On the Interaction between TCP and the Wireless Channel in CDMA2000 Networks

Karim Mattar* Ashwin Sridharan† Hui Zang† Ibrahim Matta* Azer Bestavros*
Computer Science Department, Boston University*
Sprint Advanced Technology Labs†

Technical Report BUCS-TR-2006-009
Sprint Technical Report RR06-ATL-052666

Abstract

In this work, we conducted extensive active measurements on a large nationwide CDMA2000 1xRTT network in order to characterize the impact of both the Radio Link Protocol and more importantly, the wireless scheduler, on TCP. Our measurements include standard TCP/UDP logs, as well as detailed RF layer statistics that allow observability into RF dynamics. With the help of a robust correlation measure, *normalized mutual information*, we were able to quantify the impact of these two RF factors on TCP performance metrics such as the round trip time, packet loss rate, instantaneous throughput etc. We show that the variable channel rate has the larger impact on TCP behavior when compared to the Radio Link Protocol. Furthermore, we expose and rank the factors that influence the assigned channel rate itself and in particular, demonstrate the sensitivity of the wireless scheduler to the *data sending rate*. Thus, TCP is adapting its rate to match the available network capacity, while the rate allocated by the wireless scheduler is influenced by the sender's behavior. Such a system is best described as a closed loop system with two feedback controllers, the TCP controller and the wireless scheduler, each one affecting the other's decisions. In this work, we take the first steps in characterizing such a system in a realistic environment.

I. Introduction

With advances in coding theory and cellular technology, the wireless channel need no longer be seen as an error-prone channel with constant low bandwidth. Instead, dynamic "on-the-fly" coding allows the channel to adapt to diverse noise conditions. Variable coding rates trade-off channel capacity for low bit error rate (BER) allowing the channel to operate at lower rates when there is excessive noise and at higher rates during good conditions, with almost the same BER. Current and next generation cellular networks (e.g. CDMA 2000 1xRTT and 1xEV-DO [14]) have taken this a step further by specifying sophisticated schedulers that not only incorporate channel conditions, but also take data backlog into consideration when determining coding rates.

Consequently, the channel experienced by a source has variable bandwidth and delay, which in turn are partly influenced by the data rate of the source. The dependence of the channel behavior on the source introduces a feedback control loop which to the best of our knowledge has not been studied previously, especially in the context of TCP transmissions. Specifically, TCP uses feedback from the channel to modulate its congestion window to match the capacity of the channel. On the other hand, modern high-speed cellular channel schedulers utilize the backlog in the data buffer (which is influenced by the source's sending rate), to determine what wireless transmission rates to assign. In this work, we take the first steps in characterizing such a system. Our contributions are as follows:

1) We conducted extensive active measurements in a commercial cellular network to characterize the behavior of the RF channel and evaluate the performance of TCP over such channels.
2) We provide a simple Information Theoretic framework for quantifying the correlation between different system performance metrics that explains the dependence and interplay of two controllers: the TCP controller and the wireless scheduler.

3) In terms of the RF channel, we exposed the different mechanisms that govern the operation of the channel rate scheduler and identified the main characteristics that influence its performance. We concluded that the channel rate scheduler: a) is extremely sensitive and highly dependent on the buffer occupancy over short time scales; b) performs poorly when traffic sources are more aggressive in acquiring bandwidth; c) is surprisingly insensitive to variations in channel quality and load (number of users); and d) has a rate limiting mechanism to maintain fairness by reducing the rate assigned to connections that are classified as being persistently greedy.
4) In terms of evaluating TCP's performance, we concluded that: a) the link layer optimizations (e.g. adaptive coding, power control) attempt to hide wireless errors over short timescales; b) to absorb such short-term variabilities, a large buffer is used at the Base Station Controller as evidenced by high measured delays; c) as a result, congestion induced losses are more prevalent than wireless losses, and the large buffer at the BSC introduces significant feedback delay until the TCP sender detects such congestion-induced losses; d) this feedback delay is even more pronounced when the buffer-dependent wireless scheduler rate-limits the TCP connection, especially when TCP is not limited by flow control; e) more aggressive sources achieve lower throughput as a result of higher oscillatory behavior.

The rest of the paper is organized as follows. Section II outlines the architecture of a CDMA2000 network and highlights the relevant features that we study. Section III presents a description of the various experiments that we conducted, as well as the metrics that we measured. Section IV explains our empirical evaluation methodology which is based on correlating time series which capture the evolution of various system parameters. Section V characterizes the various elements of the RF channel and quantifies their impact on the wireless scheduler. Section VI presents an evaluation of TCP's performance over the wireless channel. Finally, Section VII summarizes our conclusions.

II. The CDMA2000 1xRTT System

In this section, we illustrate the architecture of modern cellular data networks, as well as identify salient properties of CDMA2000 1xRTT, a 2.5G technology, which is widely used in these networks and was the technology available at the time we conducted our experiments. In particular, we highlight key features of the CDMA2000 network, that either directly or indirectly affect higher layer performance and motivate the need to characterize their impact in realistic environments. We believe our findings may also be applicable to current 3G networks based on the 1xEV-DO technology [14] because they share some similar features.

A. Network Architecture

Figure 1 sketches the architecture of a typical cellular data network. The network consists of two main components: the radio network and the data network. The data network is an all-IP network comprising of the PDSN (Packet Data Serving Node), the HA (Home Agent) and the AAA (Authentication, Authorization and Accounting) server. The PDSN, residing in the data network, acts as the interface agent between the two networks. It establishes a PPP session for each cellular user and forwards traffic received from the radio network to the HA and vice versa. The HA is responsible for IP address allocation, forwarding cellular IP traffic to (and from) the Internet and more importantly, manages user mobility via Mobile IP [6]. The AAA server mainly addresses the requirements of authentication, billing etc.

The radio network, which is actually the focus of this study, comprises the air interface and two basic elements: a Base Transceiver Station (BTS) and a Base Station Controller (BSC). The BTS, or simply put, the base station, is essentially a "dumb terminal" in the CDMA2000 1xRTT network, comprising only of antenna arrays to efficiently radiate RF (Radio Frequency) power to mobile users, as well as receive signals from them. Hence, it acts as the interface between the "wireline" network and the "wireless" hop. Each such base station represents a "cell". For purposes of efficient frequency re-use, the cell is typically split into three sectors by suitable alignment of the antenna profile into three geographically

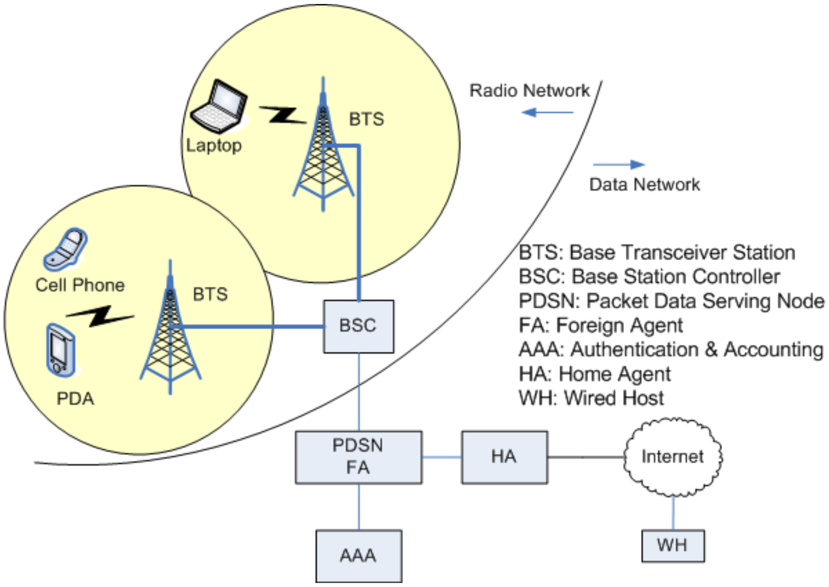

Radio Network
Data Network
Laptop
BTS
Cell Phone
BTS
PDA
BSC
PDSN
FA
HA
Internet
AAA
WH
BTS: Base Transceiver Station
BSC: Base Station Controller
PDSN: Packet Data Serving Node
FA: Foreign Agent
AAA: Authentication & Accounting
HA: Home Agent
WH: Wired Host

Fig. 1. Sprint's Network Architecture

distinct radiation beam patterns. Users in the same cell but different sectors can operate independently, but users in the same sector must share air resources.

The BSC, which is actually the main element of the radio network, is an intelligent agent that can control up to 400 base stations (or cells) that are connected to it through a low latency back-haul network. The BSC is responsible for almost all the RF layer operations that are critical for the smooth operation of the CDMA network. Among other things, it manages power control operations for all mobile users to limit interference and also controls soft-handoff [18] as users move.

From the perspective of a transport layer such as TCP, the two most critical actions controlled by the BSC that explicitly affect TCP's performance are: a) the channel *rate* allocation on the wireless hop to *each user* on both the downlink[1] *and* the uplink and b) the Radio Link Protocol (RLP), which is a link layer error and loss recovery mechanism.

While various other RF related factors like channel conditions, number of users, channel errors etc., clearly affect higher layer performance, as explained in later sections, their impact is subsumed in these two functions. In other words, they indirectly affect application performance by either affecting the channel rate or RLP behavior. Consequently, in this work, we view them as secondary factors, while the channel rate and RLP as primary factors that directly affect higher layer performance.

B. Dynamic Rate Allocation

Current and next generation cellular data networks possess the ability to *dynamically* vary the *rate* of the wireless channel assigned to a user through a combination of adaptive coding, modulation and orthogonal Walsh Codes. Clearly, variation in the assigned channel rate has a direct impact on the throughput perceived by higher layer protocols.

In a CDMA2000 1xRTT network, this operation is performed by the BSC primarily by changing the Walsh Code. Specifically, the BSC can assign a higher (lower) rate to a mobile by assigning a shorter (longer) Walsh Code. Depending on the Radio Configuration Type [18] a CDMA2000 1xRTT network can support up to six different channel rates. The network utilized for our experiments supports five channel rates. The smallest assignable rate, denoted by the *Fundamental Channel* (FCH) is 9.6 kbps. This is the standard channel assigned to all voice users and initially to a data user upon joining the network. If a

[1] The downlink is the path from the BSC to the user, while the uplink is the path from the user to the BSC.

user requires higher data rates, the BSC can assign it a *Supplemental Channel* (SCH) in bursts of short durations. The Supplemental Channel can take rates from the set $\{\mathbf{19.2}, \mathbf{38.4}, \mathbf{76.8}, \mathbf{153.6}\}$ kbps.

Though a shorter Walsh Code increases the data rate, it has two drawbacks. First, the reduced code length degrades "orthogonality", which makes the signal more susceptible to interference from other users. Second, this increases its impact as interference on *other* users. In order to minimize this adverse behavior, the BSC employs two techniques:

1) The signal strength is boosted for users assigned a higher rate channel to overcome increased interference.
2) When a user is assigned a higher rate channel, fewer users are allowed to simultaneously transmit at high rates to reduce interference. The higher the rate is, the fewer the number of users that can be assigned this rate simultaneously. In the extreme case, only one user can be assigned the highest rate, 153.6 kbps, at any point in time. Consequently, in order to provide fairness, high rate channels are allocated only in short bursts.

The BSC may assign a supplemental channel with the appropriate rate to a user based on the following potential *secondary* factors:

- *Buffer backlog* : The CDMA2000 1xRTT network deploys a per-user buffer at the BSC which is routinely monitored by the scheduler. A large data backlog is more likely to trigger assignment of a high rate Supplemental Channel for the user.
- *Channel conditions* : Each base station transmits a continuous *Pilot Signal* which is received by all users in the cell. The user then determines the channel condition by computing the Pilot SINR E_c/I_o, where E_c represents the strength of the Pilot Signal received and I_o the interference due to other users and thermal noise. A low value indicates poor channel conditions (or high loss) and vice versa[2]. This value is fed back to the BSC which utilizes this information in deciding what rate to assign. A poor channel may result in a reduction in the assigned rate to minimize channel losses.
- *Sector load* (in terms of number of users) : As mentioned earlier, shorter Walsh Codes (at the same power) experience higher interference and also cause more interference. Hence, whenever the BSC transmits a high rate SCH burst to a user, it may prevent other users from transmitting at high rates. Consequently, the BSC must take into consideration the number of other active users in a sector before determining what rate to assign.

