TECHNICAL RESEARCH REPORT

TCP Performance in Wireless Systems with Opportunistic Scheduling

by R. Srinivasan and J. S. Baras

CSHCN TR 2002-22
(ISR TR 2002-48)

TCP Performance in Wireless Systems with Opportunistic Scheduling

Roshni Srinivasan & John S. Baras
roshni@isr.umd.edu, baras@eng.umd.edu
University of Maryland

***Abstract*— Recent research on *opportunistic schedulers* for wireless data transmission has shown that link utilization is greatly improved by exploiting the independent variations in the wireless channels of users. The trade-off between spectral efficiency and fairness among users at the link level has been well studied in the literature. The focus of this paper is the impact of the *variable rate* and *delay jitter* induced by opportunistic scheduling on the congestion control and avoidance mechanisms of TCP. We use extensive simulations to expose the linkage between scheduling at the medium access (MAC) layer and TCP performance. The results in this paper underscore the need for careful design of opportunistic scheduling mechanisms in order to ensure that gains in MAC layer are also realized by TCP.**

***Index Terms*— Opportunistic Scheduling, Wireless, TCP, Delay Jitter, Variable Rate.**

I. INTRODUCTION

A. Motivation

Recent research on wireless data systems has shown that performance is strongly dependent on the schdeulers used in these systems. These schedulers operate in an especially challenging environment since they are required to optimally and efficiently share an output link (the airlink) whose bandwidth is constantly changing as the wireless channel fluctuates. We focus particular attention on the class of schedulers which fall into the broad category of "opportunistic schedulers." These are alternatively described as schedulers which exploit *multiuser diversity*. They effectively use channel information that is available at the base station to determine favorable scheduling instants for any particular user. While the overall capacity of the wireless link is seen to improve significantly with opportunistic scheduling, not much attention has been given to the rate variability and delay jitter that can adversely affect transport layer protocols such as the Transmission Control Protocol (TCP). In this paper, we study the effect of opportunistic scheduling policies at the Medium Access Control (MAC) layer on TCP performance.

B. Opportunistic Scheduling

Opportunistic schedulers represent a philosphical paradigm shift for wireless system designers who previously treated fading as an enemy to be combated to improve system throughput. Opportunistic schedulers aim to increase the aggregate cell capacity by *opportunistically* scheduling users at time instants when channel conditions are favorable. This concept is built on the information-theoretic foundations of multiuser diversity,

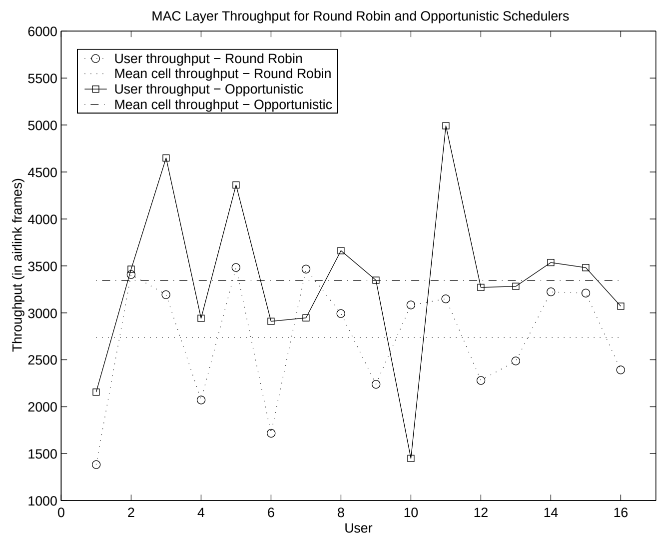

Fig. 1. A comparison of Round Robin and Opportunistic Schedulers for wireless users

first elucidated by Knopp and Humblet[12]. In a wireless system with a large number of users with independently fading channels, it is highly likely that at least one user experiences a highly favorable channel at any given time instant. A system which can track users' channel conditions constantly and make rapid scheduling decisions can therefore serve every user at the most favorable instant. The benefit to the aggregate system capacity from this technique grows with the number of users in the system. An opportunistic scheduler has actually been designed and implemented as the standard scheduler in the Qualcomm 1xEV-DO wireless data system, which has been standardized by the TIA/EIA as IS-856[7]. For descriptions of this scheduler, see [3][24].

An example serves well to illustrate this concept. Consider a base station serving 16 users in a time-slotted manner. A single user is served in every slot (the system model and assumptions are described in detail in Section III). We compared a round-robin scheduler with an opportunistic scheduler based on a simple "relative channel quality" metric (ratio of current signal-to-noise ratio to its average value). It may be observed from Figure 1 that with opportunistic scheduling, most users experience improved throughput resulting in an enhancement of about 20% in the aggregate system capacity.

However, one of the consequences of opportunistic scheduling is that scheduling epochs are no longer periodic and predictable for any particular user. To appreciate this subtle point, consider Figure 2 The topmost plot in the figure displays the channel variation for a particular user over a given time period. The middle plot indicates the scheduling epochs as well as the scheduled rate determined using an opportunistic scheduler. The corresponding plot for a round robin scheduling policy is displayed in the lowest plot in the figure.

The scheduled rate during any scheduling epoch is a function of the channel quality of the selected user at that epoch. As long as a user has data available for transmission, the round robin scheduler ensures that scheduling epochs are periodically spaced and that all users are given equal opportunities (number of transmission slots) to transmit data. At the same time, it is an interesting observation that the round robin scheduler, being channel-agnostic, often schedules the user when channel conditions are poor, including some instants when the SNR is less than the minimum required for data transmission (indicated by zero rate). On the other hand, the opportunistic scheduler picks time instants when channel conditions are favorable and schedules a higher rate at those epochs. This can be seen from the closely spaced bars of greater height at times corresponding to higher SNR. By serving the user at highly favorable time instants, it increases the throughput for this user (user 0) by more than 50%. An important point to note is that while the opportunistic scheduler improves aggregate throughput in the cell, it also introduces significant delay jitter in the scheduling instants. In this example, the opportunistic scheduler introduces upto 1 second of delay jitter for this particular user! If the scheduling policy focuses exclusively on MAC layer throughput without adequate consideration of jitter, the performance of network layer protocols such as TCP can be adversely impacted. The central focus of this paper is an investigation of the dynamics between opportunistic scheduling mechanisms at the MAC layer and TCP performance.

