}) + P(\mathfrak{T},\mathfrak{T})}. \tag{16}$$

The difference between $P_{\text{def}}(X,Y)$ and $P_{\text{def}}(Y,X)$ characterizes asymmetry. Larger the difference, higher is the asymmetry. Due to the nature of our approach, the asymmetry is tested between a node pair at a time. A positive (negative) difference indicates that Y (X) gets a bandwidth advantage due to asymmetric carrier sensing. In our evaluation, we have used the difference with a simple normalization as the 'metric of asymmetry,' $\eta(X,Y)$, except when the two probabilities are both close to zero. Thus, when both $P_{\text{def}}(X,Y)$ and $P_{\text{def}}(Y,X) < \epsilon$ (ϵ was chosen to 0.01 in the evaluations), the metric of asymmetry, $\eta(X,Y)$, is given by,

$$P_{\text{def}}(Y,X) - P_{\text{def}}(X,Y), \tag{17}$$

else it is given by,

$$\frac{P_{\text{def}}(Y,X) - P_{\text{def}}(X,Y)}{\max(P_{\text{def}}(Y,X), P_{\text{def}}(X,Y))}. \tag{18}$$

Note that $\eta(X,Y) = -\eta(Y,X)$.

6.6.2 Selecting Witnesses

In general, each network node X must be evaluated for selfish behavior. By default, every other node Y acts as a witness and the above metric of asymmetry is evaluated for the pair (X, Y). Thus, for each network node X, we take the *average of the metric of asymmetry* $\eta(X, Y)$ *over all the witnesses* Y *that provide a positive value.* The negative values are discounted as they will be accounted when Y is evaluated with X as the witness. We call this average the 'selfishness metric'. We will evaluate this metric later in our simulations.

However, if X and Y are not within carrier sense range of each other (i.e., they never hear each other), Y cannot serve as an effective witness. This is because $P(\mathfrak{D}, \mathfrak{T})$ or $P(\mathfrak{T}, \mathfrak{D})$ would evaluate to zero. (In practice, due to measurement noise, they evaluate to a very small value close to zero.) Thus, the metric of asymmetry is zero. While this is correct, this does present a problem. Assume that X is indeed selfish in a 4 node network and witness Y_1 detects a very large (i.e., $\eta(X, Y_1)$ is close to 1) metric of asymmetry. However, witnesses Y_2 and Y_3 do not hear X at all (and vice versa). They offer the metric ($\eta(X, Y_2)$ and $\eta(X, Y_3)$) as close to 0. Here witness Y_1 is an effective witness while witness Y_2 and Y_3 are ineffective witnesses. Without any further information, if we aggregate these measures using an average, we obtain a low confidence in X's selfishness (about 0.3 in this example), even when we have one perfect witness and the other witnesses are clearly ineffective. On the other hand, relying on a single witness (e.g., Y_1) that points to a severe asymmetry may not be right as this may simply be due to random wireless channel/interface effects and not due to a systematic selfish behavior. Thus, this can raise false alarms.

This problem cannot be addressed without some additional knowledge of the network topology regarding which node can serve as an effective witness. Ideally, we should only rely on witnesses that are within the carrier sensing range from a potential selfish node. The more such nodes, the better.

To address this issue, we use two simple heuristics named as H_1 and H_2. For heuristic H_1, we assume that the sniffer locations are known, as well as some bounds on the carrier sense range (R_C) and transmit range (R_T) for the network nodes. Then the sniffers that are separated by at least $R_C + 2R_T$

distance, must sniff nodes that cannot hear each other. Thus in other words, for a node X sniffed by a sniffer S_X[10] and another node Y sniffed by a sniffer S_Y, node Y will not be an effective witness of node X if S_X and S_Y are separated by at least $R_C + 2R_T$ distance. This simple heuristic eliminates many nodes that should not serve as witness to each other. Note that this may not remove all ineffective witnesses, and if the bounds are incorrect, this technique may even remove some effective witnesses. But this technique is practical and easy to use, and at minimum eliminates a large number of far-away witnesses that cannot be effective by being outside the carrier-sense range.

For heuristic H_2, we do not even need to assume anything. In H_2, Y is an effective witness of X, if they are both sniffed by a common sniffer. H_2 will surely remove all the ineffective witnesses, and may also remove some effective witnesses.

For any given heuristic, for each network node X we take the average of the metric of asymmetry $\eta(X, Y)$ over all the nodes Y that are selected as effective witnesses by the heuristic and that provide a positive value for η. Then we calculate the 'selfishness metric' by a simple averaging.

6.7 Evaluating Selfish Carrier-Sense Detection

In this section we evaluate our technique to detect selfish carrier-sense behavior. We have performed two sets of evaluations: (i) a set of microbenchmarking experiments to understand the effectiveness of the approach and (ii) a set of ns2 simulations to study larger networks and complex selfish behaviors.

