Effect of Channel Variation in IP/cdma2000 Interconnection Performance

Vikas Paliwal, Parsa Larijani, Ioannis Lambadaris, and Biswajit Nandy
Broadband Networks Laboratory
Carleton University, Ottawa, ON, Canada
{vpaliwal, larijani, ioannis, bnandy}@sce.carleton.ca

Abstract—**In order to support high data rate requirements and effectively manage the scarce wireless resources, additional bandwidth channels are allocated and taken away from mobile stations in 3G wireless data networks quite frequently. A TCP sender connected to the mobile, on seeing ACKs coming at a faster pace after additional bandwidth allocation, turns overtly optimistic and injects data into the network in a more bursty manner that might be excessive for an intermediate router, thereby leading to loss of multiple packets and subsequent prolonged recovery and periods of underutilization. In this work, we characterize this problem using an analytical model for losses based on continuous flow approximation as well as an extensive simulation setup. We also illustrate how bandwidth oscillations create more severe congestion than an increase in number of users to the extent that even RED algorithm is unable to check the sharp growth of queues. As a result, multiple packets are lost in a droptail fashion. We further demonstrate the dependence of congestion due to bandwidth allocation on the time during which mobiles' rates are increased and observe the degradation in performance for typical load scenarios.**

I. Introduction

The current trend in cellular market is to provide data services together with voice services. This can be seen in evolution of newer protocols and standards to support high-speed data services [1], [2]. However, cellular wireless networks are challenged by several problems like losses over wireless links and scarce channel resources. While the losses can largely be mitigated by suitable link layer retransmission mechanism such as RLP [3] to provide added reliability over the wireless links, limited RF spectrum is still a problem that needs to be addressed effectively. One solution to effectively manage the limited resources is to dynamically share them among the users. This scheme calls for assignment of additional data rate channels to mobile users for specific durations based on user's demand, radio conditions and data backlog.

Transmission Control Protocol (TCP) [15] is the most widely used transport layer protocol. However, TCP works with assumptions that do not work well with problems specific to wireless networks. One such problem arises when the aggregate bandwidth of radio links changes abruptly due to allocation of additional channels and the TCP senders begin to receive ACKs at a faster pace (sometimes referred to as *ACK compression*) and in response that, they put new data packets into the network at a rate that might be excessive for some other link in the network, which leads to loss of an excessive number of packets. This scenario for cdma2000[1] networks is shown in figure 1, where a mobile user is having a TCP connection with a fixed host in the Internet. The radio link beyond the base-station controller (BSC) is a variable rate link that might cause the TCP sender to inject more data into the network than it can bear. The new shifted bottleneck could be anywhere in the path of the TCP connection. However since BSC is the terminal point where mobiles' data flows split into various link buffers and it serves all the mobiles, it is most likely to be the new bottleneck. Thus, because of radio links' rate variability, it might be possible that bottlenecks keep on shifting between the radio link and one of the intermediate shared buffers, e.g. BSC's input buffer.

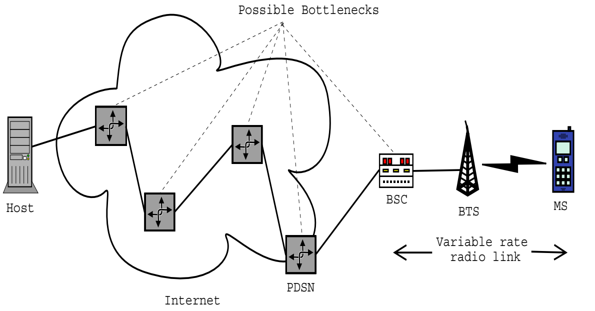


Fig. 1. A network view of a mobile's TCP flow over an IP/cdma2000[1] interconnection.

Congestion in communication networks has traditionally been assumed to be caused due to greater number of users joining to share a network's limited resources. However in this work, we introduce a new phenomenon of network congestion due to variable bandwidth radio links, with number of users fixed, and we show that it is a more severe source of congestion. This is because of the fact that in a variable rate scenario with ACK compression phenomenon, the TCP sender, even in a steady congestion avoidance phase, might be fooled to believe that greater bandwidth is available in the whole network and begins to send at that rate. Because of its large window size, it keeps on putting packets into the network till the highly undesirable event of multiple packet drop at one of intermediate routers occurs and a possible prolonged timeout is triggered afterwards.

Further, we show that the most widely used RED mechanism in shared routers, if configured as per existing guidelines,

This work was supported in part by grants from: Natural Sciences and Engineering Research Council of Canada (NSERC) and Communications and Information Technology Ontario (CITO).