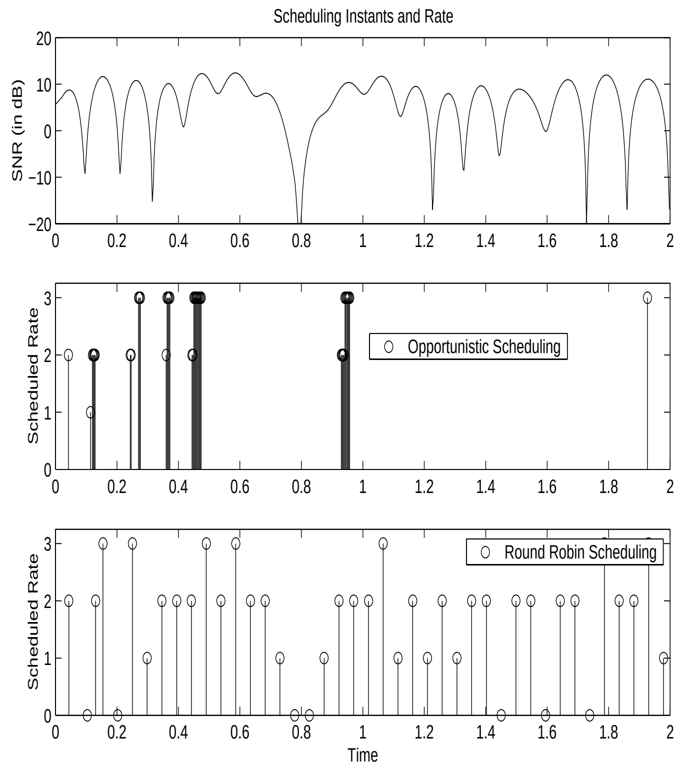

Fig. 2. Scheduling instants and rate for round robin and opportunistic scheduling.

II. Related Work

A. TCP - Congestion Control and Avoidance Mechanisms

TCP [20] [21] is the most widely used transport protocol for reliable data delivery in the internet. The TCP sender sends data in segments ordered by sequence number. The receiver acknowledges the receipt of segments through ACK packets that are cumulative: the ACK packet that indicates segment n as the *next expected segment* acknowledges the receipt of all data segments up to $(n-1)$. The sender is allowed to have a window of unacknowledged packets that varies dynamically in response to ACKs and detection of packet loss. Window evolution is governed by the slow start and congestion avoidance algorithms [21]. In the former, the window grows rapidly until the detection of a packet loss or until it reaches the slow start threshold (ssthresh) . The window growth then slows down when ssthresh is reached and the congestion avoidance phase begins. The window is incremented by one for every window's worth of packets, roughly by one segment per round trip time (RTT). Packet loss is detected at the sender either when multiple (typically three) duplicate acknowledgements (DUPACKs) are received or when no acknowledgement is received on the expiry of a timer (usually set to a weighted average of the RTT and its mean deviation), causing a retransmission timeout (RTO). TCP not only provides reliable delivery by retransmitting lost packets but it simultaneously reacts to congestion by reducing its transmission window and backing off its retransmission timer.

The sliding window flow control mechanism used by TCP is effective in probing the network for available bandwidth and reacting to delays and packet loss resulting from congestion in wireline networks. In such networks, TCP will operate in the congestion avoidance phase and the window will grow approximately linearly between packet drops. Ignoring the initial slow start phase, if $2R$ is the maximum rate attained at the end of a given congestion avoidance period, then it would drop to R at the start of the next congestion avoidance period, making the effective TCP throughput approximately $(3/2)R$ [4].

In wireless networks , however, the mobility of the end hosts and the variations in the quality of the wireless channel can cause packet losses and delays from sources other than network congestion. TCP reacts to such conditions by resetting its retransmission timer and initiating congestion control or congestion avoidance mechanisms. Bandwidth utilization my be throttled unnecessarily resulting in reduced throughput.

B. TCP over Wireless Networks - Early Approaches

A number of approaches to improve TCP performance in wireless networks without opportunstic scheduling have been suggested in the literature. In [4], the authors have analyzed the performance of a TCP connection over a time-varying wireless channel. In this work the authors consider the role played by the link-layer protocol, but the underlying assumption is one of a point-to-point link and not a cellular wireless scenario. The I-TCP protocol in [8] splits the TCP connection between the mobile and the fixed end host at the base station into two TCP connections. The connection over the wireless link can either be TCP or an enhanced wireless network protocol as in [26]. This approach requires significant software overhead from data traversing two additional TCP stacks and copying data at the base station. It also alters the end-to-end TCP semantics in order for the TCP sender to see improved throughput. In [2] the snoop protocol performs caching at the base station and local retransmissions across the wireless link. TCP performance is improved without altering the end-to-end semantics. Link Layer Error Recovery (LLER) is the basis for improved performance in all the above schemes. LLER that brings the packet loss probility down to 1% has now been incoporated in the standards in 3G1X-EVDO [27] in the US and its Asian/European counterpart, UMTS [25].

C. Scheduling in Wireless Networks

The roots of opportunistic scheduling lie in the information-theoretic results presented by Tse and Hanly in [23] and [6] which derive results for optimal power and rate allocation for users in a multiple access system operating over fading channels. Knopp and Humblet were the first to introduce the notion of multiuser diversity [12]. A team of Qualcomm engineers, in collaboration with David Tse, developed the Proportional Fair scheduling algorithm [24][3][11], which is implemented in the Qualcomm 1xEV-DO system. In recent work, Liu et al.[15] present algorithms for optimal opportunistic scheduling policies which satisfy different Quality of Service (QoS) requirements. Several other pieces of research on scheduling for wireless systems have appeared in the literature eg.,[16] and[18], but these are not "opportunistic."

Around the same time that the work in this paper was documented in detail as a technical report [22], a more general study of TCP/IP performance in 3G networks was published in [9]. The simulation results which are based on a TCP model with rate and delay variations being modeled by general distributions, e.g., uniform and exponential clearly show the degradation in TCP performance. In our paper we implement the Proportional Fair Scheduler used in the 3G 1X-EVDO (HDR) system and focus on the rate and scheduling jitter introduced by this scheduler . While our results are more system specific, they model the correlation between scheduling epochs in opportunistic scheduling more accurately than the distributions used in [9]. In our findings, we observe that TCP dynamics are strongly affected by the scheduling policy at the base station and simple design considerations can cause significant improvement in TCP performance.

III. System Model

We consider a cellular wireless scenario with a base station serving multiple mobile users. The cornerstone of the system under consideration is a time-slotted downlink combined with an asynchronous cicrcuit-switched uplink. We make this assumption for two reasons. Firstly, it simplifies the simulation setup and allows us to focus on the downlink as the bottleneck link. Secondly, most data applications are fundamentally asymmetric and very little data flows on the uplink. In fact, the system model assumed here is architecturally very similar to the Qualcomm 1xEV-DO data system and a number of assumptions outlined below are actually implemented features of this system.