6.7.1 Experiments

The experiments essentially achieve careful micro-benchmarking using similar setup described in Section 6.5.1.2. Only two network links are used but wireless channel quality, traffic load and selfish behaviors are varied over a wide range.

[10] We say a node is sniffed by a sniffer, when the packets transmitted from the node can be heard by the sniffer

One transmitter is configured as 'selfish'; the other transmitter is regular and acts as the sole 'witness.' A sniffer node, located in close proxomity of each transmitter, monitors the traffic on corresponding link. In this experiment we use 802.11a and channel 52 with 6 Mbps PHY layer rate and a large packet size (1470 bytes). We use Soekris boards as the transmitters and laptops running linux as sniffers.

A node achieves selfishness by not sensing carrier before transmitting. To make a node selfish, we have used the antenna switching technique described in [32]. There are two antenna connectors on 802.11 interface for diversity where either of them can be selected for receiving/transmitting using driver-level command. We have connected one antenna to one connector, kept the other connector unconnected. Selecting the unconnected antenna as the receiving antenna effectively disables carrier sense.[11] The impact of the selfish behavior can be varied by simply varying the distance between the selfish and witness nodes. A close distance means the witness node is impacted significantly due the selfish behavior as the RSS at the witness node is high. A large distance means that RSS is low and often the witness node cannot hear the selfish node due to channel fading, and thus the selfishness causes little impact.

The benchmarking experiments are performed by increasing the distance between the two transmitters (selfish and witness) from a very small value at steps of 3 ft in 28 discrete steps. For each position, (i) the average SNR from the selfish to the witness transmitter is measured, and (ii) UDP packets are transmitted at different offered loads on their respective links for 60 sec. We use offered loads of 6 and 4 Mbps, denoting high and low loads, respectively. We experiment with both loads on the selfish node, while the witness node has only high load.

Figure 56 plots the estimated metric of asymmetry η for the <selfish, witness> node pair for each of the experiments. The plots are color-coded based on the load. The asymmetry is clearly higher with higher SNR. Note

[11] Note that selfishness can also be achieved by resetting the CCA threshold as in [108]. However, in our hardware we have found that the antenna switching technique is more foolproof than using an increased CCA threshold.

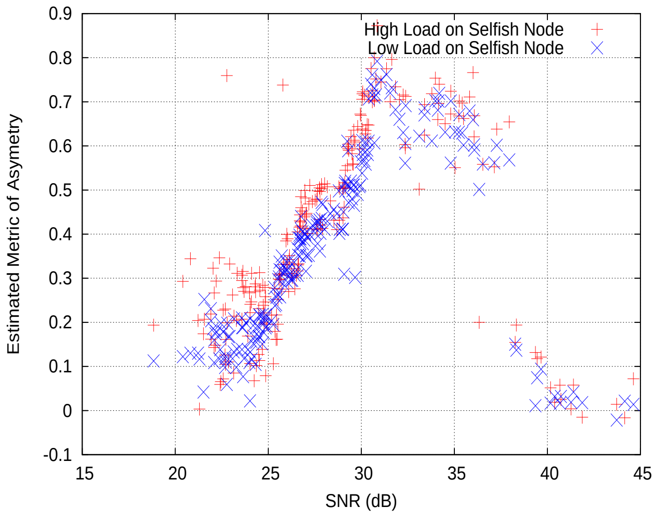


Figure 56: Experimental results with varying load on the selfish node.

that with lower load on the selfish node the asymmetry tends to be somewhat lower as expected. Also, note significantly lower asymmetry when the SNR is very high (i.e., nodes are very close). This is an artifact of our experimental technique. The selfish node starts picking up some signal at close ranges even when the antenna is disconnected, and thus it stops being selfish. So, much lower asymmetry is detected for very high SNRs.

Note that the above two node micro-benchmarking is sufficient to derive an insight into what would happen in a multiple node network. Essentially, nodes still need to be evaluated in a pair wise fashion. For each potential selfish node, we need to evaluate the metric of asymmetry with each possible witness node independently. Note again (as discussed in Section 6.3), we are currently considering pairwise interference only. But several other issues remain to be evaluated – (i) how to effectively combine the metric of asymmetry for a selfish node as provided by multiple witness nodes into a single measure, defined as 'selfishness metric' in Section 6.6.2, (ii) how suitable are the witness nodes. We will explore these issues via a packet level simulation using the ns2 simulator.

6.7.2 Simulations

Ns2 simulations let us implement various degrees of selfishness, where the selfish node senses carrier with only a certain probability. We use the term *degree of selfishness* (P_s) to indicate that the selfish node senses carrier with probability equal to $1 - P_s$. Ns2 simulations also make it easier to investigate larger networks, where there are many nodes, possibly with more than one selfish node with varying traffic and degrees of selfishness.