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

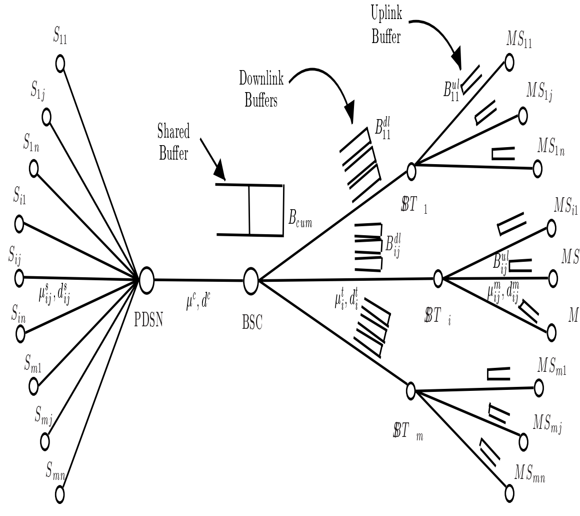


Fig. 2. Buffering scheme in cdma2000 data services (Table I describes the symbols used). MS stands for mobile station, BTS for base-station transceiver subsystem and BSC stands for base-station controller.

is unable to check this congestion due to bandwidth variations and behaves like an ordinary droptail gateway that drops multiple packets at once. This leads to various problems such as global synchronization, reduced throughputs and network underutilization. We show that if all the mobiles served by a shared buffer change their link rate in a time duration that is not significantly larger than the round trip delay, these problems persist. We further show that for certain values of frequency of bandwidth variations, the overall system utilization can fall as low as 80%.

II. Background

A typical buffering scheme based on the cdma2000 standard [1] is shown in figure 2. Radio link rate variations are quite frequent in wireless networks [3]. For instance, in cdma2000 networks, on top of fundamental channels (FCH) (of base rate 9.6 kb/s for Rate Set 1) supplemental channels (SCH) could be assigned by the Base Transceiver Subsystem (BTS) for a particular cell to the mobile stations for specific durations in the range 20ms to 5.12 seconds. The rates for the SCH are 19.2, 38.4, 76.8 and 153.6 kb/s. Such allocations can be done in a variety of schemes at individual BTSs, and one such scheme is *finite-burst* mode [9] (explained later in section IV-E). Clearly, sudden allocation of a supplemental channel means an equivalent swing in the total bandwidth of the radio link which might cause problems at intermediate shared buffers, as discussed earlier.

Queue Management (QM), and specially Active Queue Management (AQM), has been an active area for research ever since the introduction of Random Early Drop (RED) [5], but all the work in this area focused on mainly wired networks with nearly fixed bandwidths and delay constraints. Network designers for these networks, after identifying the bottleneck, based on their experience of some a priori estimates on mean delay and bandwidths were able to comfortably come up with parameters for router settings. The problem of queue management in wired domains was simple in the sense that only a few parameters had to be tuned for a well-understood and well-behaved wired network. However such assumptions are not true in wireless domains where bandwidths oscillate and delays can vary sharply. This poses problems for both queue management techniques and transport protocols' design. Whereas a sharp rise in bandwidth poses problems for the former due to a sudden enormous burst of packets delivered by the TCP sources due to ACK compression, on the other hand, the latter is confronted with the possibility of spurious timeouts caused by sudden removal of supplemental channels [9]. It can be shown that the latter problem is rare in occurrence and occurs for particular setups of TCP window settings and the authors of [9] report that window sizes somewhat larger than delay-bandwidth product of the network can eliminate the problem altogether. Queue management problem is one that demands greater attention due to its complex nature and the fact that conventional wisdom of "large window sizes help wireless" means greater losses in cases of congestion at intermediate routers, as we will see later.

The problem of queue management is complicated due to the fact that the location of bottleneck changes frequently based on current state of resource allocation. With the proposed introduction of higher data rate techniques like CDMA 1X-EVDO [2] that aim at providing higher data rates to the level of 2 Mb/s, a global view of congestion and queue management(QM) issues that does not assume wireless links as the only bottleneck would be more suitable. A review paper [8] on current ongoing research on wireless links considers a study of QM issues

TABLE I
Symbols used in Figure 2 and Simulation Parameters

Description	Symbol	Value(s)
Application		FTP
TCP version		Reno
TCP Segment Size		460, 960 Bytes
Receiver's buffer size		32 kB
TCP Segments per ACK		1
TCP Minimum RTO		$1s$
IP Packet Size		500, 1000 bytes
Number of BTS	m	1, 10
Number of MSs in each BTS	n	1, 10
Mobile's Rate	R_{ij}	$9.6 - 163.2$ kb/s
Bandwidths	μ^s_{ij} μ^c μ^t_i μ^m_{ij}	100 Mb/s, for all i, j 100 Mb/s 100 Mb/s, for all i R_{ij} kb/s, for all i, j
Delays	d^s_{ij}, d^m_{ij}, d^t_i, d^c	37.5ms each
BSC's Buffer	B_{cum}	Service rate[1] = $m \cdot n \cdot 38.4kb/s$ Droptail(cap.= 5 kB) RED(cap. = 300 kB, $min_{th} = 20kB$, $max_{th} = 60kB$).
Link Buffer	B^{dl}_{ij}	service rate = R_{ij} cap. = ∞

[1] BSC's service rate is chosen such that it is less than the maximum aggregate bandwidth in air interface.