- Every mobile user experiences a time-varying channel. We model this by a flat fading channel model where user i receives a signal $y_i(n)$ in the n^{th} time-slot given by:

$$y_i(n) = h_i(n)x(n) + w_i(n) \tag{1}$$

  where $h_i(n)$ is the time-varying channel attenuation and $w_i(n)$ is additive white Gaussian noise with variance σ_i^2. Assuming unit-energy signals, the nominal signal-to-noise ratio (SNR) for user i is:

$$C_{NOM,i} = \frac{1}{\sigma_i^2} \tag{2}$$

  The instantaneous SNR for this user, C_i is given by

$$C_i = \frac{h_i^2(n)}{\sigma_i^2} \tag{3}$$

- Every mobile constantly reports its measured channel quality, C_i to the base station through a dedicated control channel on the uplink. The rate requested by the mobile is a function of its channel quality. If the channel quality for the selected user is below the minimum required to support transmission ($C_i < C_{MIN}$), it requests zero rate.
- In every time-slot, the base station picks a single user, $\mathcal{U}_n$ based on the scheduling policy of choice. The rate that is scheduled for the selected user in this time slot, r_n, is the rate requested by the selected mobile for this time slot . Needless to say, any reasonable opportunistic scheduler will not select a user whose requested rate is zero. Note that practical considerations such as channel quality measurement errors, delay in channel report, etc. can be absorbed into a conservative choice of scheduled transmission rate.
- The MAC layer segments network layer packets into airlink segments, assumed to be of size B bytes. These segments are typically fairly small to enable efficient transmission across the airlink. Depending on the transmission rate determined on the basis of the channel

quality report, one or more of these airlink segments is transmitted in a single slot.

- Transmission in any downlink slot is assumed to fail with a certain probability, p_{err}.
- Transmissions on the uplink are assumed to be asynchronous and circuit switched, i.e., all mobiles can transmit packets at any time on the uplink. To make this assumption a realistic one, we only consider scenarios where uplink traffic consists exclusively of TCP ACKs.

A. Wireless Channel Model

We use the Jakes [10] channel model to simulate time-varying channels experienced by the mobile users. The Jakes model uses a sum of complex exponentials to approximate a Rayleigh fading channel. The complex channel gain at time t is given by

$$h_i(t) = \sum_{j=0}^{K-1} h_{i,j} exp(j2\pi f_d^i t cos(2\pi\phi_j)) \tag{4}$$

where $h_{i,j}, j = 0,..,K-1$ are complex unit variance gaussian random variables with zero mean representing the magnitudes of the subpaths. Each subpath has a phase delay, ϕ_j, which is uniformally distributed in $0, 2\pi$. The doppler frequency of the user is given by f_d^i. In this paper, we assume a flat fading channel, and hence we simulate a single Rayleigh fading path. The Jakes model produces a sequence of attenuation coefficients that is very close to a Rayleigh fading process, and in particular has the same correlation properties.

B. System Scheduler

In the rest of this paper, we focus on a particular flavor of opportunistic scheduler known as the Proportional Fair (PF) scheduler[24]. The metric used to select a user in any given scheduling epoch is

$$M_i(n) = \frac{log(C_i(n))}{T_i(n)}, \tag{5}$$

where $T_i(n)$ is an estimate of the user's average MAC layer throughput in some window of time prior to the current instant. The motivation behind this choice of scheduling metric is made clear by the following argument. Suppose each user were to be allowed to transmit all the time. In this case, this user's MAC layer throughput will be roughly proportional to $log(C_{nominal,i})$, which is roughly the average capacity of the link for that user. Since the scheduler tries to equalize any metric of choice, $M_i(n)$ will converge to a common value for all users over a long period of time. Convergence of the PF metric implies that every user gets an approximately equal number of transmission slots. The PF scheduler is fundamentally *resource fair*. The user's throughput is therefore roughly proportional to the average channel capacity of his or her link.

The PF scheduling metric provides an implicit mechanism to increase aggregate cell throughput (at the MAC layer) at the expense of increased delay jitter in scheduling any particular user. To see this, consider $T_i(n)$, the estimate of average user throughput. Suppose that this estimate is obtained by exponentially averaging the user's throughput, i.e.,

$$T_i(n) = \alpha R_i(n) + (1-\alpha)T_i(n-1) \tag{6}$$

where $R_i(n)$ is the rate for the user in the current slot and α is the exponential weighting parameter. a very value of α, which implies a very large averaging window, ensures that the estimate of the user's throughput dies down slowly after a scheduling epoch in which the user was selected, keeping the user's metric low for a longer duration. This allows the scheduler much more flexibility in picking highly favorable epochs in which the user can be scheduled. This in turn, leads to higher throughput for the user, and higher aggregate throughput in the cell at the expense of much higher scheduling jitter. At the other extreme, with a large value of α, the user is guaranteed not to have large delays between two scheduling epochs, and therefore low scheduling jitter. However, the large value of α causes the user's throughput to die down quickly after a scheduling epoch, therby increasing his metric and forcing the scheduler to select the user even when the user's channel conditions may not be extremely favorable.

While the PF scheduler can be biased to realize higher and higher MAC layer throughput, the scheduling delay jitter increases to a point where it starts to adversely affect TCP. This is because the effective service time, as seen by a packet in the queue, now has high variance. Queue buildup increases and packets are dropped. If the variance is very high, it can also lead to RTO, forcing TCP into slow-start which has a significant impact on throughput.

C. Simulation Setup

We use the *ns2*[17] network simulator for all of our experiments. We modified several features of *ns2* to realistically simulate a time-slotted wireless data system and we summarize these changes in this section. The base station (BS) node employs per-flow queueing. This is possible because of the relatively small number of users. Consequently, there are fewer flows which share the spectrum available in a single cell or a sector. Each queue is drop-tail with a maximum queue length of 10 packets. We have created a special queue class in *ns2* to enable the output link of the base station node to allow per-flow queueing. This queue class also contains a scheduler method which selects a single user to transmit to in every time-slot.