In our simulated scenario, there are 40 network nodes distributed randomly in a square region. We chose a deployment typical of dense WiFi client distribution in indoor office environments, assuming that there is one node in 300 sq. feet on average. The default ns2 wireless channel model is extended to include *shadowing* [116] effects. This introduces randomness in the transmission range of a node instead of making it a perfect disk. Shadowing parameters are taken from [77] where a set of measurements was done to model such parameters in an indoor environment. A set of feasible network links are chosen randomly and 1-hop UDP flows are generated with randomly chosen loads (between 0.5-1 Mbps). Each flow is active (and then inactive) only for a random interval of time. Both intervals are chosen from an exponential distribution with a mean of 5 sec. Note that the exact traffic parameters are not important for our work. All that is important is that *enough traffic is recorded so that for each pair of nodes that are potentially within the carrier sense range there are concurrent packet transmission attempts.* This ensures that any possible selfish node will find enough witnesses.

We deploy a set of 10 sniffers at random locations. Among the 40 network nodes, 1, 2 or 3 nodes are selfish. The degree of selfishness is varied. For each pair of nodes, we evaluate the metric of asymmetry by using the procedure in Section 6.4. For each network node X, we measure the selfishness metric in three ways as discussed in Section 6.6.2: (i) using all possible witness nodes (also called "no heuristic" case), (ii) using witness nodes based on heuristic H_1, and (iii) based on heuristic H_2.

Figure 57 plots the selfishness metric of each node in the scenario with one selfish node with varying degree of selfishness where the witness nodes are selected using heuristic H_2. Note that the metric has a very visible peak only

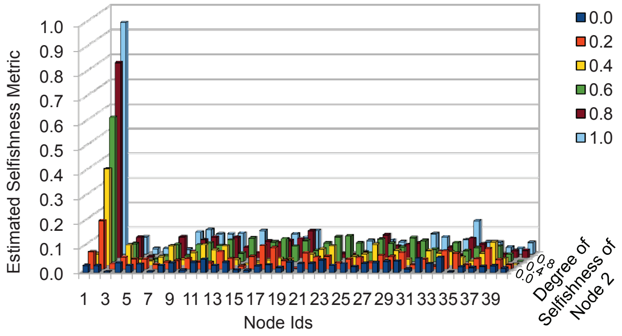


Figure 57: Simulation results for a 40 node network. Node 2 is the only selfish node. The estimated selfishness metric using heuristic H_2 is shown for each node for each of the 6 sets of simulations that are run with different degree of selfishness of node 2.

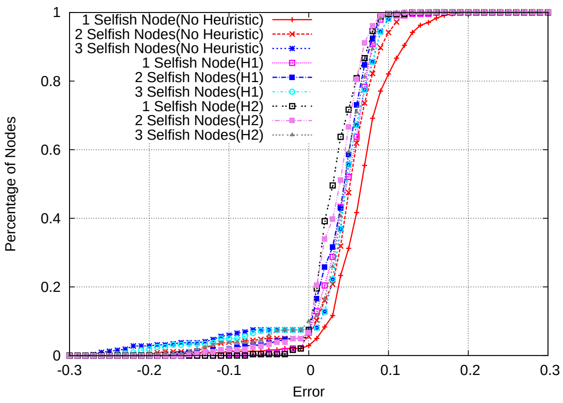


Figure 58: CDF of 'estimation error' for the selfishness metric. Three different scenarios are presented where number of selfish nodes are varied (1, 2 or 3) and witness nodes are identified in three different ways.

for the selfish node. The values of metric for the selfish nodes are roughly similar to the degree of selfishness.

Because of space limitation we do not present the similar plots for the scenarios with 2 and 3 selfish nodes using different heuristics. We instead show the overall statistics that summarizes how good our detection is. For each scenario and for each type of witness node identification technique, we evaluate for each node the 'estimation error' as the algebraic difference between

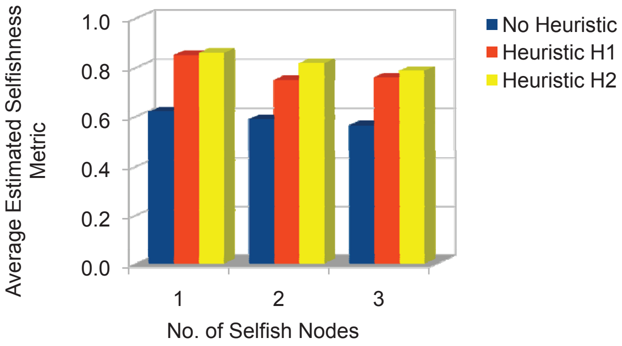


Figure 59: Simulation results for the sparse network.

the *computed selfishness metric* and the *actual degree of selfishness* of that node. All nodes (selfish and regular) are included. The estimation error is plotted as a CDF in Figure 58. Nine plots are shown for three techniques used to identify the witness nodes and for three different numbers of selfish nodes. The CDF shows that the estimation error is very small in general and heuristic H_2 performs somewhat better than the other two techniques in general.