C. Radio Link Protocol: RLP

Apart from wireless channel rate allocation, the other feature of the BSC that can directly affect higher layer performance is the Radio Link Protocol (RLP). The RLP is a NACK-based ARQ re-transmission mechanism developed in order to minimize the losses perceived by higher layers. The motivation for such a mechanism is the high latency on wireless links which can induce large delays before end-to-end recovery mechanisms sense and recover from a packet loss.

The BSC maintains an RLP session with each mobile user which works as follows. The BSC breaks incoming IP packets from the PDSN into radio frames which are then transmitted to the mobile via the BTS. The mobile, on detecting missing (or corrupted) RLP frames requests re-transmission of the corresponding RLP frames.

D. Impact of Channel Assigned Rate and RLP on TCP

It has been traditionally assumed that the RLP re-transmission rate is closely related to the Frame Error Rate (FER) of the channel. In this context, the impact of the Radio Link Protocol on TCP has been

[2] Typical values for good channels are around -3 to -7 dB, while values less than -11dB indicate a poor channel.

extensively researched theoretically [9], [13], [2], [4] from the perspective of trade-off between reduced error probability and increased latency to maximize throughput. The link layer increases the reliability seen by higher layers through re-transmissions or stronger error correcting codes. Both mechanisms attempt to reduce the likelihood of TCP throttling its sending rate due to packet losses. On the other hand, these mechanisms increase latency since packets are retained longer by the link layer for successful transmission, which in turn can degrade throughput.

To the best of our knowledge, *neither* the RLP re-transmission rate and its dependence on the channel FER, *nor* the impact of the link layer on TCP dynamics have been quantified *in practice* on commercial networks. Even less research has been conducted on the impact of dynamic wireless channel rate allocation on TCP performance (see [1], [7] for models) or to which extent each of the secondary factors influences the channel assigned rate. The only experimental study we are aware of is the one presented in [8] which evaluated the impact of bandwidth variation in CDMA2000 networks on the TCP timeout mechanism in a lab environment.

As discussed in Section II-B, the assigned channel rate can be affected by three secondary factors: channel conditions, application data rate and sector load. However, it is not clear in practice which factor dominates. If the channel conditions play the dominant role in determining the assigned channel rate, an argument similar to that for RLP could potentially be made in that it, too, trades-off channel bandwidth to minimize channel errors, and thus should have an impact similar to that of RLP.

However, an important difference from RLP is that the *data sending rate* of the higher layer protocol also affects the assigned channel rate. This is crucial because, TCP, the most widely used transport layer protocol, is a *reactive protocol* which adjusts its rate based on feedback from the receiver. Hence, the system becomes a closed-loop system where both TCP and the BSC scheduler vary their rate based on feedback from the other. This may result in unexpected interactions between the two control regimes, possibly leading to performance degradation.

The objective of this study is to precisely characterize these issues. Through a series of experiments, we evaluate which secondary RF layer factors affect the assigned channel rate and the Radio Link Protocol the most. We also study the impact of both RLP and the wireless scheduler on TCP dynamics.

III. Experiments and Data sets

In this section we outline our experiments and the data collection process. Our experiments were primarily end-to-end in nature, involving transfer of data via either UDP or TCP from a RedHat Linux server to a laptop running Windows XP that was connected to the cellular data network through a CDMA2000 1xRTT aircard. The experimental set-up is shown in Fig. 2.

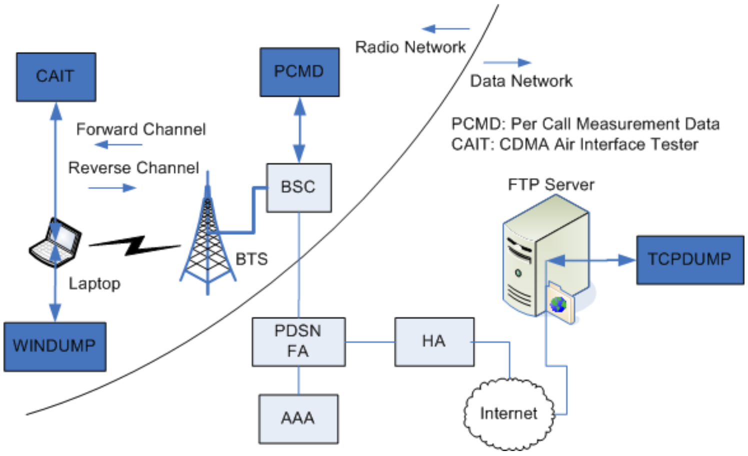


Fig. 2. Experimental Setup

For each experiment, we collected data from higher layer protocols through standard UDP/TCP logs at the client and server, as well as RF channel statistics for the various parameters discussed in Sections II-B and II-C. The experiments are described in detail below.

A. *UDP Experiments*

The UDP experiments were primarily utilized to characterize the behavior of the RF channel. More specifically, these experiments were utilized to study how the various secondary factors identified in the previous section, such as channel conditions, frame error rate, sector load and higher layer data sending rate, affect the two primary RF factors, namely the channel assigned rate and the RLP re-transmission rate.

We chose UDP because the *higher layer data sending rate* is one of the secondary factors affecting the wireless scheduler. Since UDP is non-reactive, it is not subject to the feedback loop mentioned in Section II-D and hence does not vary its data sending rate. This allowed us to quantify the impact of the other secondary factors on the assigned channel rate without worrying about feedback.

The experiments consisted of sending either constant bit rate (CBR) traffic with different *rates* or on-off traffic with different *on-off durations* and different *peak rates*. The typical duration of each experiment was around 20 minutes. The complete set of UDP experiments is listed in Table I.

TABLE I
UDP EXPERIMENTS: NUMBER OF EXPERIMENTS FOR DIFFERENT TYPES AND RATES.
FOR ON-OFF EXPERIMENTS, THE RATE IS THE PEAK RATE.

	Rate (kbps)								
Type	9.6	19.2	38.4	48.0	57.6	78.6	115.2	144	153.6
CBR	1	3	3	6	2	3	0	0	1
On-Off	0	1	0	0	2	2	2	1	2

B. *TCP Experiments*

We conducted a large number of TCP downloads of 5MB files to characterize the impact of the RF channel on TCP. Experiments were conducted with different TCP advertised receive window (*arwnd*) sizes ranging from 8KB to 64KB. The TCP version used was the SACK-enabled Linux TCP [16].

For each experiment, we studied the relationship between the two primary RF layer characteristics, namely assigned channel rate and RLP and various TCP characteristics such as round trip time, packet loss rate, instantaneous throughput etc., as well as compared their relative impact on TCP performance.

The complete set of TCP experiments is listed in Table II. We also conducted TCP experiments with multiple users and multiple sessions to evaluate the fairness of the wireless scheduler but omit the results from this paper.

TABLE II
TCP EXPERIMENTS: NUMBER OF EXPERIMENTS FOR DIFFERENT ADVERTISED RECEIVE WINDOW SIZES

ARWND (KB)	8	12	16	24	32	48	64
# of Exps.	7	7	10	8	7	6	7

C. *Data Collection*

The UDP and TCP dynamics were monitored using windump [21] at the client and tcpdump [17] at the server. For the UDP experiments, the client and server also dumped log files giving detailed packet arrival information. These logs were utilized to compute time series of various higher layer metrics such as the instantaneous data sending rate, Round Trip Time (RTT) and packet loss rate. Data related to the RF channel were collected from two sources:

1) CDMA Air Interface Tester (CAIT): CAIT [15] is a tool developed by Qualcomm for testing and analyzing the air interface. It logs layer-2 statistics of the RF channel to and from a mobile device in a cellular network allowing us to monitor a rich variety of channel related information. We used CAIT to log the following information for each experiment:

- Assigned channel rate[3]
- Channel condition (E_c/I_0)
- RLP re-transmission rate
- Frame Error Rate (FER)

We used a tool called "*WindCatcher*" [20] to parse the CAIT logs and generate time series for each one of these metrics. The smallest time granularity for *WindCatcher* to generate these time series is one second. Hence, one second was the smallest time scale at which we observed events[4].

2) Per Call Measurement Data: Recall that apart from channel conditions and user sending rate, the *sector load* is another factor that can potentially affect decisions made by the RF scheduler regarding the assigned channel rate. For CDMA, the sector load is simply the number of users in the sector where the laptop conducting the experiment is located. We utilized a measurement feature supported by the BSC, called *Per Call Measurement Data* (PCMD), to infer the number of users in a particular sector. PCMD logs contain measurement data for all voice and data calls, such as call type, call length and call location (in terms of cell and sector) and allow us to determine the call volume in a particular sector during any experiment.

IV. EMPIRICAL EVALUATION : METHODOLOGY AND TOOLS

In this section we explain the methodology and tools we used to analyze the experiments. Recall that our goals are two-fold: one, quantitatively characterize the impact of the various secondary RF factors on the assigned channel rate and RLP and two, perform a similar characterization of the impact of these two primary factors on TCP.

In order to achieve these goals, we must be able to measure the effect of different performance metrics and parameters on one another. For some of the objectives we can rely on standard statistical metrics like the expected mean. However, a large portion of our analysis involves quantifying the correlation between performance metrics and parameters that come in the form of time series, capturing the evolution of different aspects of our system. To tackle this aspect of our study, we require a robust technique to evaluate the correlation between time series.

We chose *normalized mutual information* as the correlation measure to accomplish this task. Section IV-A introduces the metric and motivates this choice. We were faced with some implementation-related technicalities when applying this measure. They, as well as the relevant solutions are discussed in Sections IV-B and IV-C.

A. Mutual Information as a Correlation Measure

There are numerous correlation measures that have been extensively used in the literature. The most commonly used ones being Pearson's correlation coefficient and covariance. These techniques are limited, however, to only being able to measure linear dependencies. *Mutual information*, on the other hand, is a correlation measure that can be generalized to all kinds of probability distributions and is able to detect non-linear dependencies between variables. Consequently, since it was unknown whether the system under consideration was linear or not, we use mutual information in our work to correlate between time series.

[3] This is the actual rate assigned by the network on the forward link. The amount of data actually transmitted may be less.

[4] CDMA2000 1xRTT rate assignment durations are typically at least 320 ms and hence the one-second granularity allows us to observe an aggregate of 2-3 such allocations.

Mutual information can be thought of as the reduction in uncertainty (entropy) of one variable due to the knowledge of the other. It is mathematically defined as follows. Let X denote a discrete random variable that takes a value $x \in \mathcal{X}$ with probability $p(x)$. The *entropy* of X is given by the well-known definition [10]:

$$H(X) = -\sum_{x_i \in \mathcal{X}} p(x_i) \log p(x_i) \tag{1}$$

The mutual information between two random variables X and Y is then given by:

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) & (2) \\ &= H(X) - H(X|Y) & (3) \end{aligned}$$

where $H(X,Y)$ represents the joint entropy of random variables X and Y, and $H(X|Y)$ represents the conditional entropy of X given Y.

In order to obtain a consistent interpretation of the correlation measure across different experiments, we utilize the *normalized mutual information* (NMI), defined as:

$$I_N(X;Y) = \frac{I(X;Y)}{H(X)} = 1 - \frac{H(X|Y)}{H(X)} \tag{4}$$

To illustrate the intuition behind $I_N(X;Y)$, assume Y completely determines X (i.e., Y captures all the information in X), then $H(X|Y)$ would be close to 0 and $I_N(X;Y)$ would be close to 1. On the other hand, if Y contains no information about X, then $H(X|Y)$ would be close to $H(X)$ and $I_N(X;Y)$ would be close to 0. The closer $I_N(X;Y)$ is to 1, the larger the amount of information that Y carries about X. Note that $I_N(X;Y)$ is asymmetric. Eqn. 4 computes the *relative* amount of information that Y contains about X given the entropy of X. If we simply wanted to compute the normalized mutual information irrespective of direction, we could divide $I(X;Y)$ by $min(H(X), H(Y))$. In our work, we are more interested in the amount of information that one variable has about another and therefore chose to use Eqn. 4.