In the configuration shown in Figure 3, FTP applications at the sources generate data that travels through a network with a wireless link as its last hop. The base station node performs link-layer segmentation for efficient transmission over the airlink. Network layer packets are segmented into link-layer segments of 8 bytes. The base-station transmits one or more link-layer segments corresponding to a particular user over the airlink in every slot, each of which is a fixed duration of 1.667 ms. The number of segments transmitted

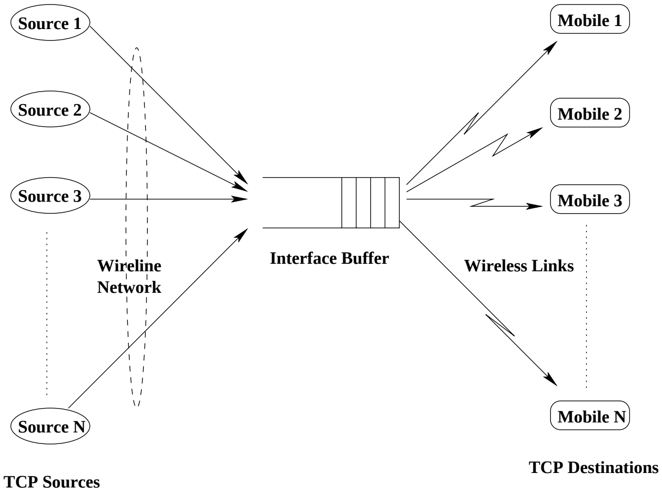

Fig. 3. Block Diagram of the Forward Data Path

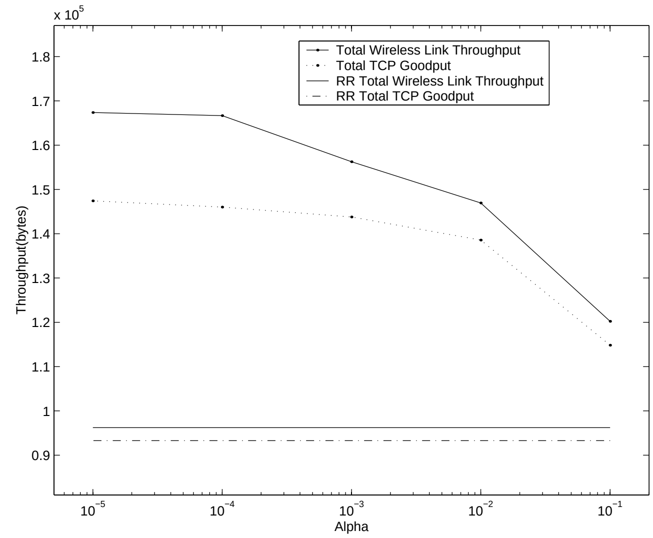

Fig. 4. Total MAC Layer Throughput vs Total Goodput of TCP connections

in a slot depends on the current SNR of the selected user. Table I lists the transmitted rate in link-layer segments as a function of the SNR. These rates and the size of the link-layer segments, 8 bytes, were chosen so that the maximum rate of 64 segments per slot corresponds to about 3Mbps if a single user is scheduled constantly. This peak rate is similar to the peak rates of 3G wireless systems such as 1xEV-DO as well as WCDMA. When all the link-layer segments corresponding to the packet at the head of the queue for a particular user have been transmitted over the airlink, the packet is deemed to be successfully transmitted. To simulate this, the packet is dequeued at this point and transferred to the node corresponding to the particular mobile user.

TABLE I

TRANSMISSION RATE PER SLOT AS A FUNCTION OF SNR

SNR (in dB)	Rate (Kb/s)
-12.5	38.4
-9.5	76.8
-6.5	153.6
-5.7	204.8
-4	307.2
-1.0	614.4
1.3	921.6
3.0	1228.8
7.2	1843.2
9.5	2457.6

We focus our attention on the base station node which sits at the head of the bottleneck link, which is the airlink. We assume that the data path from the source to the wireless link interface and the return path for the ACKs do not introduce loss or delay variations. The scheduling policy at the bottleneck link can be configured to be Round Robin or Opportunistic scheduling with configurable metrics. For opportunistic scheduling we have simulated the Proportional Fair scheduler that is used as the standard scheduler in the Qualcomm 1xEV-DO system.

IV. SIMULATION RESULTS

As discussed in Section III-B, the tradeoff between MAC layer throughput, and variable rate/delay jitter in opportunistic schedulers is the core issue of interest. To this aim, we conducted the following experiments. We consider a wireless cellular system as described in Section III with 16 users. This number is approximately the same as that which can be supported in typical cellular scenarios, for eg., in IS-95 CDMA systems. All 16 mobiles initiate FTP downloads over TCP connections within a window of 0.1s. For each value of α, the FTP sessions run for a duration of $T(\alpha)$ seconds, at the end of which the TCP sessions are torn down. The duration of each experiment is made dependent on α for the following reason. The PF scheduler is guaranteed to provide approximate resource fairness over time intervals that are at least $1/\alpha$, which holds closely for intervals that are several times that number. Hence, we choose $T(\alpha)$ to be at least a few orders of magnitude larger than the time constant $1/\alpha$. Extensive Monte Carlo simulations were performed and several statistics are collected for each user at the end of this exercise:

- mean, maximum and standard deviation of scheduling jitter at the base station for each user.
- mean, maximum and standard deviation of the queueing delay at the base station for each user.
- aggregate and user MAC layer throughput.
- total TCP throughput, TCP time sequence and congestion window evolution

Figure 4 displays the aggregate MAC layer throughput and the aggregate TCP goodput as a function of α, the exponential averaging parameter. The behavior of the MAC throughput is consistent with intuition. As α becomes smaller, the throughput increases as expected. With smaller α, the scheduler is operating with fewer constraints and is able to schedule users at highly favorable time instants. Observe that the MAC layer throughput saturates for very small values of α. This saturation occurs because of the negligible effect of the averaging time (which is related to $1/\alpha$) which is much