In this scenario, the heuristics do not perform much better than the no heuristic case, because the no heuristic case itself performs very well. The reason for this is the high density of the network. To demonstrate the power of the heuristics we consider a sparser network with 40 nodes distributed randomly in squared region with one node in 1500 sq. feet on average. Different scenarios are created by varying the number of selfish nodes (1, 2 or 3) with degree of selfishness = 1. Because of the sparsity of the network we now have to deploy more sniffers to capture all network traffic. So, this time we deploy 40 sniffers randomly as before. Figure 59 shows the average estimated selfishness metric measured in three ways as before only for the selfish node(s). Note that as expected (i) estimation becomes better when we identify witness nodes using the heuristics in comparison to using all the nodes as witnesses; (ii) H_2 is generally a better heuristic, and (iii) estimation becomes worse with a larger number of selfish nodes. The reason for H_2 performing better is that it only considers effective witnesses, while H_1 may include ineffective witnesses as well. The reason for the third observation is that selfish nodes cannot be used to correctly identify other similarly selfish nodes.

6.8 Conclusions

In this chapter, we have investigated a novel machine learning-based approach to estimate interference and to detect selfish carrier-sense behavior in an 802.11 network. The technique uses a merged packet trace collected via distributed sniffing. It then recreates the MAC layer interactions on the sender-side between network nodes via a machine learning approach using the Hidden Markov Model. This coupled with an estimation of collision probability on the receiver-side is helpful in inferring the probability of interference in the network links. Significant asymmetry in the sender-side interaction in favor of a particular node witnessed by multiple other nodes indicates selfishness. The power of this technique is that it is purely passive and does not require any access to the network nodes. Although our technique works offline, it can be used periodically every few minutes (for example). Moreover, interference relationship can be used for efficient network design and capacity allocation. It can be used as a third-party solution for detecting MAC-layer misbehavior in 802.11 networks. Evaluations show the effectiveness of the tool for both the applications.

There are indeed some limitations of the technique as presented here. So far, we have estimated deferral behavior assuming only pairwise interference and have ignored physical interference (see discussions in Section 6.3.1) arguing that the improvement in accuracy will be relatively minor. Also, 802.11 retransmissions were ignored in the modeling to reduce complexity. These are not fundamental limitations and can be accommodated with higher computational cost, but are likely unnecessary. So long as enough of the common baseline case that we modeled indeed show up in the traffic trace, we will have a very good estimation accuracy. Our future work will include more evaluations to demonstrate this aspect. We will also study the impact of inaccuracy in trace gathering.

Chapter 7

Conclusions

In this dissertation, we have focused on to understand the traffic dynamics of cellular data networks and proposed several approaches both for broadband cellular and WiFi networks to help the service provider and the administrator with improved traffic management, resource allocation and monitoring schemes. We have used a large scale data set collected inside a nation-wide 3G network for our cellular network analysis and evaluation of our approaches. This has made sure that our analysis has the global view of the network from the perspective of the service provider. Specifically, we have made the following contributions.

- First, we have conducted a detailed analysis to understand the traffic dynamics in cellular data networks both from the subscriber and network perspective. We have analyzed individual subscriber behaviors, characterized subscriber mobility and temporal activity patterns, and identified their relation to traffic volume. We have investigated the efficiency in radio resource usage by different subscribers as well as by different applications. We also have analyzed the network traffic from the point of view of the base stations and found significant temporal and spatial variations in different parts of the network. Our observations have delivered important insights into network-wide resource usage and indicated implications in pricing, protocol design and resource and spectrum management.

- Then, we have shifted our focus to investigate the spatial characteristics of network resource usage in cellular data networks. We have started with characterizing the spatial correlation in radio resource usage using different statistical techniques. We have used the notion of spectral clustering to show how base stations can be clustered based on how correlated they are in terms of radio resource usage. We also have used the concept of Granger causality to understand the underlying functional connectivity and flow of influence in the network. We have showed that roughly one-third of neighboring base station pairs exhibit statistically significant Granger causality, and long causal paths exist in the network. Our observations can lead to development of new techniques for network monitoring and resource management in future cellular data networks.

- We have proposed a new traffic management technique to improve network and user perceived performance. We have considered the existence of a higher-layer, agent-based scheduling system that can potentially delay scheduling of low priority flows at peak loads. The idea is to potentially move the low priority flows in time and space opportunistically to reduce the overall resource needs. We have developed and evaluated two scheduling schemes and demonstrated the potential of these approaches in reducing base station resource requirements. This indicates that the provider can potentially accommodate a significant number of additional subscribers in the same network without expending any additional resource if only a small fraction of flows is treated as low priority.

- We have proposed an approach to develop a spatial sampling technique that estimates the loads on all the base stations based on actual measurements only on a small subset of base stations. We have used a machine learning technique to learn the underlying conditional dependence and independence structure in the base station traffic loads and exploited the model to reduce the traffic monitoring efforts. We have taken special care to develop a sparse model that focuses on capturing only key dependences. We have demonstrated the trade offs between accuracy and monitoring complexity from the perspective of two real applications.