We also evaluated other variants of mutual information: 1) *mutual information of state transitions* where each sample in a time series represents a state and we are interested in capturing dependencies in the *transitions* between these states as opposed to the states themselves, 2) *mutual information of magnitude variations* where we are interested in capturing dependencies in the magnitude changes between consecutive samples, and 3) *mutual information rate* proposed by Gillblad et al. in [12], which is more suited for correlating time series than mutual information but requires making assumptions about the probability distributions of the variables being correlated to obtain meaningful results. The use of these correlation measures only supported the conclusions we made based on normalized mutual information and the results were therefore omitted from this paper.

Next, we address two key issues that we faced in utilizing the normalized mutual information (NMI) as our correlation measure: 1) accounting for delays between time series when performing time series correlation and 2) the discretization of time series to compute the joint and marginal probability distributions necessary for evaluating NMI.

B. Time Series Correlation

In general, when correlating two time series capturing the evolution of two processes, one must consider possible delays between them because of the potential time lag between when a state change in one process actually affects the other. For example, if we were to correlate the instantaneous data sending rate (measured at the sender) with the instantaneous data receiving rate (measured at the receiver), we need to consider the one-way delay between the sender and the receiver (including any possible queuing delays in the network). To overcome this problem, we compute the normalized mutual information between each pair of time series over a wide range of possible time shifts. The mutual information is now defined as:

$$I(X;Y;d) = H(X) + H(Y) - H(X, Y_d) \tag{5}$$

where $H(X, Y_d)$ denotes the joint entropy of the random variable X and a time-delayed version of Y if $d > 0$ or a time-advanced version of Y if $d < 0$. The NMI is then defined as:

$$I_N(X;Y;d) = \frac{I(X;Y;d)}{H(X)} \tag{6}$$

C. Time Series Discretization

Observe from Eqn. 3 that in order to use mutual information to quantify the correlation between two time series, we need to estimate the marginal and joint probability distributions of both time series. Since time series like RTT are real-valued, they must be discretized for this purpose. Towards this end we utilized two techniques for discretization[5].

The first technique we used was proposed by Dimitrova et al. [11] which seeks to 'bin' any real-valued time series data into a finite number of discrete values. The algorithm assumes no knowledge about the distribution, range or discretization thresholds of the data. It is based on the single-link clustering (SLC) algorithm and aims to minimize the information loss (measured by the entropy), which is inherent to any discretization. The algorithm has also been shown to maintain prior correlation between the original time series, which was one of our main criteria for the selection of this technique.

Although we found the technique to be quite effective, the range of time series behavior in our experiments is quite large and there were cases where discretization by this technique fails to capture important properties. This was especially significant in cases with slowly varying signals with sudden variations. Hence, we also utilized standard binning with equidistant bin sizes as an alternate discretization technique for such cases. The bin sizes we chose were: 10kbps for rate time series, and 500ms for RTT time series. For purposes of verification, we compared equidistant binning with the discretization technique by Dimitrova et. al. in cases where the latter worked. The NMI values were found to be the same thus confirming the suitability of these values.

It finally remains to discuss how the Normalized Mutual Information, $I_N(X;Y;d)$, was used to identify potential correlation. Clearly in order to identify any potential correlation, the value of $I_N(X, Y; d)$ needs to be sufficiently 'large'. However, like most measures, the NMI is a continuous metric. Hence, it only measures the 'strength' of correlation and it is up to the interpreter to judge whether the strength is sufficiently large. In general, it is very hard to justify choice of a specific threshold to categorize the strength of correlation as large or small. This difficulty is compounded even further in our case because: a) discretization of the time series can introduce noise and b) strong correlation between two time series requires the presence of a characteristic delay over which they interact. However, the stochastic nature of the system we study can 'spread' the delay range over which the time series are correlated. Both factors can not only *lower* the intrinsic NMI values, but also magnify the potential range of NMI values that a correlated pair of time series may take.

To circumvent this issue, we exploit two aspects in our study. One, when studying any feature like TCP RTT or assigned channel rate, we are only interested in the *relative impact* of various factors on this feature. Hence, we need only focus on the relative NMI values. Two, if strong correlation exists between two time series, the NMI values should peak at some characteristic delay despite the distortion in delay due to stochastic perturbation. Put another way, a sharp NMI peak at a particular time shift[6] indicates the

[5] Most of the time series we were correlating were discrete in nature thus allowing us to model them as *discrete* random variables.

[6] We use the terms *delay* and *time shift* interchangeably throughput this paper.

presence of correlation large enough to overcome potential delay perturbations. This is also corroborated by our experiments, where NMI values between two time series that did not exhibit any strong peak invariably had very low values (compared to those that exhibited peaks).

We found these two guidelines to be quite useful in analyzing the various features on a case-by-case basis without having to resort to choosing a specific threshold value for strong correlation. Specifically, when studying the impact of various factors on a given feature, as a first step the sharpness of the NMI curves as a function of the time shifts helps narrow the potential correlations. The peak value of the NMI curve for each factor is then used to rank the relative strength of correlation.

D. Analysis at Multiple Time Scales: Wavelets

The last aspect of evaluation that we wish to touch upon is the time scale of different events. Specifically, some RF factors like channel rate, RLP re-transmissions and channel conditions vary over a very small time scale[7] while others like the sector load change more slowly. Similarly, TCP reacts at the time scale of round trip times, which for wireless links, we show can be in the order of seconds.

Consequently, it is of interest to study the correlation between time series of various parameters at different time scales. For example, an important case that we study is whether changes in TCP sending rate at small time-scales are correlated to the rapid variations in the wireless channel rate.

Wavelet is an ideal tool for the purpose of multi-time-scale analysis. We employed the wavelet decomposition strategy outlined by the authors of [3] to decompose each time series into low and high frequency signals. The low frequency signal extracts the general slow-varying trend of the original signal. The high frequency signal captures the fine-grained details of the original signal, such as spontaneous variations. Continuing with our example, correlation between the high frequency time series of the TCP sending rate and the wireless channel rate would indicate that the former is affected by (or causes) rapid changes in the latter. Similarly, correlation of the low frequency signals would indicate that the TCP sending rate tracks the assigned channel rate over long time scales (or vice versa). In Fig. 3 we show the decomposition of a sample forward channel assigned rate signal. In all our wavelet decompositions, the slow-varying component captured variations over a 32-second duration (which is roughly 10 times the average RTT for many of our connections) and the fast-varying component captured variations over a 2-second duration (which is less than the average RTT for many of our connections).

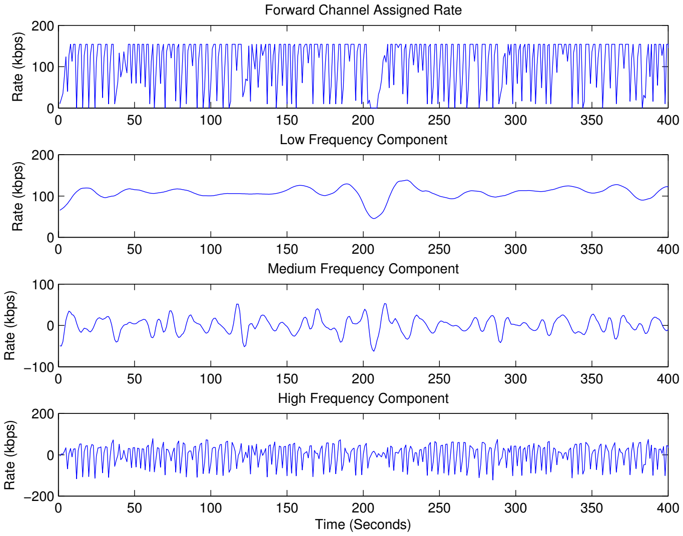


Fig. 3. Wavelet Decomposition of a Sample Forward Channel Assigned Rate Signal

[7] In our experiments the observable variations are lower bounded by 1 sec.

V. Characterization of the primary RF Factors

In this section, we study the behavior of the CDMA2000 wireless channel. In particular, our focus is on two key components: the channel rate scheduler and the Radio Link Protocol, since both components have a direct effect on higher layer performance.

In order to probe the cellular channel and characterize its behavior, we initiated many UDP sessions. We chose the UDP traffic class since it is not influenced by feedback from the channel thus reducing the system to an open loop system. This allows us to study the behavior of the wireless channel in isolation. We implemented a UDP application consisting of a client and a server capable of transmitting data in many different traffic patterns.

A. Wireless Scheduler

Recall from Section II-B, that the wireless scheduler's decisions can be affected by three factors:

- the data sending rate or buffer backlog,
- the channel conditions, and
- the sector load

We quantify the impact of each of these factors in this section.

1) Impact of Data Sending Rate and Buffer Backlog: We performed numerous UDP experiments using constant bit rate (CBR), as well as on-off traffic sources where the on and off durations, as well as the peak data rate were varied. The CBR traffic source allows us to determine if the scheduler tracks the user's sending rate over long time scales. An on-off traffic source, on the other hand allows us to probe the channel with a bursty-like data source to see if the channel rate scheduler is able to track the data source over short time scales.

Figure 4 plots the average throughput obtained by a UDP connection over the entire experiment's duration (1200secs) versus the different data sending rate for the CBR experiments. Observe that for data rates up to about 50 kbps, the channel tracks the source closely and is able to honor the requested rate. Beyond that, however, the throughput drops significantly. We hypothesize that this is because of a "rate-limiting" mechanism built in the scheduler which preempts users that are persistently greedy. We will discuss this aspect further in Section V-A.5.

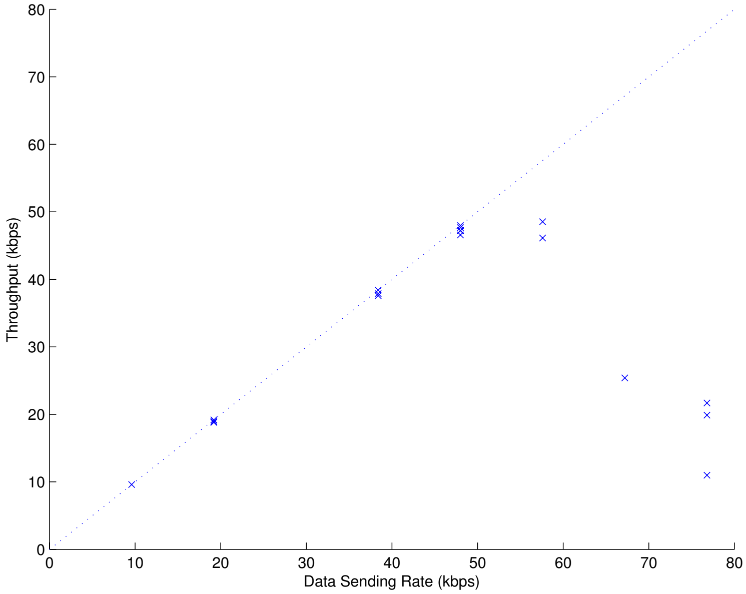
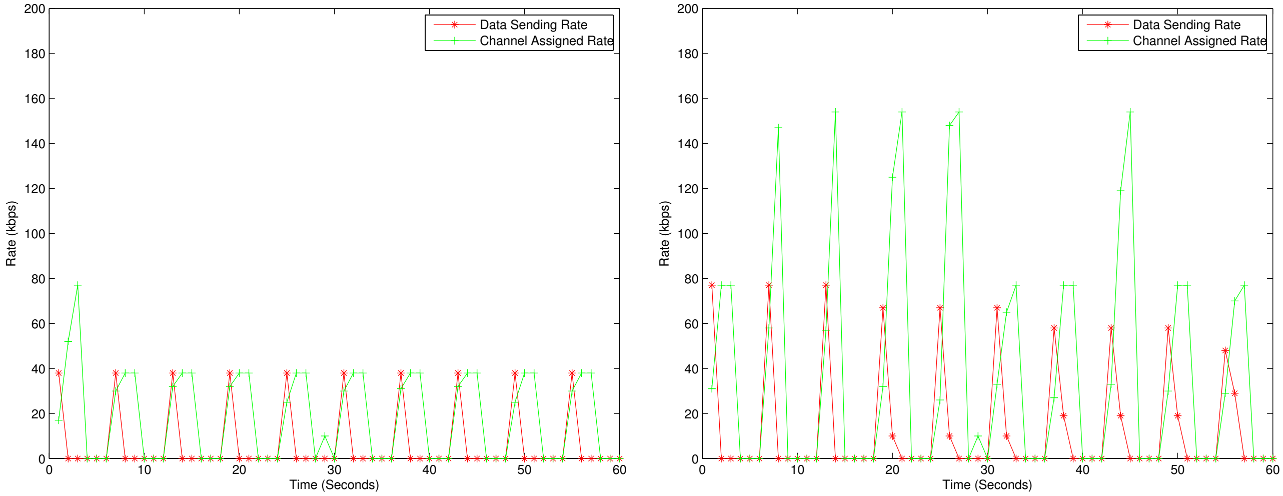


Fig. 4. Average Throughput for UDP CBR Experiments

Turning next to shorter time scales, in Fig. 5 we plot the time series of both the assigned channel rate and the data sending rate for two on-off experiments that had on-durations of 1 second and off-durations of 5 seconds. It clearly shows that the channel rate scheduler is highly sensitive to the data rate even over short time scales since it is able to assign the appropriate rates for every burst of data transmitted. This indicates that the scheduler checks the user's buffer occupancy at time scales less than 1 second before assigning an appropriate rate.