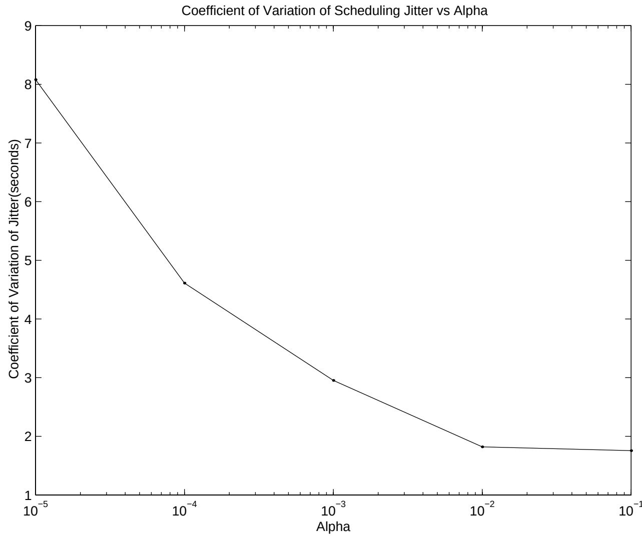

Fig. 5. Coefficient of Variation of Scheduling Jitter

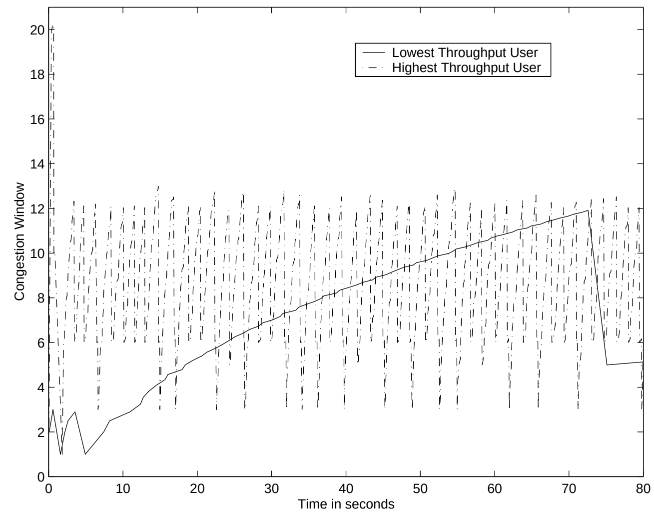

Fig. 6. Congestion Window Evolution for connections with Highest and Lowest TCP throughputs

larger than the maximum channel coherence time of the users. Now, while the mean TCP goodput displays approximately similar behavior, the true picture is a little more subtle. For large value of α, the TCP goodput tracks the MAC layer throughput fairly closely. However, as α gets smaller, the TCP goodput increases at a slower rate than the MAC layer throughput and the two curves diverge. The reason for this is that very small values of α imply larger delay jitter in the scheduler which in turn starts to affect TCP goodput. We also plot the MAC throughput and TCP goodput for a Round Robin (RR) scheduler with the same distribution on SNRs. The RR scheduler eliminates scheduling jitter since the scheduling epochs are equally spaced. As long as channel conditions support the minimum rate and TCP data is available, a flow is served at constantly spaced intervals. As in Figure 1, we observe that the opportunistic scheduler does better than the RR scheduler even for alpha as large as 0.1.

As one would expect, the scheduling jitter increases with decreasing α as the scheduler is allowed more flexibility in scheduling users. We obtain the mean and standard deviation of the scheduling jitter for each Monte Carlo simulation as we vary α. In Figure 5 we see a sharp increase in the coefficient of variation (mean/standard deviation) as alpha is decreased.

V. Analysis of TCP Performance

In order to understand the effect of jitter and rate variability on TCP, we superposed the scheduling epochs and rates on various plots. We used TCP Time Sequence graphs (TSG) that were generated using the *tcptrace* tool [19]. The TSG plots the activity of the connection by showing the evolution of the sequence number with time. The general slope of the graph indicates throughput of the connection. The jumps with arrowheads track the sent segments, with the lower and upper arrows coresponding to the first and last byte in the segment respectively. The solid line tracks the ACKS returned by the receiver. The letter R marks the time instants at which retransmissions take place. The number 3 marks the receipt of 3 duplicate acknowledgements. We analyzed the behavior of the connections with the best (User B) and worst (User W) throughputs for each value of α and describe the analysis for results corresponding to a randomly picked Monte Carlo iteration for $\alpha = 0.01$.

- ***Scheduled Resource Fraction***: User W was scheduled for 5.52% of the total scheduled slots and User B was scheduled for 7.02%. The mean resource fraction was 6.25% and the standard deviation was 0.44%. This roughly equal allocation of scheduled slots accross users is consistent with the resource fair nature of the Proportional Fair Scheduler.
- ***Nominal SNR and Requested Rate***: It must be noted that User B's nominal SNR is 12.2370dB while User W's nominal SNR is 0.0833dB. With a higher nominal SNR, User B requested an average rate of 56.2517 frames each time it was scheduled, the maximum schedulable rate being 64 frames per slot. User W on the other hand, requested an average rate of only 1.8851 frames per scheduled slot. Therefore, while the PF scheduler awards both users approximately equal scheduling slots, user B consistently transmits at a much higher rate in each scheduled slots.
- ***Congestion Window Evolution***:Fig. 6 shows the congestion window evolution for both users for the first 80s. Each connection has a buffer size of 10 packets and an initial slow start threshold, *ssthresh* of 20 packets. When the connection starts, User B, which is served with less jitter and higher rate, quickly ramps up to *ssthresh*. Thereafter it remains in cyclic congestion avoidance with each cycle being terminated by a buffer overflow, when the queue size exceeds 10 packets. User W on the other hand, being unable to support high data rates, suffers multiple drops at the beginning, repeatedly halving its slow start threshold. TCP packets from User W suffer

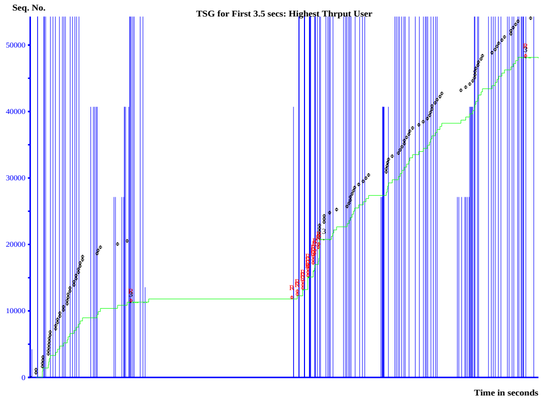

Fig. 7. Time-sequence graph for connection with highest TCP throughput

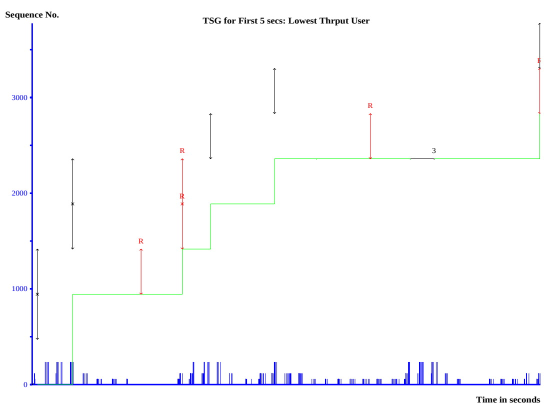

Fig. 8. Time-sequence graph for connection with lowest TCP throughput

long delays at the bottleneck link, increasing its RTT and causing it's window to grow very slowly. This can be seen by the gradual slope of the graph in the congestion avoidance phase.