- Finally, we have presented a tool to estimate the interference between nodes and links in a live WiFi network by passive monitoring of wireless traffic using a set of sniffers. We have used a machine learning approach to analyze the trace to infer the carrier-sense relationship between network nodes. We also have demonstrated an important application of this tool–detection of selfish carrier-sense behavior. This is based on identifying any asymmetry in carrier-sense behavior between node pairs and finding multiple witnesses to raise confidence. We have evaluated the effectiveness of the tool for both the applications using extensive experiments and simulation.

Bibliography

[1] AirMagnet WiFi Analyzer. http://www.airmagnet.com/products/wifi_analyzer/.

[2] Airpatrol's Wireless Threat Management Solutions. http://www.airpatrolcorp.com.

[3] AT&T Wireless Data Plan Press Release. http://www.att.com/gen/press-room?pid=4800\&cdvn=news\&newsarticleid=30854.

[4] Auto-regressive moving average. http://en.wikipedia.org/wiki/Autoregressive-moving-average_model.

[5] Cisco Netflow. http://www.cisco.com/warp/public/732/Tech/netflow.

[6] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20112016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.

[7] Covariance. http://en.wikipedia.org/wiki/Covariance.

[8] Cross-correlation. http://en.wikipedia.org/wiki/Cross-correlation.

[9] Endace data acquisition and generation cards. http://www.endace.com/.

[10] Jarque-Bera test. http://en.wikipedia.org/wiki/Jarque-Bera_test.

[11] Likelihood function. http://en.wikipedia.org/wiki/Likelihood_function.

[12] Lte encyclopedia. https://sites.google.com/site/lteencyclopedia/home.

[13] Message from Mobile Wireless Congress 2010. http://www.analysysmason.com/About-Us/News/Insight/The-message-from-MWC-2010/.

[14] Mobile World Congress. http://reviews.cnet.com/8301-13970_7-10454065-78.html.

[15] Multi-commodity flow problem. http://en.wikipedia.org/wiki/Multi-commodity_flow_problem.

[16] Multivariate Normal Gaussian. http://en.wikipedia.org/wiki/Multivariate_normal_distribution.

[17] The Lasso. http://www.di.ens.fr/~mschmidt/Software/lasso.html.

[18] Third generation partnership project 2 (3gpp2). http://www.3gpp2.org/.

[19] Third generation partnership project (3gpp). http://www.3gpp.org/.

[20] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, 19:716–723, 1974.

[21] F. R. Bach and M. I. Jordan. Learning spectral clustering. Technical report, EECS Department, University of California, Berkeley, Jun 2003.

[22] P. Bahl et al. DAIR: A framework for troubleshooting enterprise wireless networks using desktop infrastructure. In *Proc. ACM HotNets-IV*, 2005.

[23] P. Bahl et al. Enhancing the security of corporate Wi-Fi networks using DAIR. In *Proc. ACM MobiSys*, 2006.

[24] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

[25] R. Beckman, K. Channakeshava, F. Huang, A. Vullikanti, A. Marathe, M. Marathe, and G. Pei. Implications of dynamic spectrum access on the efficiency of primary wireless market. In *Proc. DySPAN*, 2010.

[26] G. Bianchi. Performance analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE J. on Selected Areas in Communication*, 18(3):535–547, 2000.

[27] N. Brownlee. Internet measurement. In *Proc. IEEE Internet Computing*, 2004.

[28] M. W. Bulach. *Canonical Auto and Cross Correlations of Multivariate Time*. Dissertation.Com, 1999.

[29] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux. On selfish behavior in CSMA/CA networks. In *Proc. IEEE Infocom*, 2005.

[30] J. Candia, M. Gonzalez, P. Wang, T. Schoenharl, G. Madey, and A.-L. Barabasi. Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical*, 41:1–11, 2008.

[31] H. Chang, V. Misra, and D. Rubenstein. A general model and analysis of physical layer capture in 802.11 networks. In *Proc. IEEE Infocom*, 2006.

[32] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: Performance measurements and experience. In *Proc. ACM MobiCom*, 2006.

[33] Q. Chen, H. Chang, R. Govindan, and S. Jamin. The origin of power laws in internet topologies revisited. In *Proc. IEEE INFOCOM*, 2002.

[34] W. Chen, S. Licking, T. Ohno, S. Okuyama, and T. Hamada. Performance measurement, evaluation and analysis of Push-to-Talk in 3G networks. In *Proc. IEEE International Conference on Communications*, 2007.

[35] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. *Proc. ACM SIGCOMM*, 2006.

[36] J. Chesterfield, R. Chakravorty, J. Crowcroft, P. Rodriguez, and S. Banerjee. Experiences with multimedia streaming over 2.5G and 3G networks. In *Proc. BROADNETS*, 2004.

[37] B.-Y. Choi, J. Park, and Z.-L. Zhang. Adaptive random sampling for load change detection. In *Proc. ACM Sigmetrics*, 2002. Extended Abstract.