(a) On-off 1s/5s 38.4kbps

(b) On-off 1s/5s 76.8 kbps

Fig. 5. Wireless Scheduler Behavior at Short Time Scales

To further illustrate the strong dependency of the assigned channel rate on buffer occupancy as well as showcase the operation of NMI, we plot the NMI values between the two time series in Fig. 6 for various on-off experiments at different time shifts[8]. The figure shows several interesting features. Experiments that had low peak rates or short burst (on) periods show high NMI peaks (0.2 and higher) indicating that the scheduler assignment and the data sending rate are strongly correlated, i.e., the wireless scheduler is able to closely track the sending rate. Furthermore, the NMI values have multiple peaks at more than one characteristic delay since the data rate (and hence the channel rate also) are periodic signals, as can be seen in Fig. 5.

The NMI correlation also reflects the impact of the scheduler's fairness mechanism. Bursts with high duty cycles and high on-rates cause the wireless scheduler to deny resource allocation in which case the channel assigned rate stops tracking the data source resulting in a drop in correlation. As an example, the experiment with an on-rate of 115.2 kbps but a high duty cycle of $4/6$ yields very low NMI values compared to the experiment with the same on-rate but a lower duty cycle of $2/8$ which results in high NMI peaks.

Another interesting observation we make regarding the scheduler is that due to the small discrete set of supplemental channel rates, the rate scheduler may assign a much higher rate than the one requested, as shown in Fig. 5(b), which could have implications on the stability of the system.

2) Dependence on Channel Quality: It is well known that channel conditions can introduce significant signal distortion. However modern technologies like CDMA2000 incorporate techniques like rate control (through adaptive coding, modulation, Walsh Code length) as well as power control that allow them to either vary the rate or increase the power to adapt to channel conditions without sacrificing packet integrity. We wish to quantify the role of the former factor, i.e., adaptive changes in the channel rate assigned to the mobile in response to channel conditions, since it can directly affect higher layer protocols.

[8] Recall from Section IV-B, that when correlating time series we compute the NMI for different delays between the time series.

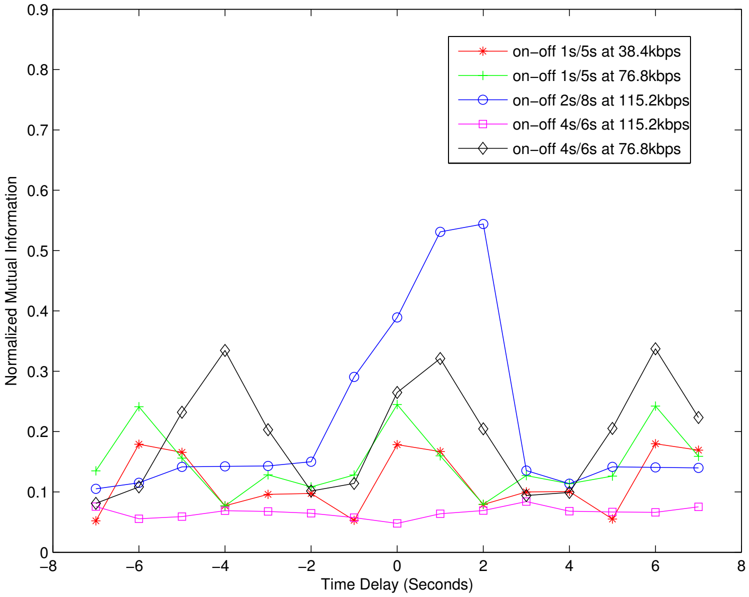


Fig. 6. Rate Scheduler's Sensitivity to Buffer Occupancy

As explained in Section II, the channel quality in CDMA networks is estimated using the metric E_c/I_o, where E_c is the pilot strength, and I_o, the overall interference. This metric was logged in our experiments by CAIT at a granularity of 1 second. Figure 7 shows a sample E_c/I_o signal. In order to quantify the correlation between the assigned channel rate by the wireless scheduler and E_c/I_o, we utilized the UDP CBR experiments, since the data sending rate is constant and hence does not affect the channel rate.

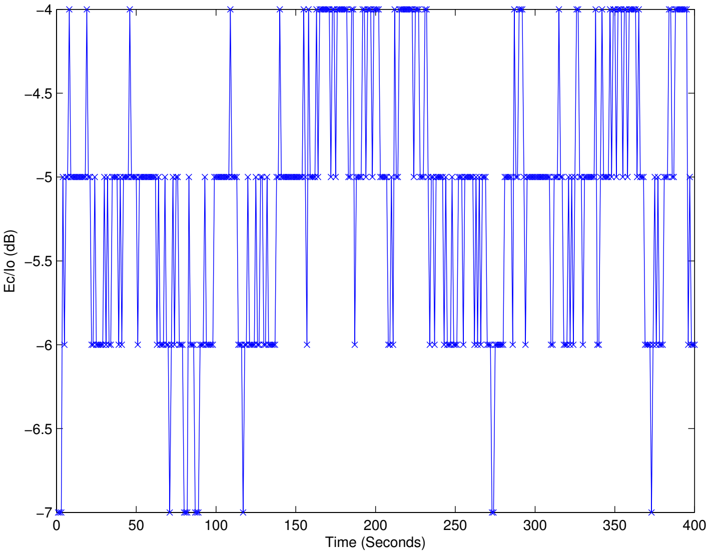


Fig. 7. Sample E_c/I_o Signal

Figure 8 shows the *maximum* NMI between the assigned channel rate and E_c/I_o for *all* the CBR experiments that we conducted. For each experiment, the maximum NMI was obtained over all time shifts. One can see that compared to the NMI values obtained when quantifying the impact of the data sending rate, the NMI of E_c/I_o is much smaller (by two orders of magnitude) across all time shifts. Hence, the empirical evidence indicates that in our experiments, the channel condition did not have a significant impact on the assigned channel rate. While the lack of correlation between channel conditions and the assigned rate may be surprising, we believe that this is because of the availability of sufficient sector power, which allows the CDMA network to temporarily boost the strength of the signal to combat adverse channel conditions. In other words, the network adapts to channel conditions via power control rather than explicit rate control.

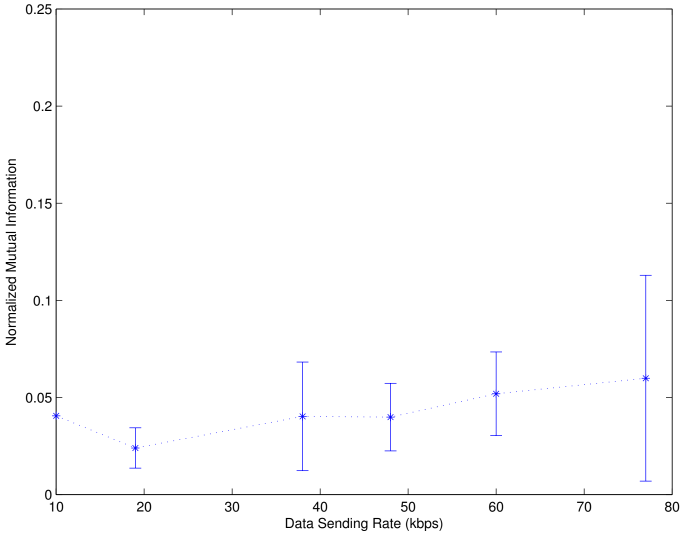


Fig. 8. Impact of Channel Conditions on Wireless Scheduler

3) Dependence on Sector Load: The last factor that we analyze is the sector load. The sector load time series represents the number of active voice and data calls that originate and/or terminate in the same sector as our client. We also computed the *maximum* NMI value between the sector load time series and the assigned channel rate for all the CBR experiments. Figure 9 shows the *maximum* NMI value between the sector load time series and the assigned channel rate for all the CBR experiments. As with the channel quality, the sector load does not have a significant impact on the assigned channel rate either. In most cases, this happened due to low sector load conditions. We also hypothesize that another contributing factor could be the small number of concurrent active data sessions and consequently little cross-traffic from other data users.

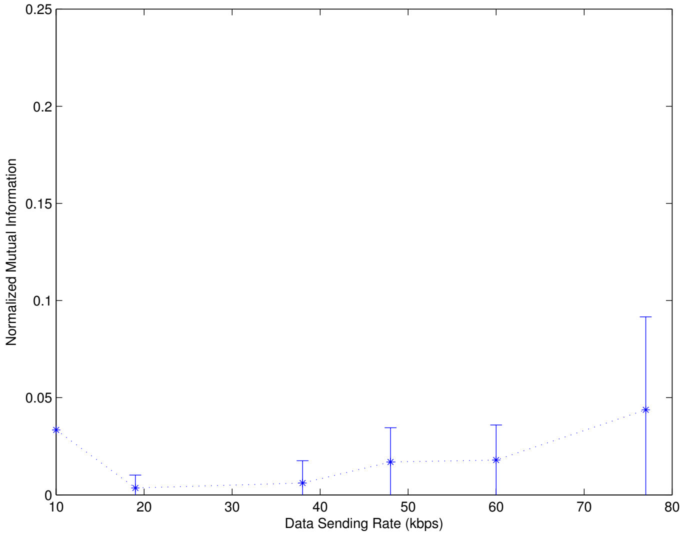


Fig. 9. Impact of Sector Load on Wireless Scheduler

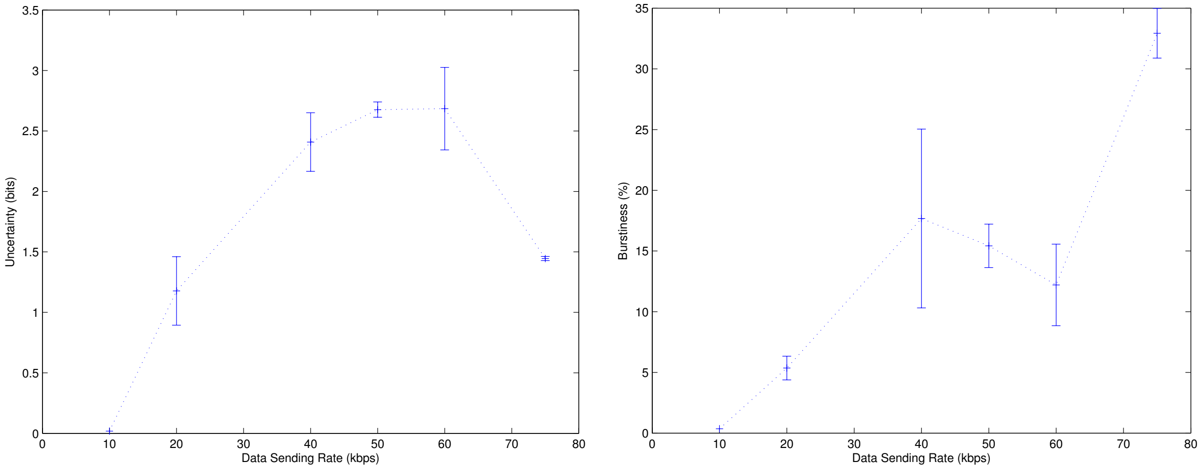
The set of experiments above clearly show the dominant influence of the data sending rate on the wireless scheduler. We further explore characteristics of the wireless scheduler as a function of the higher layer traffic pattern since they will be useful in analyzing the impact of the scheduler on TCP.