- ***Time Sequence Graphs***: We observe from the TCP time sequence graphs that the throughput of User B (max. seq. no. 3885976) is more than 37 times that of User W (max. seq. no. 104312). User B's higher nominal SNR and consequently higher scheduled rate is largely responsible for this difference in throughput. On superposing the scheduling instants and scheduled rates (vertical bars) and zooming in on the first few seconds in in Figs. 7 and 8, we found that the large scheduling jitter and low rate had a significant impact on User W. After the exponential growth in the initial phase, User B quickly overflows its buffer. However, it responded well to the fast retransmit after the first instance of 3 duplicate acks and did not suffer from RTO's. The scheduler empties the buffer quickly and for a short period no data is available. Once the sender responds to the buffer overflow, the scheduler resumes service to User W and it ramps up quickly. It remains in cyclic congestion avoidance for the rest of the simulation with consistently high scheduled rate and low jitter. We observed that User W is scheduled less frequently and with far lower rate in the first 5 seconds. It suffers multiple RTO's in the initial portion, causing *ssthresh* to be reduced drastically. The sequence number for the same time interval clearly shows the difference in the throughputs for the two connections.
- ***Packet Scheduling Jitter***: The mean, standard deviation, minimum and maximum jitter for User W were 996.1ms, 342.9ms 30.9ms and 1927.5ms respectively. The corresponding values for User B were 23.6ms, 27.2, 0.058ms and 231ms respectively. User B's higher nominal SNR could have contributed to its metric being maximized for a larger fraction of the time, implying scheduling epochs that are closely spaced in time. User W on the other maximized the metric only when its throughput dropped significantly. This caused scheduling instants to be spaced further apart than for User B.
- ***Round Trip Time***:The mean, standard deviation, minimum and maximum RTT for User W were 5754.3ms, 1893.0ms, 514.9ms and 9629.9ms respectively. The correponding values for User B were 187.7ms, 81.1ms, 35.3ms and 448.4ms respectively. It can be seen that the mean jitter is 17.31% of the mean RTT for User W as compared to 12.57% for User B.The combination of low rate and high jitter adversely effect the RTT for User W. Successive RTO's in the beginning of the connection only worsen the problem.
- ***Queuing Delay at the Bottleneck Link***:The mean, standard deviation minimum and maximum delay for User W were 6785.5ms, 2108.4ms, 30.9ms and 9734.1ms resp. The correponding values for User B were 154ms, 83.3.0ms, 0.058ms and 432.2ms resp. Increasing the queue size to smooth out the effects of scheduling jitter and rate variations will only increase the queueing delay further.

VI. Variable Fairness Metrics

In principle, the proportional fair scheduler is designed to be *resource-fair* in a time-slotted system i.e., it ensures that all users, irrespective of channel conditions, are scheduled an equal fraction of the total number of time slots. Naturally, this does not optimize aggregate system throughput which can always be increased by reducing the number of time slots allocated to users with poor channel conditions and allocating them to the users with better channels who are capable of supporting higher data rates. This section examines the inherent tradeoff between aggregate system throughput and fairness.

As usual, let $R_k(t)$ be the requested data rate from mobile user k at time t. In an ideal system, this will be proportional to $log(1 + SNR_k(t))$, where $SNR_k(t)$ is the SNR experienced by the same mobile user at the scheduling instant t. Let $T_k(t)$ represent an exponentially smoothed throughput estimate for the same user. This quantity is a reflection of the service received by the user in some window of past time slots. Consider the metric where fairness varies as