[38] B.-Y. Choi and Z.-L. Zhang. Adaptive random sampling for traffic volume measurement. *Telecommunication Systems*, 34:71–80, 2007.

[39] M. Chuah and W. Luo. Impacts of inactivity timer values on UMTS system capacity. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 897–903, 2002.

[40] B. Clarke, E. Fokou, and H. H. Zhang. *Principles and Theory for Data Mining and Machine Learning.* Springer, 2009.

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press and McGrawHill, 2001.

[42] S. Das, D. Koutsonikolas, Y. Hu, and D. Peroulis. Characterizing multiway interference in wireless mesh networks. In *Proc. ACM WINTECH Workshop*, 2005.

[43] S. Das, H. Viswanathan, and G. Rittenhouse. Dynamic load balancing through coordinated scheduling in packet data systems. In *Proc. IEEE Infocom*, 2003.

[44] S. K. Das, S. K. Sen, and R. Jayaram. A dynamic load balancing strategy for channel assignment using selective borrowing in cellular mobile environment. *Wireless Networks*, 1997.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[46] A. Deshpande, C. Guestrin, S. R. Madden, and J. M. H. andWei Hong. Model-driven data acquisition in sensor networks. In *Proc. VLDB*, 2004.

[47] P. Deshpande, A. K. C. Sung, and S. R. Das. Predictive methods for improved vehicular WiFi access. In *Proc. ACM Mobisys Conference*, pages 263–276, 2009.

[48] I. Dobre and A. A. Alexandru. The USA shadow economy and the unemployment rate: Granger causality results. *Journal of Applied Quatative Methods*, 5, 2010.

[49] L. Du, J. Bigham, L. Cuthbert, P. Nahi, and C. Parini. Intelligent cellular network load balancing using a cooperative negotiation approach. In *Proc. IEEE WCNC*, 2003.

[50] N. Duffield. Sampling for passive internet measurement: A review. *Statistical Science*, 19:472–498, 2004.

[51] N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *Proc. ACM SIGCOMM Workshop on Internet Measurement*, 2001.

[52] N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: Control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51:1756–1775, 2005.

[53] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In *Proc. SIGCOMM*, 2000.

[54] C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems*, 21:270–313, 2003.

[55] D. Everitt and D. Manfield. Performance analysis of cellular mobile communication systems with dynamic channel assignment. *IEEE Journal on Selected Areas in Communications*, 1989.

[56] K. Fazel and S. Kaiser. *Multi-Carrier and Spread Spectrum Systems: From OFDM and MC-CDMA to LTE and WiMAX*. John Wiley & Sons, 2008.

[57] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 2001.

[58] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 2008.

[59] G. Fusco, M. Buddhikot, H. Gupta, , and S. Venkatesan. Finding green spots and turning the spectrum novel: Dial techniques for green mobile wireless networks. *IEEE DySpan*, pages 613–617, 2011.

[60] D. Ghosh and A. M. Chinnaiyan. Classification and selection of biomarkers in genomic data using Lasso. *Journal of Biomed Biotechnol*, 2005.

[61] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.

[62] C. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 1969.

[63] J. Guo, F. Liu, and Z. Zhu. Estimate the call duration distribution parameters in GSM system based on K-L divergence method. In *Proc. WiCom*, 2007.

[64] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[65] E. Halepovic and C. Williamson. Characterizing and modeling user mobility in a cellular data network. In *Proc. Workshop on PE-WASUN*, 2005.

[66] B. Harangsri, J. Shepherd, and A. H. H. Ngu. Selectivity estimation for joins using systematic sampling. In *Proc. DEXA Workshop*, 1997.

[67] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[68] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. ACM Mobicom*, 2004.

[69] C. Hiemstra and J. D. Jones. Testing for linear and nonlinear granger causality in the stock price-volume relation. *Journal of Finance*, 49, 1994.

[70] N. Hohn and D. Veitch. Inverting sampled traffic. In *Proc. IMC*, 2003.

[71] O. Holland, P. Cordier, M. Muck, L. Mazet, C. Klock, and T. Renk. Spectrum power measurements in 2G and 3G cellular phone bands during the 2006 Football World Cup in Germany. In *Proc. DySPAN*, 2007.

[72] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan. Understanding the Real-World Performance of Carrier Sense. In *Proc. E-WIND*, Philadelphia, PA, August 2005.

[73] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding congestion in ieee 802.11b wireless networks. In *Proc. ACM IMC*, 2005.

[74] Y. Jin, N. Duffield, A. Gerber, P. Haffiner, W.-L. Hsu, G. Jacobson, S. Sen, S. Venkataraman, and Z.-L. Zhang. Making sense of customer tickets in cellular networks. In *Proc. IEEE INFOCOM*, 2011.

[75] R. Joyce, B. Graves, T. Griparis, I. Osborne, and T. Lee. Case study: The capacity of a WCDMA network-Orange UK. In *Proc. 3G Mobile Communication Technologies*, 2004.