4) General Behavior: Intuitively, when characterizing the rates assigned by the scheduler, we are interested in how *bursty* the scheduler is, for example, whether or not it oscillates rapidly between different rates, and if so, how many *different* rates it cycles through.

To answer these questions, we introduce two metrics: the *burstiness* (denoted by β) and *entropy* (denoted by μ) of the assigned channel rates[9]. In order to compute β, we utilize the wavelet decomposition of the original signal. Specifically, the burstiness is defined as the ratio of the average energy in the fast-varying component to the average energy in the slow-varying component of the original signal. This captures the magnitude of fast variations in the channel rate *relative* to the average rate.

An increase in μ implies an increase in the number of rates being allocated and an increase in β implies an increase in the *rate* of variations. Hence, we can think of μ and β as measures of the scheduler's stability. If the rate scheduler was tracking the CBR data source perfectly, then both β and μ would be close to zero.

We plot μ and β as functions of the data sending rate for the various UDP CBR experiments in Fig. 10. Figure 10(b) clearly shows that as the data sending rate increases, the burstiness in the assigned channel rate generally increases. Figure 10(a) shows that the entropy of the channel initially increases with the sending rate, till the rate-limiting mechanism kicks in at high data rates. Beyond that point, the allocated rate frequently drops to zero, which results in a drop in the entropy of the rates assigned by the wireless scheduler.



(a) Channel Entropy

(b) Channel Burstiness

Fig. 10. Properties of the Wireless Scheduler

To further illustrate the scheduler's behavior, we plot the time series of the assigned channel rate for some representative UDP CBR experiments in Fig. 11. They clearly highlight the increase in burstiness of the channel assigned rates as the data sending rate increases. We next discuss the rate-limiting mechanism as a potential source of this burstiness.

5) Fairness Mechanism: In the previous paragraph, we showed that as the user's data rate increases, the channel becomes bursty, i.e., it is unable to closely track the sending rate. Furthermore, from Fig. 4, we note that at high data rates, the throughput actually *drops* implying that the scheduler stops honoring these rate requests. These observations indicate the presence of some kind of "rate-limiting" mechanism in the wireless scheduler to potentially maintain fairness among all the connections being serviced.

[9] The channel assigned rate signal, which is a time-average of the FCH and SCH rates over 1 second bins, was discretized using a bin-width of 10kbps to compute the marginal probability distribution.

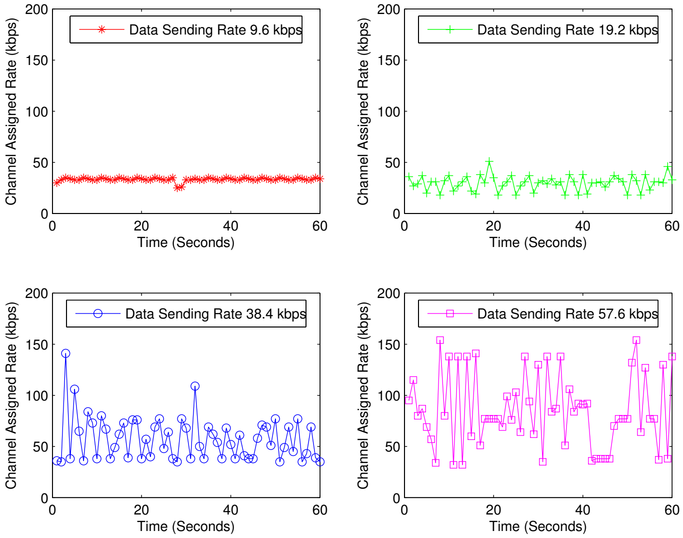


Fig. 11. Time Series of Assigned Channel Rate for UDP CBR Experiments

Although a precise inference of the mechanism is difficult to achieve solely through experiments, we highlight some specific features of its operation based on our observations. The scheduler periodically monitors the rates assigned to all connections. Connections that persistently request high channel rates (i.e., are continuously backlogged) are likely to be denied rate requests (or are assigned lower rates) for a period of time. Our experiments indicate that the likelihood of denial increases with the intensity of the previous assigned rate as well as the duration of the assignment. This was seen to occur over long time scales in Fig. 4 and Fig. 6 shows that it occurs over short time scales also. Specifically, for the experiments with a peak rate of 115.2 kbps, the wireless scheduler stops tracking the sender rate when the duty cycle increases from $2s/8s$ to $4s/6s$, even though the peak rate remains the same.

To further verify the existence of such a mechanism, we probed the channel using an on-off traffic source with a peak rate of 153.6kbps. The experiment was conducted at 6am on a weekend, to eliminate the effect of sector load (3 users on average), and from a location that is geographically close to the BTS to guarantee a good channel (E_c/I_o between -3dB and -4dB all the time). Fig. 12 clearly shows that after a duration of around 40 seconds, the wireless scheduler stops tracking the user's sending rate and the assigned rate drops to zero.

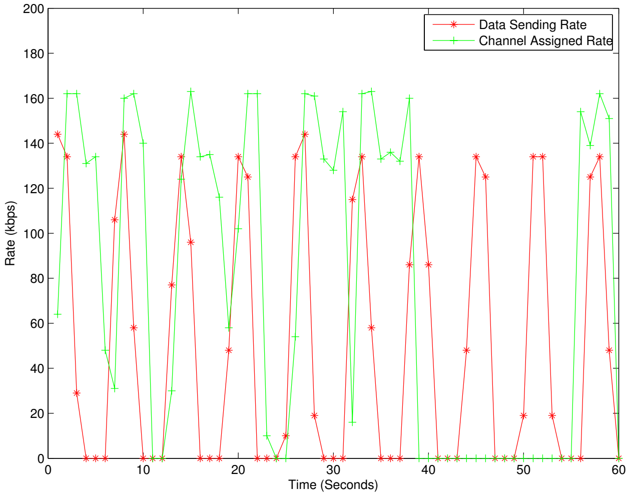
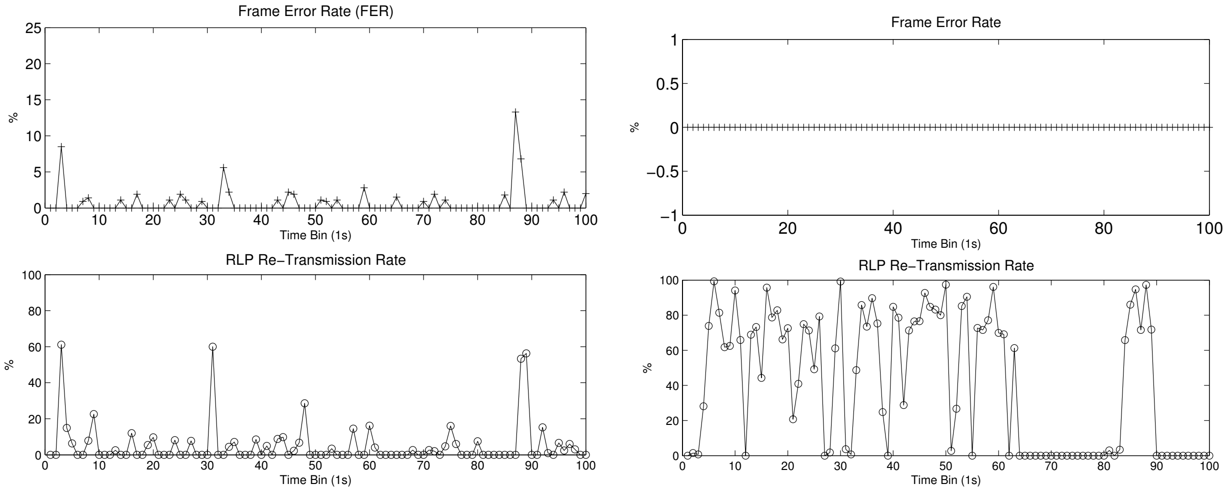


Fig. 12. Channel Scheduler Tracking an on-off Data Source with Peak Rate of 153kbps

B. Radio Link Protocol

The Radio Link Protocol is designed for fast recovery of link losses in wireless networks. Traditionally, these losses have been assumed to primarily arise from channel errors that can corrupt the radio frames. Hence, one can expect that the RLP re-transmission rate is highly correlated with the *Frame Error Rate* (FER). To verify this aspect, we analyzed the correlation between the RLP re-transmission rate and the FER. In most of our experiments, the FER was typically very low (zero) indicating strong error-correction and accurate power control. In experiments where there were instances of high FER on the channel, we did find a corresponding increase in the RLP re-transmission rate. However, quite surprisingly, we also observed several experiments where even though the FER was at or near zero, there were significant RLP re-transmissions. In Fig. 13 we plot the FER and RLP re-transmission rate from two experiments that highlight both scenarios. Figure 13(a), for a low rate UDP experiment, shows that spikes in the FER (upper graph) result in corresponding jumps in the re-transmission rate (lower graph). However Fig. 13(b), for a high rate experiment (153.6 kbps) shows that even in the absence of frame errors, the RLP re-transmission rate is often very high.



(a) UDP CBR 38.4kbps: Retransmissions due to FER

(b) UDP CBR 153.6kbps : Retransmissions due to losses

Fig. 13. Sources of RLP Re-transmissions

The presence of significant RLP re-transmissions, even in the absence of FER suggests that, apart from wireless channel errors, packet losses may be occurring in the back-haul network between the BSC and BTS *after* IP packets are converted into RLP frames, potentially due to congestion from cross-traffic.

C. UDP Connections: RTT, Packet Loss, and Reordering

Before analyzing the interaction between the wireless channel and TCP in the next section, we give some preliminary insight into higher layer metrics important for TCP performance, such as RTT, packet loss, and packet reordering that were experienced by the *UDP connections*. This will aid in our discussions regarding TCP in Section VI.

The client in our UDP application responded to received packets with acknowledgments (the server's transmission rate is not influenced by this). The data packets and acknowledgments had sequence numbers that allowed us to compute the RTT, identify the packets that were lost and infer any potential reordering of packets.

1) Round Trip Time: In Fig. 14 we plot the Cumulative Distribution Function (CDF) of the RTT for the different data sending rates. The RTT increases by a factor of 8 as the data sending rate increases from 19.2kbps to 76.8kbps which indicates the existence of a large buffer at the BSC.

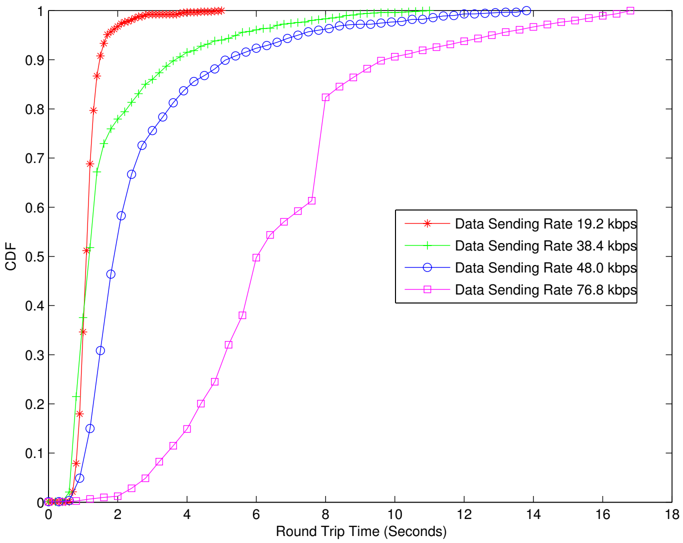


Fig. 14. CDF of Round Trip Time for UDP CBR Experiments

2) Packet Loss: In Fig. 15 we plot the CDF of the fraction of transmitted packets lost per second for a few different data sending rates. As the data source increases its sending rate the fraction of packets lost per second increases significantly. This indicates that the channel rate scheduler is unable to support high data rates. More importantly, it also indicates that packet losses in the cellular network are due to congestion rather than wireless losses. The exact cut off region, of about 50kbps, when the scheduler's rate-limiting mechanism kicks in is shown in Fig. 4.