$$B_k^{(\lambda)}(t) = log(R_k(t)) - \lambda log(T_k(t)) \tag{7}$$

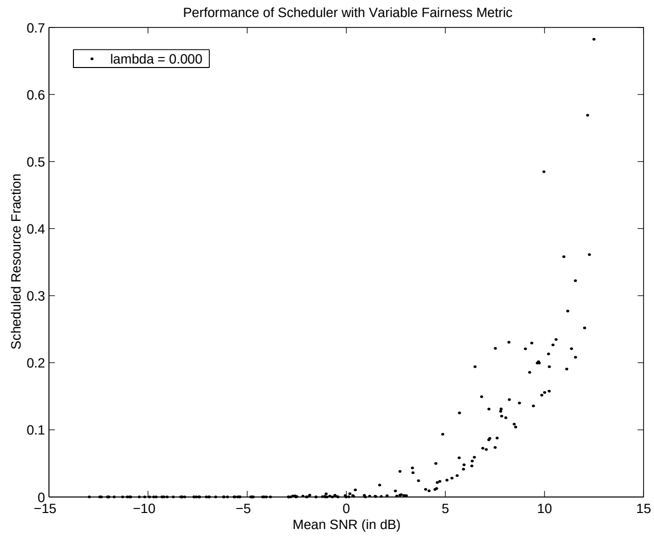

Fig. 9. Maximum SNR Scheduler

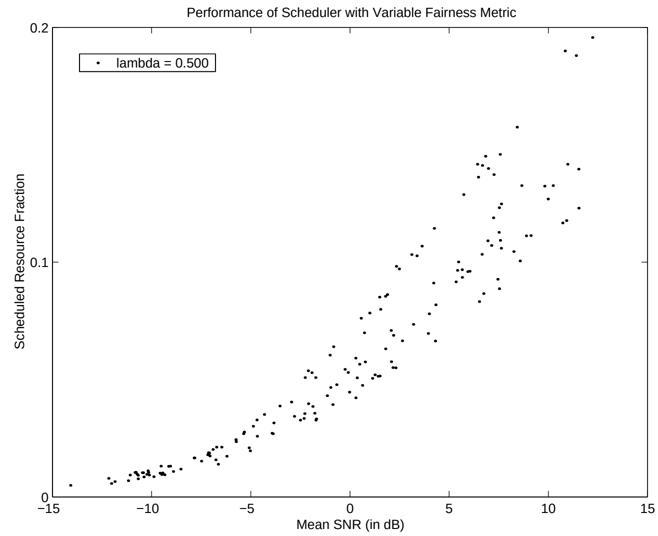

Fig. 11. Trade-off between Resource Fairness and Rate Fairness $\lambda = 0.5$

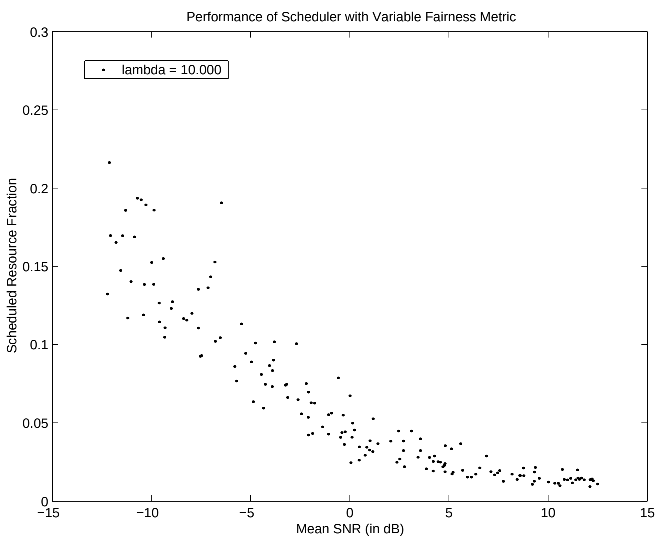

Fig. 10. Rate Fair Scheduler

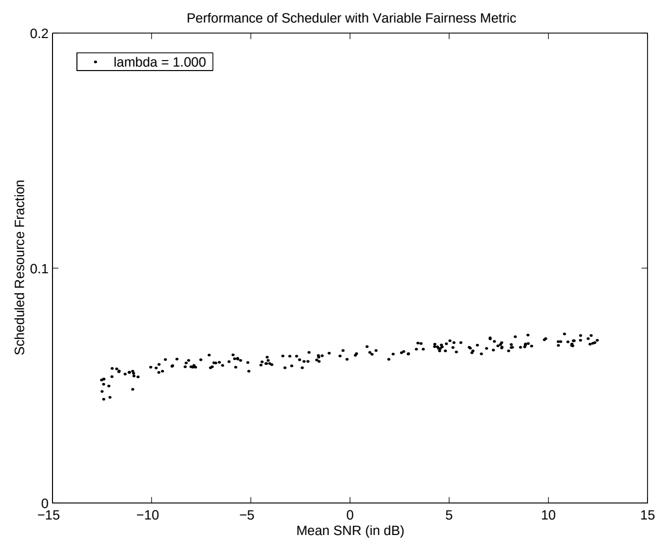

Fig. 12. Proportional Fair Scheduler - Resource Fair

The parameter λ can be chosen to satisfy various fairness criteria. We performed Monte Carlo simulations with the set up as described in Section III and Section III-B with the exponential throughput smoothing parameter set to 0.01. An extreme case of opportunistic scheduling is one in which $\lambda = 0$. This scheduler favors the user with the best channels and is completely unfair to users with poor channels. This is illustrated in Figure 9 which plots the scheduled fraction of time slots for users as a function of their nominal SNR. Users with higher SNRs get a larger share of the scheduled slots at the expense of the users with lower nominal SNRs.

On the other hand, an operator may choose to deploy a service that attempts to guarantee reasonable rates even for users at the edge of a cell with poor channel conditions by awarding a higher fraction of time slots to such users. This flavor of service (tending towards *rate-fair*) can be realized by using a higher value of λ. The effect of this is illustrated in Figure 10. This metric, with $\lambda = 10.0$ has the opposite effect to that with $\lambda = 0.0$

The aggregate system throughput can be optimized at the expense of fairness to weaker users by choosing a small value of λ. The trade-off between resource fairness and rate fairness is evident in Figure 11. With $\lambda = 0.5$ the scheduler allocates increased resources to weaker users at the expense of users with better channels.

When $\lambda = 1$, this is identical to the proportional fair metric, $\frac{R_k(t)}{T_k(t)}$. The resource fraction is seen to be roughly equal, to about 3.122% for all users as in Figure 12.

To further understand the trade-off between the gains in system throughput and fairness in the

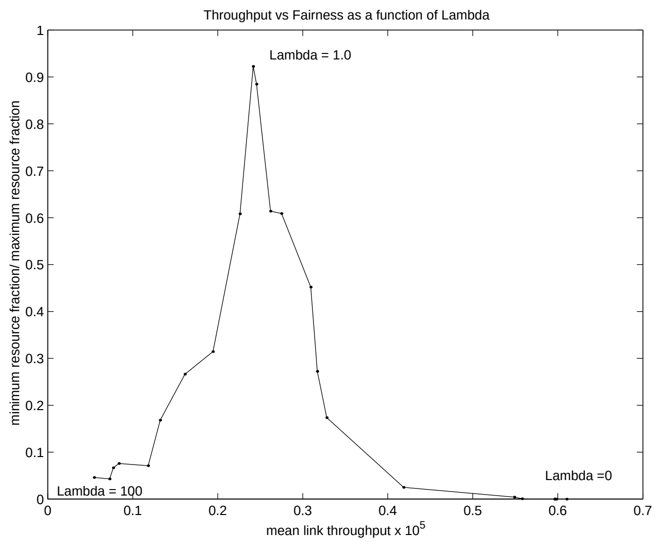

Fig. 13. Throughput vs Ratio of Minimum Resource Fraction to Maximum Resource Fraction as a Function of λ

allocation of scheduled slots we studied the trade-off between the throughput on the wireless link and $minimum\, resource\, fraction/maximum\, resource\, fraction$ as a function of $lambda$. For each value of lambda ranging from 0 to 10.0, we determined the mean link throughput and the ratio of $minimum\ resource\ fraction$ and $maximum\ resource\ fraction$ over 10 Monte Carlo iterations. It can be seen from Figure 13 that the throughput is maximum at $\lambda = 0$ resulting from the scheduling of higher SNR users that can support higher data rates. As λ is increased from 0, the throughput monotonically decreases. The ratio $minimum\, resource\, fraction/maximum\, resource\, fraction$ is a measure of fairness. When λ is 0, practically no resources are allocated to the users with low SNR, making the minimum resource fraction 0. At the other extreme, when λ is increased beyond 10, virtually all slots are allocated to the users with bad channels. When λ is 0, practically no resources are allocated to the users with low SNR, making the minimum resource fraction almost 0. At the other extreme, when λ is increased beyond 100, the system becomes $rate\ fair$ and almost all slots are allocated to the users with bad channels. Once again, the $minimum\ resource\ fraction$ approaches zero and dominates the ratio. The difference between the numerator and the denominator decreases as λ approaches 1.0, as indicated by the peak in Figure 13. The metric with $\lambda = 1.0$ is identical to that in the Proportional Fair Scheduler. From the graph, we observe that the PF Scheduler maximizes the resource fairness metric at the expense of link throughput which is at 40% of its maximum value.

VII. Design Considerations to Improve TCP Performance with the Proportional Fair Scheduler

In this section we address design issues that can alleviate the effect of delay jitter and rate variations on TCP performance.