[76] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed priority scheduling and medium access in ad hoc networks. *Wireless Networks*, 2002.

[77] A. Kashyap, S. R. Das, and S. Ganguly. Measurement-based approaches for accurate simulation of 802.11-based wireless networks. In *Proc. ACM MSWIM*, 2008.

[78] A. Kashyap, S. Ganguly, and S. R. Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *Proc. ACM MobiCom*, 2007.

[79] A. Kashyap, U. Paul, and S. R. Das. Deconstructing interference relations in wifi networks. In *Proc. IEEE SECON*, 2010.

[80] J. Keilson and O. Ibe. Cutoff priority scheduling in mobile cellular communication systems. *IEEE Transactions on Communications*, 1995.

[81] R. Keralapura, A. Nucci, Z.-L. Zhang, and L. Gao. Profiling users in a 3G network using hourglass co-clustering. In *Proc. ACM MobiCom*, 2010.

[82] Y. Kim, R. Balani, H. Zhao, and M. B. Srivastava. Granger causality analysis on IP traffic and circuit level energy monitoring. In *Proc. BuildSys*, 2010.

[83] M. Kohlwes, J. Riihijärvi, and P. Mähönen. Measurements of TCP performance over UMTS networks in near-ideal conditions. In *Proc. VTC 2005-Spring*, 2005.

[84] R. R. Kompella and C. Estan. The power of slicing in internet flow measurement. In *Proc. IMC*, 2005.

[85] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proc. ACM Mobicom*, 2002.

[86] A. Kuzmanovic and E. W. Knightly. TCP-LP: Low-priority service via end-point congestion control. *IEEE/ACM Transactions on Networking*, 2006.

[87] P. Kyasanur and N. Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *Proc. IEEE DSN*, 2003.

[88] Y. Le. Measured TCP performance in CDMA 1x EV-DO networks. In *Proc. PAM*, 2005.

[89] C.-C. Lee, H. Yeh, and J.-C. Chen. Impact of inactivity timer on energy consumption in WCDMA and CDMA2000. In *Proc. Wireless Telecommunications Symposium*, 2004.

[90] J.-W. Lee, R. Mazumdar, and N. Shroff. Joint resource allocation and base-station assignment for the downlink in CDMA networks. *IEEE/ACM Transactions on Networking*, 2006.

[91] J. V. Leeuwaarden, S. Aalto, and J. Virtamo. Load balancing in cellular networks using first policy iteration. Technical report, Helsinki University of Technology, 2001.

[92] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell Syst. Tech. J.*, 62(4):1035–1074, 1983.

[93] E. Magistretti, O. Gurewitz, and E. Knightly. Inferring and mitigating a link's hindering transmissions in managed 802.11 wireless networks. In *Proc. ACM MobiCom*, 2010.

[94] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *Proc. ACM SIGCOMM*, 2006.

[95] K. Mattar, A. Sridharan, H. Zang, I. Matta, and A. Bestavros. TCP over CDMA2000 networks: A cross-layer measurement study. In *Proc. PAM*, 2007.

[96] N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 2006.

[97] A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, and R. Schober. Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. In *Proc. Innovative Smart Grid Technologies (ISGT)*, 2010.

[98] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37, 1950.

[99] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

[100] M. Osborne, B. Presnell, and B. Turlach. On the Lasso and its dual. *Annals of Statistics*, 9, 2000.

[101] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *Proc. Internet Measurement Conference (IMC)*, 2005.

[102] P. Pathirana, A. Savkin, and S. Jha. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In *Proc. ACM MobiHoc*, pages 213–221, 2003.

[103] U. Paul, M. Buddhikot, and S. R. Das. Opportunistic traffic scheduling in cellular data networks. In *Proc. DySPAN*, 2012.

[104] U. Paul, S. R. Das, and R. Maheshwari. Detecting selfish carrier-sense behavior in wifi networks by passive monitoring. In *Proc. IEEE DSN*, 2010.

[105] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das. Understanding traffic dynamics in cellular data networks. In *Proc. IEEE INFOCOM*, 2011.

[106] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das. Understanding spatial relationship in resource usage in cellular data networks. In *Proc. IEEE NetSciCom*, 2012.

[107] V. Paxson. End-to-end routing behavior in the internet. *IEEE/ACM Transactions on Networking*, 5:601–615, 1997.

[108] K. Pelechrinis, G. Yan, S. Eidenbenz, and S. V. Krishnamurthy. Detecting selfish exploitation of carrier sensing in 802.11 networks. In *Proc. IEEE Infocom*, 2009.

[109] C. Peng, S.-B. Lee, S. Lu, H. Luo, and H. Li. Traffic-driven power saving in operational 3g cellular networks. In *Proc. ACM Mobicom*, 2011.

[110] K. Pentikousis, M. Palola, M. Jurvansuu, and P. Perl. Active goodput measurements from a public 3G/UMTS network. *IEEE Communications Letters*, 9:802–804, 2005.