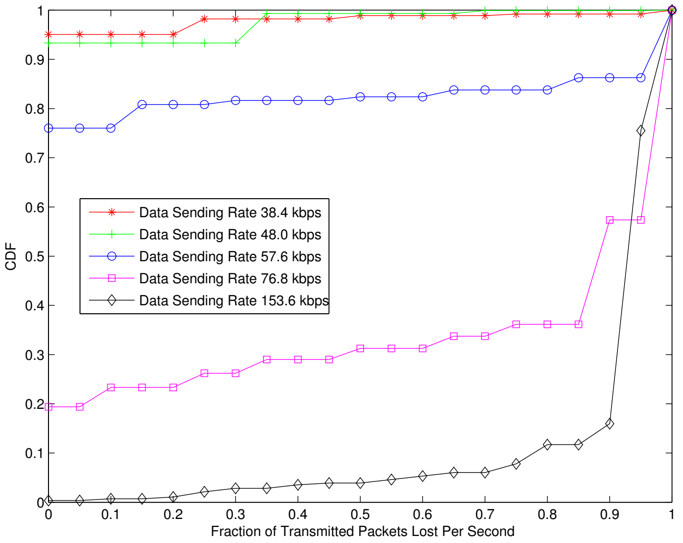


Fig. 15. CDF of the Fraction of Transmitted Packets Lost Per Second for UDP CBR Experiments

3) Packet Reordering: In general, packets can be reordered due to the traversal of a flow on multiple paths as a consequence of load balancing or RLP retransmissions. In practice, however, routers typically utilize per-destination load balancing to avoid packet reordering. As a result, packets belonging to the same flow[10] are not routed on different paths. The RLP retransmission mechanism can also reorder packets in the following fashion. Assume packet p_1 was transmitted before p_2. In the cellular network, both are

[10] A flow is typically identified using the source and destination IP addresses where every unique pair of addresses constitutes a flow.

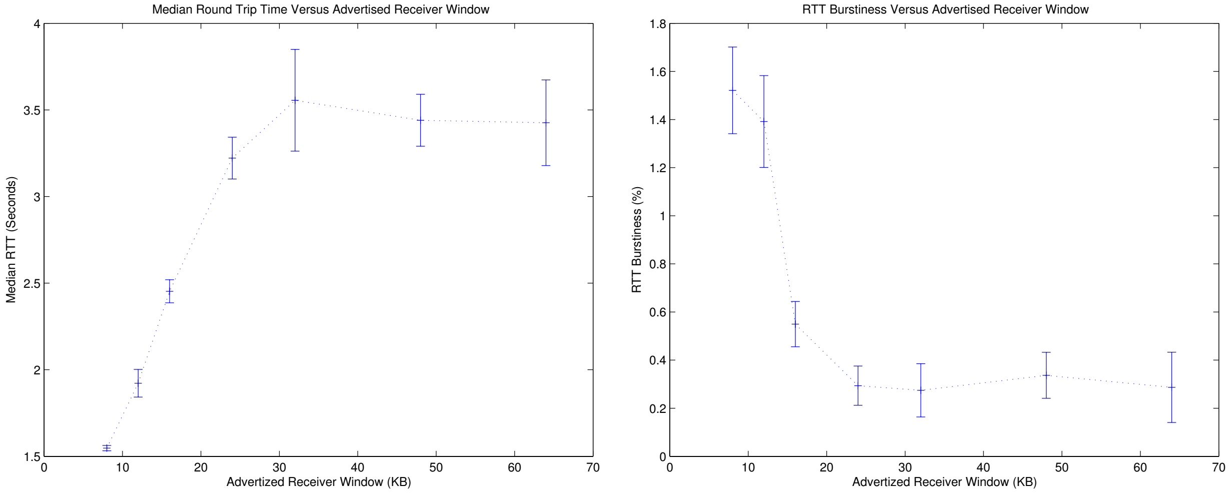
converted into radio frames before transmission on the wireless channel. Assume they are transmitted back to back. It may then happen that some frames belonging to packet p_1 are delayed due to channel/congestion loss causing repeated retransmissions. In such a scenario, packet p_2 would be reconstructed first and sent up to the higher layer before p_1. Analysis of *all* our UDP CBR experiments shows that packets were never reordered in our network.

VI. TCP AND RF CHANNEL COUPLING

In this section, we study the interaction of TCP with the two RF mechanisms: the wireless scheduler and the Radio Link Protocol (RLP). The TCP metrics we considered were the round trip time (RTT), packet loss, instantaneous (cumulative) throughput and timeouts. In our experiments, we varied the size of the advertised receiver window (*arwnd*) in an attempt to control the amount of interaction between the TCP controller and the channel rate scheduler. In general, when *arwnd* limits the growth of the congestion window (*cwnd*), the TCP controller's role diminishes to maintaining a fixed number of packets in the network. Thus in the absence of packet losses (which we argue is a reasonable assumption for small *arwnd*'s) we expect the TCP controller to be less influenced by the wireless scheduler's behavior. We show that the interaction between the two controllers is maximized when the *cwnd* is not limited by *arwnd* and TCP's congestion control mechanism is invoked in response to packet losses. In this case, we observe an unpredictable system behavior in terms of attainable throughput.

A. Round Trip Time

We infer the instantaneous RTT observed by the sender from the tcpdump log. Every packet that is transmitted is timestamped. For every explicit (cumulative) acknowledgement or sack that is received for a transmitted packet, we compute the RTT. We only compute RTT estimates for packets that were sent out once to avoid ack ambiguity. If duplicate acknowledgements are received we only consider the first one. Each RTT estimate is timestamped with the time at which the packet was *sent*.



(a) Median RTT

(b) RTT Burstiness

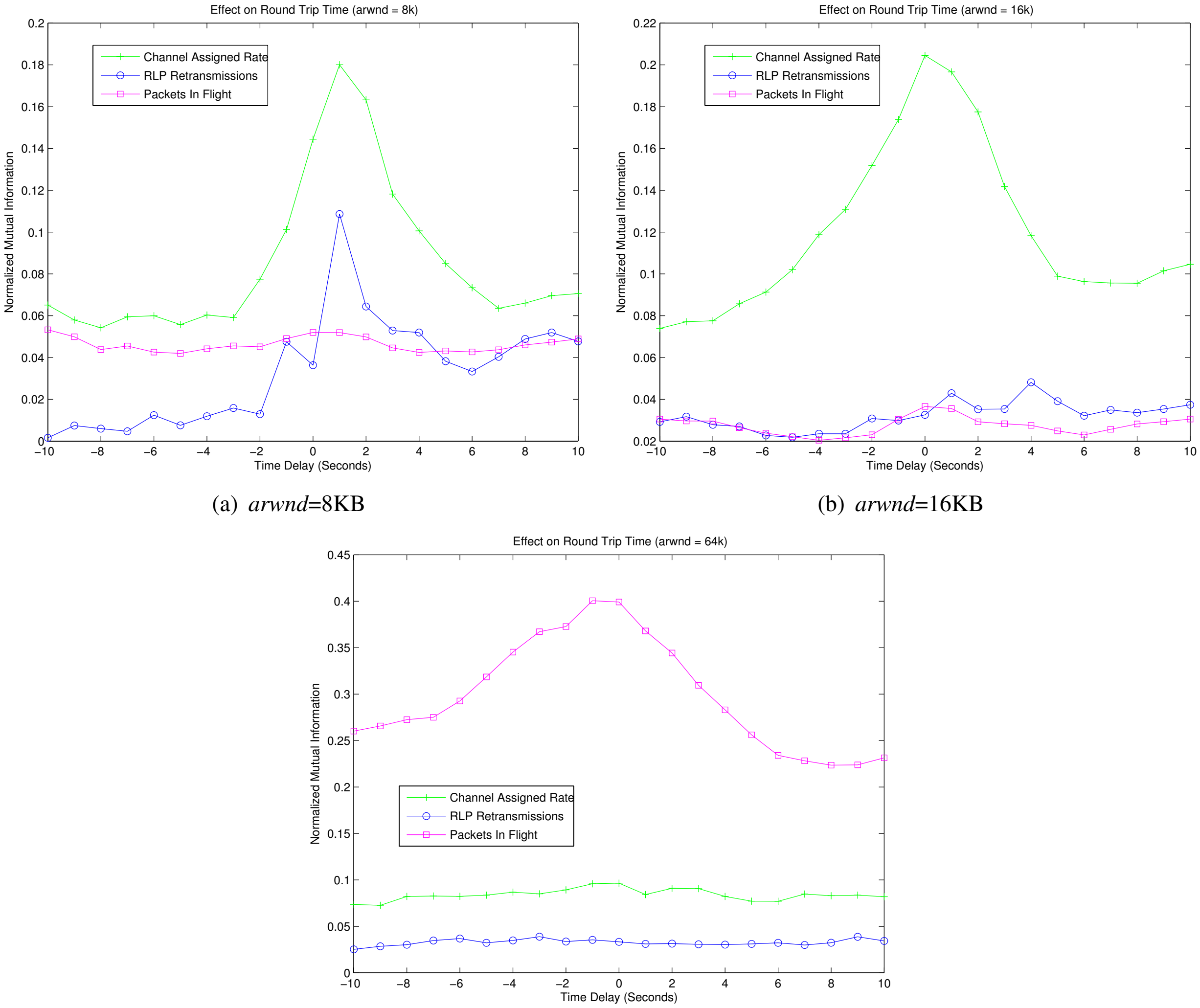
Fig. 16. RTT Behavior as a function of Receiver Window Size

In Fig. 16 we plot the median and burstiness of the RTT as a function of *arwnd*. The burstiness was computed in the same fashion as in Section V-A.4. The values reported are in the 90% confidence intervals.

Fig. 16(a) shows that as *arwnd* increases, the RTT increases significantly, indicating high queuing delays. For comparison purposes, we note that the authors of [5] have shown that in the Internet, the correlation between the RTT and the amount of data in flight (indirectly the receiver's window size) is quite weak. This discrepancy is not entirely unexpected since wireless links have typically far less bandwidth.

Fig. 16(b) shows that the RTT burstiness, on the other hand, decreases as *arwnd* increases since the average RTT increases significantly causing the *relative* burstiness to decrease. This could potentially decrease in the number of spurious timeouts caused by inaccurate RTO estimates (since the RTT estimates are *relatively* smoother). We study the effect of *arwnd* on TCP timeouts in Section VI-D.

In order to quantify the relative impact of the RF factors on RTT, Fig. 17 plots the normalized mutual information (NMI) between the RTT, the assigned channel rate and the number of RLP retransmissions (which affects time spent at the link layer) time series. We also computed the NMI between RTT and the number of packets in flight (which affects the queuing delay). The impact of the three factors was found to vary as a function of *arwnd*.



(a) *arwnd*=8KB

(b) *arwnd*=16KB

(c) *arwnd*=64KB

Fig. 17. Impact of RF Factors and Queuing on RTT

In general, we found that the number of RLP retransmissions has a limited effect on RTT even when *arwnd* is very small (less than 12KB). Fig. 17(a) shows one of the few connections with *arwnd* of 8KB where the number of RLP retransmissions had the highest effect on RTT. The dominant factor that impacts RTT for *arwnd* up to 32KB is the channel rate assigned by the wireless scheduler. Fig. 17(b) shows a sample TCP connection with *arwnd* of 16KB. When *arwnd* is increased beyond 32KB, the number of packets in flight becomes the dominant factor since the queuing delay increases significantly. Fig. 17(c) shows a sample TCP connection with *arwnd* of 64KB.

B. Packet Loss

The packets lost were identified by combining the information obtained from both the tcpdump log (collected at the sender) and the windump log (collected at the receiver). In general, if the network did not reorder packets (which is indeed the case) we can infer which packets were lost by keeping track of all the sent and received packets and marking the discrepancies between the two sequences of observations. A missing observation at the receiver implies that the packet was lost. To account for accidental packet reordering, we assume that a packet cannot be interleaved by more than three out of order packets. The number three was arbitrary yet sufficient based on our characterization of packet reordering in Section V-C.3.