- ***Trade-off between Resource Fairness and Rate Fairness***: In a typical cell, there is a reasonable variance in the users' nominal SNR's. Our findings in Section V clearly show that a fairness metric that ensures resource sharing alone can cause high variation in TCP performance. In the metric used in Section VI Increasing the parameter λ biases the system towards *rate-fairness*. Using a metric with $\lambda > 1$ may improve performance users with towards the edge of the cell with lower nominal SNRs. We are currently investigating the trade-off between rate-fairness and resource-fairness.
- ***Choice of α in the PF Scheduler***:Our simulation results show diminishing gains in wireless link throughput on decreasing α much below 0.01. TCP performance begins to suffer from increased scheduling jitter for very small α.
- ***TCP Sack***: We are currently investigating the use of TCP Sack instead of TCP Reno. Preliminary findings show that the effect of rate variation and jitter can be alleviated to some extent by using TCP Sack.
- ***Larger Queue Size***: Larger queue sizes are often proposed as a solution to absorb jitter. The bandwidth-delay product of the bottleneck link [14], which is the metric determining TCP performance, is about 10 packets or less in the system considered here. Hence, larger queue sizes only serve to increase delay without improving TCP throughput.

VIII. Conclusions and Future Work

This paper has comprehensively investigated the interactions between opportunistic scheduling mechanisms at the MAC layer and TCP performance. Our findings suggest system design considerations that can be used by operators. While our study was limited to long lived TCP connections, we intend to study the effects of opportunistic scheduling on transient connections such as HTTP downloads. We are currently investigating the use of TCP Sack and the trade-off between rate-fairness and resource-fairness to improve TCP performance. We are also in the process of characterizating opportunistic scheduling delay jitter and its effects on TCP in an analytical/quantitative framework.

Acknowledgments

The first author acknowledges extremely useful discussions with M. Srinivasan of Flarion Technologies Inc. The authors also thank all contributors to the large body of *ns* modules and discussions on the ns-users mailing list.

References

[1] H. Balakrishnan, S. Seshan, R.H. Katz, *A Comparison of Mechanisms for Improving TCP Performace over Wireless Links*, IEEE/ACM Transactions on Networking pp. 756-769,Vol.5, December 1997.
[2] H. Balakrishnan, S. Seshan, R.H. Katz, *Improving Reliable Transport and Handoff Performace in Cellular Networks*, ACM Wireless Networks Vol 1, No. 4, December 1995.
[3] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi, *CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users*, IEEE Communications Magazine, pp. 70-77, July 2000.

[4] 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 pp. 605-615, Vol.7, No. 5, October 1999.

[5] A. Gurtov, *Effect of Delays on TCP Performance*, Proceedings of IFIP Personal Wireless Communications '2001, August 2001.

[6] S. Hanly and D. Tse, *Multi-Access Fading Channels: Part II: Delay-Limited Capacities*, IEEE Transactions on Information Theory, v. 44, No. 7, Nov., 1998, pp. 2816-2831.

[7] Qualcomm 1xEV-DO, overview available at www.qualcomm.com/cdma/1xEV/papers

[8] A. Bakre and B.R. Badrinath, *Hand-off and System Support for Indirect TCP/IP*, Proceedings of the Second Usenix Symposium on Mobile and Location-Independent Computing, April 1995.

[9] Mun Choon Chan and Ramachandran Ramjee, *TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation*, Proceedings Mobicom, September 2002.

[10] W. C. Jakes, *Microwave Mobile Communications*, Piscataway, NJ : IEEE Press, 1993.

[11] A. Jalali, R. Padovani and R. Pankaj, *Data throughput of CDMA-HDR a high efficiency high data rate personal communication wireless system*, IEEE Vehicular Technology Conference 2000.

[12] R. Knopp and P. Humblet, *Information Capacity and Power Control in Single Cell Multiuser Communications*, IEEE ICC '95, Seattle WA, June 1995.

[13] J. Korhonen, O. Aalto, A. Gurtov and H. Laamanen, *Measured Performance of GSM HSCSD and GPRS*, Proceedings of the IEEE Conference on Communications, June 2001.

[14] T.V. Lakshman, U. Madhow and B. Suter, *TCP/IP Performance with Random Loss and Bidirectional Congestion*, IEEE/ACM Transactions on Networking, vol8, no. 5, Oct. 2000

[15] X. Liu, E.K.P. Chong and N. Shroff, *Transmission Scheduling for Efficient Wireless Resource Utilization with Minimum-Performance Guarantees* Proceedings of the 2001 Vehicular Technology Conference, pp. 824-828, October 2001.

[16] S. Lu, V. Bhargavan and R. Srikant, *Fair scheduling in wireless packet networks*, IEEE/ACM Transactions on Networking, vol. 7, pp.473-489, August 1999.

[17] Network Simulator, ns (version2) *www.isi.edu/nsnam/ns*

[18] T.S.E. Ng, I. Stoica, H. Zhang, *Packet fair queueing algorithms for wireless networks with location-dependent errors*, IEEE Infocom'98, New York, vol.3, 1998

[19] S. Ostermann, *tcptrace*, http://www.tcptrace.org

[20] J. B. Postel, *Transmission Control Protocol, RFC 793*, RFC, Information Sciences Institute, CA, September 1981.

[21] W. Stevens, *TCP Slow Start, Congestion Avoidance, Fast Retransmit and Recovery Algorithms, RFC 2001*, IETF RFC, January 1997.

[22] R. Srinivasan and J. Baras, *Opportunistic Scheduling in Wireless Systems and TCP Performance*, Technical Report, TR 2002-48, I.S.R. University of Maryland, October 2002.

[23] D. Tse and S.V. Hanly, *Multi-Access Fading Channels: Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities*, IEEE Transactions on Information Theory v. 44, No. 7, Nov., 1998, pp. 2796-2815.

[24] D. Tse, *"Opportunistic Communications: Smart Scheduling and Dumb Antennas"*, presentation available at http://degas.eecs.berkeley.edu/dtse/opp_comm.html

[25] *"Third Generation Partnership Project"*, RLC Protocol Specification (3G TS 25.322:), 1999.

[26] R. Yavatkar and N. Bhagwat, *"Improving End-to-End Performance of TCP over Mobile Internetworks"*, Mobile 94 Workshop on Mobile Computing Systems and Applications, December 1994.

[27] TIA/EIA/cdma2000, *"Mobile Station- Base Station Compatibility Standard for Dual Mode Wideband Spread Spectrum Cellular Systems"*, Telecommunication Industry Association, Washington, 1999.