[111] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Web caching on smartphones: Ideal vs. reality. In *Proc. ACM MobiSys*, 2012.

[112] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3G networks. In *Proc. Internet Measurement Conference*, 2010.

[113] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *Proc. ACM MobiCom*, 2007.

[114] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.

[115] S. Radosavac, J. S. Baras, and I. Koutsopoulos. A framework for mac protocol misbehavior detection in wireless. In *Proc. ACM workshop on Wireless Security*, 2005.

[116] T. S. Rappaport. *Wireless Communications: Principles and Practice.* IEEE Press, Piscataway, NJ, USA, 1996.

[117] M. Raya, J.-P. Hubaux, and I. Aad. Domino: A system to detect greedy behavior in ieee 802.11 hotsopts. In *Proc. ACM Mobisys*, 2004.

[118] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee. Diagnosing wireless packet losses in 802.11: Separating collision from weak signal. In *Proc. IEEE Infocom*, 2008.

[119] P. Reichl, M. Umlauft, J. Fabini, R. Lauster, and G. Pospischil. Project WISQY: A measurement-based end-to-end application-level performance comparison of 2.5G and 3G networks. In *Proc. Fourth Ann. Wireless Telecomm. Symp. (FTS)*, 2005.

[120] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *Proc. ACM SIGCOMM*, 2006.

[121] J. Riihijärvi, P. Mähönen, M. Wellens, and M. Gordziel. Characterizing and modelling of spectrum for dynamic spectrum access with spatial statistics and random fields. In *Proc. PIMRC*, 2008.

[122] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *Proc. ACM E-WIND*, 2005.

[123] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan, and E. Lazowska. CRAWDAD data set uw/sigcomm2004. http://crawdad.cs.dartmouth.edu/uw/sigcomm2004.

[124] S. Rosset. Tracking curved regularized optimization solution paths. *NIPS*, 2004.

[125] E. Santacana, G. Rackliffe, L. Tang, and X. Feng. Getting smart. *IEEE Power and Energy Magazine*, 2010.

[126] G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 5:461–464, 1978.

[127] A. K. Seth. Causal connectivity analysis of evolved neural networks during behavior. *Network:Computation in Neural Systems*, 16:35–54, 2005.

[128] A. K. Seth. A matlab toolbox for granger causality connectivity analysis. *Journal of Neuriscience Methods*, 186:262–273, 2010.

[129] A. K. Seth and G. Edelman. Distinguishing causal interactions in neural populations. *Neural Computation*, 19:910–933, 2007.

[130] W. Shi, G. Wahba, S. Wright, K. Lee, R. Klein, and B. Klein. Lasso-patternsearch algorithm with application to ophthalmology and genomic data. *Statistics and Its Interface*, 2008.

[131] A. P. Subramanian, H. Gupta, S. R. Das, and M. M. Buddhikot. Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks. In *Proc. DySPAN*, 2007.

[132] W. Tan, F. Lam, and W. Lau. An empirical study on the capacity and performance of 3G networks. *IEEE Transactions on Mobile Computing*, 7:737–750, 2008.

[133] J. Tang, Y. Cheng, Y. Hao, and C. Zhou. Real-time detection of selfish behavior in IEEE 802.11 wireless networks. In *Proc. IEEE VTC-Fall*, 2010.

[134] R. TibShirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*, 58, 1996.

[135] R. TibShirani. The Lasso method for variable selection in the Cox model. *Statistical Medicine*, 16, 1997.

[136] O. K. Tonguz and E. Yanmaz. The mathematical theory of dynamic load balancing in cellular networks. *IEEE Transactions on Mobile Computing*, 2008.

[137] C. Williamson. Internet traffic measurement. In *Proc. IEEE Internet Computing*, 2001.

[138] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary users in cellular networks: A large-scale measurement study. In *Proc. DySPAN*, 2008.

[139] A. S. X. Liu, S. Machiraju, M. Seshadri, and H. Zang. Experiences in a 3G network: Interplay between the wireless channel and applications. In *Proc. ACM MobiCom*, pages 211–222, 2008.

[140] X. Yang and N. H. Vaidya. Priority scheduling in wireless ad hoc networks. In *Proc. ACM Mobihoc*, 2002.

[141] J. Yao, S. Kanhere, and M. Hassan. An empirical study of bandwidth predictability in mobile computing. In *Proc. ACM WiNTECH*, pages 11–18, 2008.

[142] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless lan monitoring and its applications. In *Proc. ACM WiSe*, 2004.

[143] L. Zelnik-manor and P. Perona. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, 17, 2004.

[144] F. Zhang and Z. Lei. The evaluation of poisson packet sampling measurement techniques. In *Proc. APCC*, 2004.

[145] P. Zhao and B. Yu. Boosted Lasso. Technical report, Statistics Department, UC Berkeley, 2004.

[146] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall. Sampling and filtering techniques for IP packet selection. *RFC5475*.