In Fig. 18 we plot the average number of packets lost as a function of *arwnd* for *all* our TCP experiments. Our experiments indicate that the number of packets lost is insensitive to window sizes below 24KB but *increases* with the receiver window size above that threshold. This implies that the bottleneck buffer at the BSC is around this value. Furthermore, the small number of packets lost[11] indicates that the wireless channel is generally able to maintain a low probability of error to minimize wireless packet losses. As *arwnd* is increased beyond 24KB, however, congestion-induced losses become more prevalent.

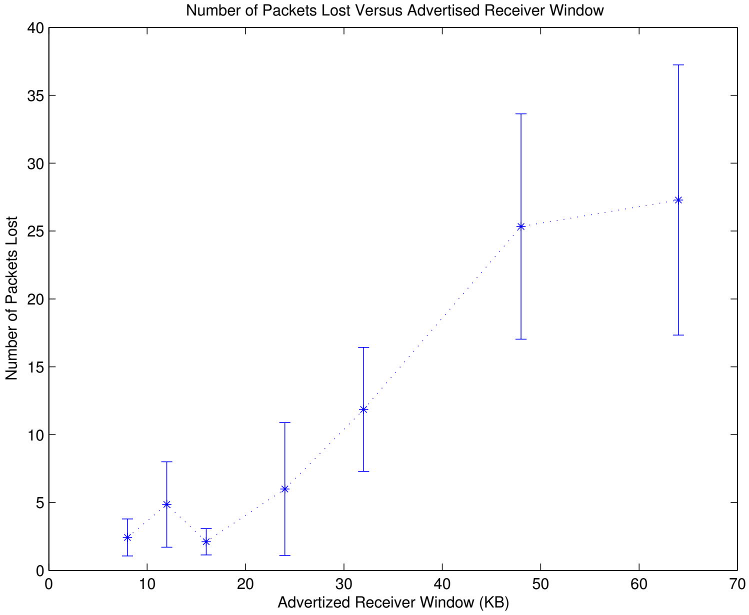


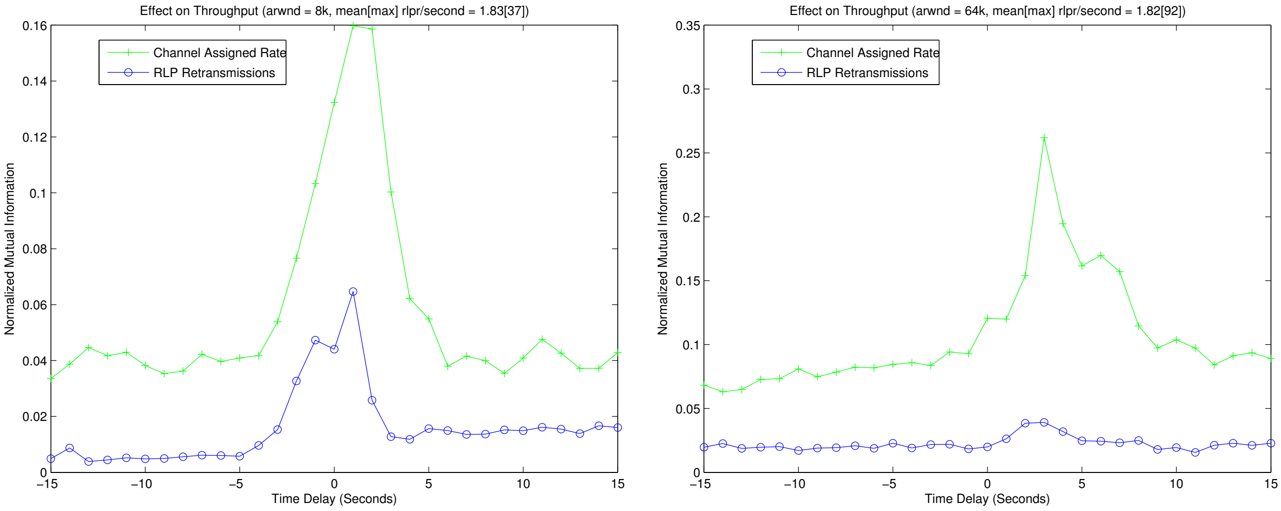
Fig. 18. Number of Packets Lost as a Function of *arwnd*

C. Throughput

As with other TCP metrics, we quantify the effect of the assigned channel rate and the number of RLP retransmissions on the instantaneous throughput. Fig. 19 plots the NMI between the two factors and the instantaneous throughput time series for two experiments with the smallest and largest window sizes. It clearly demonstrates that the assigned channel rate again has a larger impact on throughput compared to the number of RLP retransmissions. The number of RLP retransmissions has a noticeable effect only for small *arwnd*'s. Similar results were obtained for all our TCP connections.

Given the dominance of the wireless scheduler in influencing TCP's performance, we focus on several aspects of its interaction with TCP. The first aspect we investigate is how well TCP is able to track the available network capacity (i.e., channel assigned rates) over both long and short time scales. To this end, wavelet decomposition was performed on both the data sending rate and channel assigned rate time series to obtain their constituent low and high frequency signals.

[11] The TCP sender sends around 40,000 packets to transmit 5MB of data.



(a) *arwnd*=8KB

(b) *arwnd*=64KB

Fig. 19. Factors Affecting Instantaneous TCP Throughput

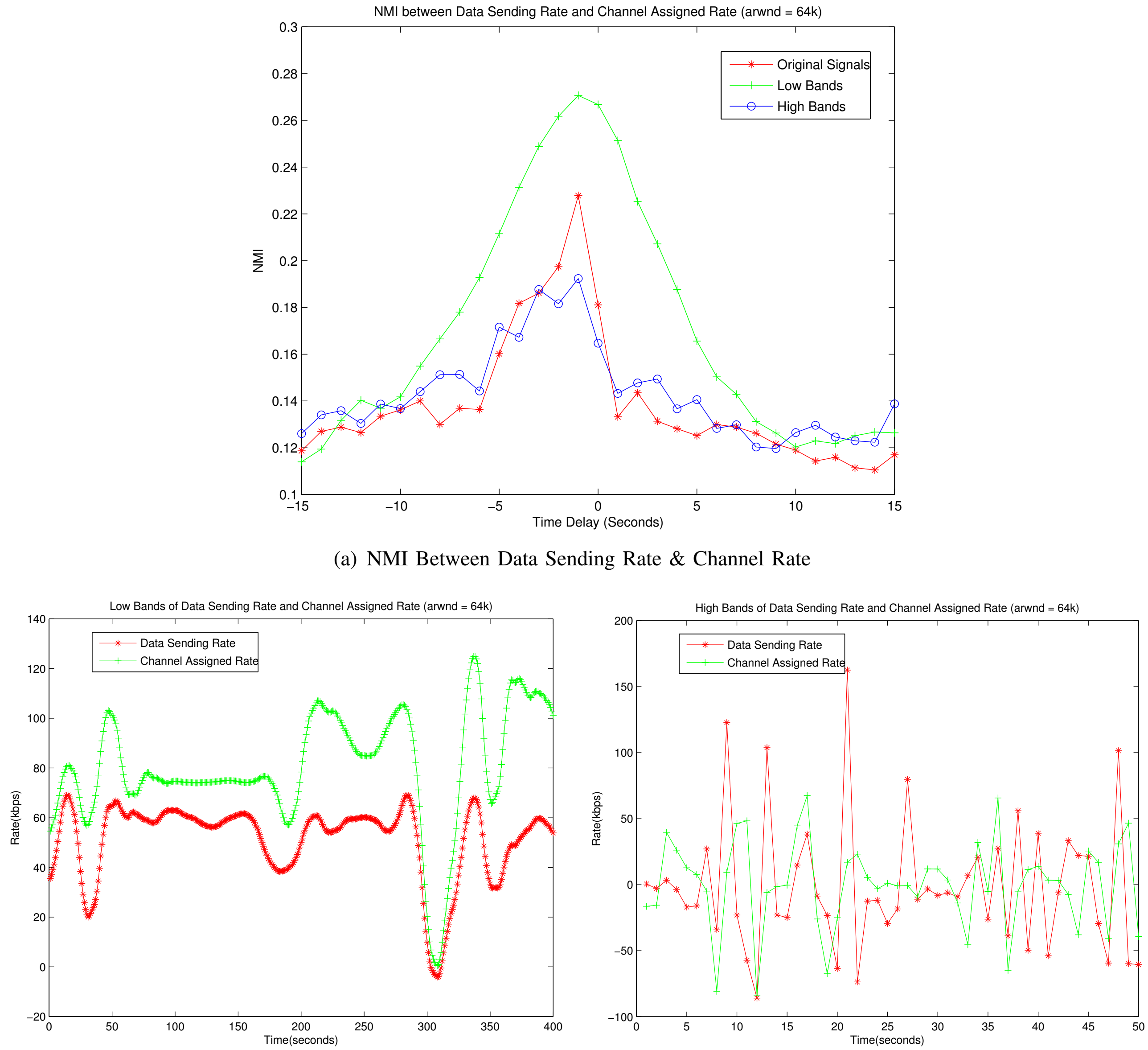
In Fig. 20, we plot the NMI between the high and low frequency signals and for completeness, the original signals as well. We also plot the low and high frequency signals in Figures 20(b) and 20(c) respectively. From the high NMI values it is clear, that the sender and the wireless scheduler are tightly coupled over both long and short timescales, with a stronger inter-dependence over long timescales. In other words, the rates assigned by the wireless scheduler are highly dependent on the data sending rate over long timescales, and vice versa.

We next turn to analyzing the impact of the wireless scheduler on TCP's *cumulative* throughput. It is well known that in wireline networks, TCP throughput initially increases with *arwnd* until the window size equals the bandwidth-delay product of the path. Any increase in *arwnd* beyond this size induces congestion, which drops the throughput to a lower fixed value, independent of how much further *arwnd* exceeds the bandwidth-delay product. Fig. VI-C plots the average throughput of TCP as a function of the *arwnd*. While we see a similar relationship between TCP's throughput and *arwnd* for the CDMA2000 channel, there are noteworthy differences. Specifically, the TCP throughput initially increases till *arwnd* reaches 24KB (which is slightly below the estimated 25K bottleneck buffer at the BSC). Increase in *arwnd* beyond that results in a drop in throughput as expected. However, unlike wireline networks, where the (lower) TCP throughput is insensitive to the amount by which *arwnd* exceeds the bandwidth-delay product, we observe that for the wireless channel, the *variability* in throughput increases and the mean cumulative throughput drops slightly. This can be attributed to the behavior of the wireless scheduler highlighted in Section V-A.4. The scheduler becomes increasingly bursty as the user's sending rate increases, which is the case for large receiver windows, and this causes an increased unpredictability in the throughput and also more chances of congestion. There does not seem to be any direct correlation, however, between the *mean* cumulative throughput and the burstiness of the channel.

D. Timeouts

In general, a timeout can be detected when a packet is retransmitted and *cwnd* drops to 1 segment[12] and TCP starts operating in the slow-start phase. We inferred timeouts from the tcpdump log collected at the server. In any TCP version (including the Sack Enabled Linux TCP [16] we are using), a packet is retransmitted either due to 1) the reception of 3 duplicate acks, 2) the reception of an ack including sacked blocks, or 3) timer expiration. In the first two cases the retransmission occurs *shortly* after receiving

[12] Some TCP implementations set the *cwnd* to 2 segments after a timeout is detected.



(a) NMI Between Data Sending Rate & Channel Rate

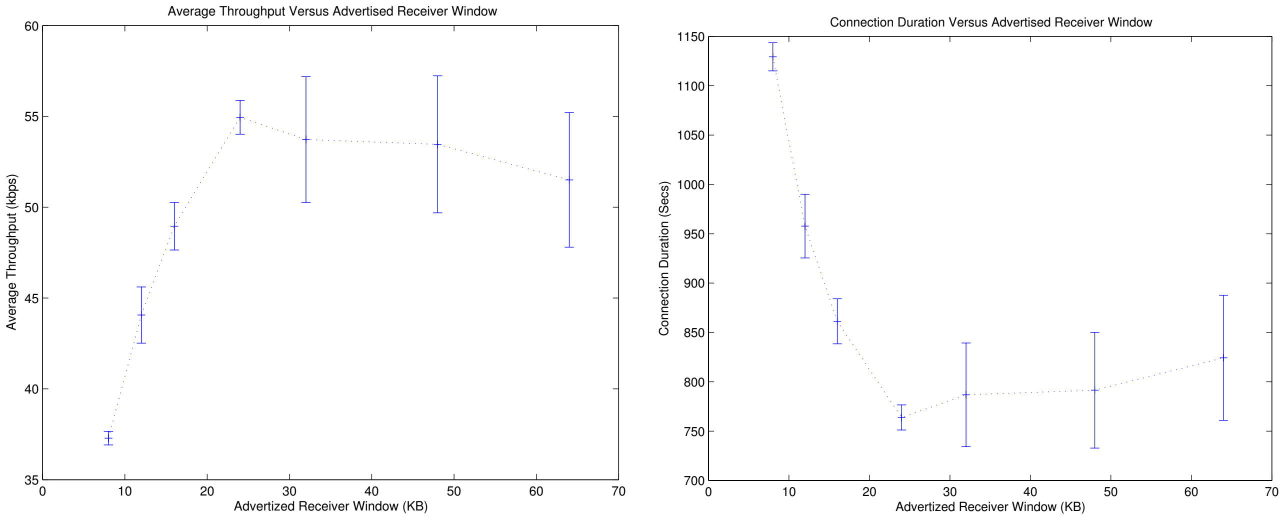
(b) Low Bands of Data Sending Rate & Channel Rate

(c) High Bands of Data Sending Rate & Channel Rate

Fig. 20. Coupling Between Sender & Channel Rate Scheduler

an ack from the client. In the third case, on the other hand, the retransmission *does not have to* occur after receiving an ack from the client. Our timeout inference algorithm is a threshold-based separation of cases 1 and 2 from 3. In Fig. 22 we show a histogram of the time delay between the occurrence of a retransmission and the reception of the last ack from the client across *all* our TCP experiments. The two bars represent the total number of retransmissions that occurred within *less* than 50ms and *more* than 500ms of the last ack received. There is a noticeable gap between the two bars (between 50ms and 500ms) where no retransmissions occurred. We therefore used a threshold of 100ms to distinguish between fast retransmissions and timeout-triggered retransmissions.

In Fig. 23 we show the number of instances a TCP connection timed out as a function of *arwnd*. Even though the RTT burstiness drops as *arwnd* increases, and one would expect the number of spurious timeouts to decrease, that is not generally the case in our network. The large number of timeouts for the connections with a small *arwnd* can be attributed to the RTT burstiness. For larger *arwnd*'s, however, the large RTT is potentially the main cause of timeout-triggered retransmissions. The BSC has a large buffer



(a) TCP Throughput versus *arwnd*

(b) Connection Duration versus *arwnd*

Fig. 21. TCP's Performance versus *arwnd*

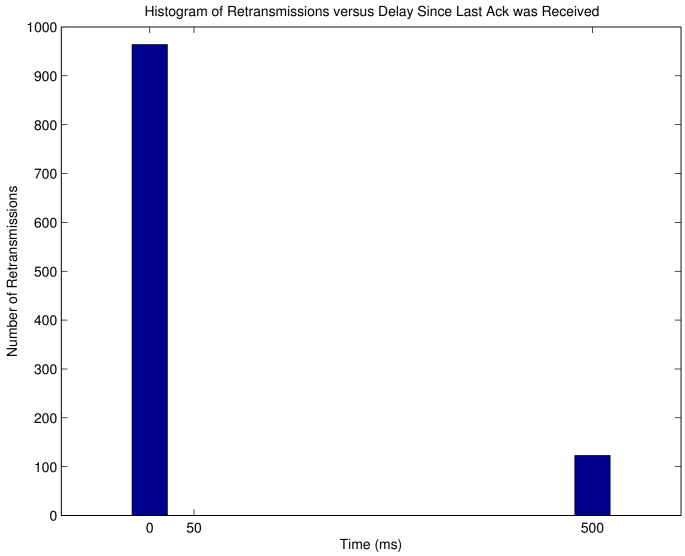


Fig. 22. Histogram of Retransmissions versus Time Delay since Last Ack was Received

to absorb the incoming packets and hide/mask the link layer optimizations from higher layer protocols and applications. This causes the TCP sender to suffer from a large feedback delay before it can infer that a packet loss has occurred. This delayed feedback could cause the TCP sender to timeout more often. It is interesting to note that all the connections with *arwnd* of 24KB consistently experienced the least number of timeout-triggered retransmissions. In fact, all our results (in terms of throughput, packet loss rate, and now timeouts) indicate that there is an ideal window size that TCP should not exceed.

In [8] the authors make several recommendations aimed at reducing the degradation in TCP's throughput when operating over wireless channels (specifically CDMA2000 channels) with varying bandwidth. Based on their experimental setup and specific TCP implementation, one of their recommendations was to increase TCP's receiver window size. In our experimental setup, however, especially with using TCP Linux [16], which is a widely used TCP version for servers, increasing the receiver window size beyond 32KB is not recommended. It is important to note that TCP Linux has a built-in mechanism that recovers from spurious timeouts whenever their occurrence is detected. TCP Linux also has a different mechanism

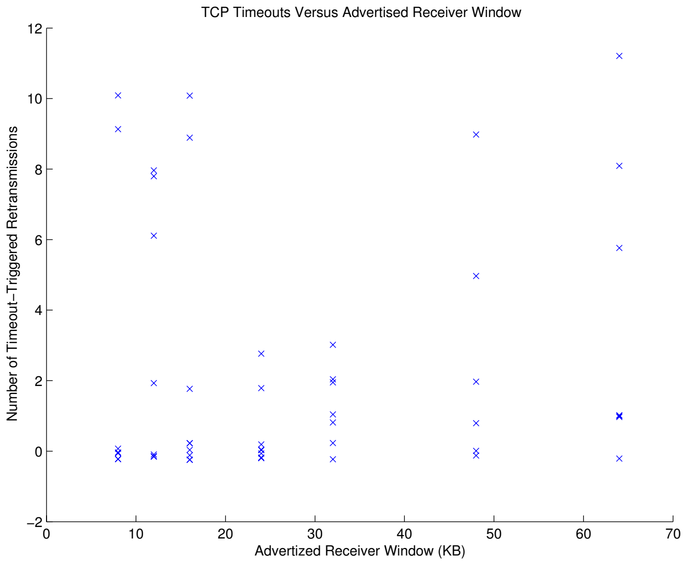


Fig. 23. Timeouts versus *arwnd*

for estimating the RTO, has support for SACK and Timestamp options, as well as ECN among other things, all meant to improve TCP's performance.

VII. Conclusions

We conducted a detailed characterization on the interaction between long TCP sessions on stationary devices and the wireless channel in commercial CDMA2000 networks. The study included several active experiments and collection of detailed RF layer statistics to characterize the behavior of the wireless channel. By applying a robust correlation measure and utilizing wavelet decomposition, we then studied the impact of the wireless channel on TCP metrics at different time scales. Our main findings can be summarized as follows:

1) Traditionally, link layer optimization of TCP performance on wireless channels has focused on the fast error recovery mechanism, *i.e.,*the RLP layer. Our study indicates that though the RLP layer is critical for good performance, changes in the RLP re-transmission rate do not have a significant impact on TCP dynamics. Analysis of the source of re-transmissions also show that CDMA2000 offers near lossless wireless links and packet losses are most likely due to congestion similar to that in traditional wireline networks.
2) Our work indicates that the adaptive rate allocation deployed by the wireless scheduler has the largest impact on TCP dynamics and we analyze this in detail. Furthermore, we show that the rates assigned by the wireless scheduler are *strongly correlated* to the TCP sending rate at all time scales. In other words, variations in the channel rate are no longer completely independent of TCP dynamics. We believe identification of such a system where both the scheduler and TCP influence each other's rates has the potential to aid design improvements.
3) We evaluated the impact of varying the TCP receiver window size, which identified the existence of a window size that maximizes throughput. Though this in itself is not surprising, the experiments also demonstrate that unlike in wireline networks, the fairness mechanism in the wireless scheduler can cause high variability in TCP throughput at large window sizes.

While our experiments were conducted on a CDMA2000 1xRTT network that utilizes power control, we believe that several of our findings are also valid in 3G networks like EV-DO. In particular the proportional fair scheduler in EV-DO utilizes the user's rate as one of its parameters, which indicates that our observations regarding strong correlation between the scheduler and TCP may hold for EV-DO also. Furthermore, given that EV-DO is expected to have better error correction codes etc. the impact of RLP variations should diminish even further. In the future, we hope to exploit the observation of such a dual controller system to introduce changes in the higher layer protocols that can leverage this information. In particular, the knowledge of the mechanics of the scheduler (which is eminently feasible for implementations like split-TCP[19]) may allow us to improve performance. We also hope to compare and contrast our findings for non-stationary mobile devices.

References

[1] Eitan Altman, Chadi Barakat, and Victor Manual Ramos Ramos. Analysis of AIMD protocols over paths with variable delay. In *Proc. IEEE INFOCOM*, Hong Kong, March 2004.

[2] Chadi Barakat and Eitan Altman. Bandwidth tradeoff between TCP and link-level FEC. *Computer Networks*, 39(5):133–150, 2002.

[3] Paul Barford and David Plonka. Characteristics of Network Traffic Flow Anomalies. In *Proc. ACM SIGCOMM IMW'01*, San Francisco, CA, USA, Nov. 2001.

[4] Dhiman Barman, Ibrahim Matta, Eitan Altman, and Rachid El Azouzi. TCP optimization through FEC, ARQ and transmission power trade offs. In *Proc. WWIC*, Frankfurt, Germany, February 2004.

[5] Saad Biaz and Nitin H. Vaidya. Is the Round-Trip Time Correlated with the Number of Packets in Flight? In *Proc. Internet Measurement Conference*, Miami, Florida, USA, Oct. 2003.

[6] Editor: C. Perkins. IP Mobility Support. INTERNET-DRAFT, `draft-ietf-mpls-rfc2002.txt`, October 1996. (work in progress).

[7] Mun Choon Chan and Ramachandran Ramjee. TCP/IP performance over 3G wireless links with rate and delay variation. In *Proc. ACM MOBICOM*, pages 71–82, 2002.

[8] E. Chaponniere, S. Kandukuri, and W. Hamdy. Effect of physical layer bandwidth variation on TCP performance in CDMA2000. In *Proc. IEEE VTC Spring*, volume 1, pages 336–342, April 2003.

[9] H. M. Chaskar, T. V. Lakshman, and U. Madhow. TCP over Wireless with link level error control: Analysis and design methodology. *IEEE/ACM Transactions on Networking*, 7:605–615, October 1999.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[11] Elena S. Dimitrova, John McGee, and Reinhard Laubenbacher. Discretization of time series data. Available at `http://polymath.vbi.vt.edu/discretization/`, 2005.

[12] Daniel Gillblad and Anders Holst. Dependency derivation in industrial process data. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 599–602, Washington, DC, USA, 2001. IEEE Computer Society.

[13] Benyuan Liu, Dennis L. Goeckel, and Don Towsley. TCP-cognizant adaptive forward error correction in wireless networks. In *Proc. IEEE Globecom*, Taipei, Taiwan, November 2002.

[14] QualComm. 1xEV: 1x EVolution, IS-856 TIA/EIA Standard. Available at `http://www.qualcomm.com/technology/1xev-do/whitepapers.html`.

[15] Qualcomm. CAIT CDMA Air Interface Tester. `http://www.cdmatech.com/download_library/pdf/CAIT.pdf`.

[16] P. Sarolahti and A. Kuznetsov. Congestion control in linux tcp. In *Proceedings of USENIX*, June 2002.

[17] TCPDUMP. Available at `www.tcpdump.org`.

[18] Vieri Vanghi, Aleksandar Damnjanovic, and Branimir Vojcic. *The cdma2000 System for Mobile Communications*. Prentice Hall, 2004.

[19] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley. Inference and Evaluation of Split-Connection Approaches in Cellular Data Networks. In *Proc. PAM (Passive and Active Measurement)*, Adelaide, Australia, March 2006.

[20] WindCatcher - Xceed Technologies. Available at `http://xceedtech.net/index_files/Page532.htm`.

[21] WinDUMP. Available at `www.wincap.org/windump`.