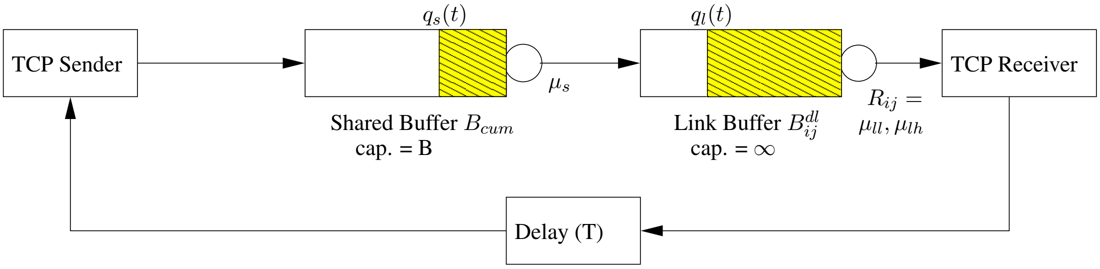


Fig. 3. Model for losses due to rate change in a radio link.

in shifting bottleneck scenarios as highly desirable (refer page 88, column 2). Our work, presents an extensive study based on analysis and simulations. It differs from earlier works in the sense that it does not consider the end wireless links as the only bottlenecks. Notably enough, some research efforts have been made towards the study of problem of radio link's QM [7] with the assumption of radio link as the only bottleneck and solutions have been proposed but no studies in our knowledge have been performed on the issue of possibility of other network bottlenecks that impact the overall performance in a significant manner. We would also like to mention that link queues have small sizes and any kind of QM technique employed for links only would not make appreciable overall difference. On the other hand, our analysis of QM that encompasses other network nodes together with wireless link, would be more useful in improving the performance.

III. Motivation

To develop an intuitive understanding of the congestion problem, we try to model the impact of rate variations by an analytical model for losses based on continuous flow approximation as used in some previous works [11], [13], [14]. Figure 3 shows the model we are considering for analysis of a single mobile's flow which is assumed to have a fixed share of resources at the shared buffer, i. e. an individual service rate of μ_s, and an available buffer space B. Note that we are assuming this to be reserved for the particular mobile under consideration. The downlink buffer, B^{dl}_{ij}, is the radio link buffer that stores the frames destined for the mobile. The service rate of this link buffer is determined by the aggregated rates of channels allocated to the mobile and in a *dynamic* environment, it keeps on switching between a lower value, $R_{ij} = \mu_{ll}$ corresponding to the base FCH rate, say 9.6 kb/s and the higher rate of μ_{lh} equal to sum of rates of an FCH and an SCH, say $R_{ij} = FCH_{1X}+SCH_{16X} = (9.6+153.6)kb/s = 163.2kb/s$. Shared buffers are normally designed for nominal loads and we are specifically interested in scenarios when μ_s falls in between these two values (say a rate of mere SCH_{4X} = 38.4 kb/s for these values of μ_{ll} and μ_{lh}) as this configuration causes the *oscillation* of *queued workloads* between shared buffer and link buffer as R_{ij} fluctuates. Link buffers usually have very large capacities and most implementations would involve assigning a supplemental channel when data backlog is too much, so drop events in link buffers are rare. On the other hand, shared buffers, if not designed properly to absorb the oscillations, might cause excessive losses as shown next. The other conventions that we use are the contents of link buffer and shared buffers, $q_l(t)$ and $q_s(t)$, respectively and the round-trip propagation delay, T, lumped between the TCP receiver and the sender, although similar results can be derived by distributing this delay elsewhere.

To illustrate the impact of a single rate switching, let us assume that the mobile was operating with the lower link rate of μ_{ll} for a long duration prior to time t_0. Assuming that the window size of TCP sender is fixed at W, the contents of link and shared buffers for long-term steady state prior to t_0 are $q_l(t) = W - \mu_{ll} \cdot T$ and $q_s(t) = 0$. Now if at instant t_0, the link rate is switched to the higher value of μ_{lh}, the queued workload in link buffer begins to move to the shared buffer since link buffer empties. This is due to the fact that ACKs are generated at a faster rate and TCP sender sends traffic at a very fast rate to the shared buffer. To calculate the losses at shared buffer, we are specifically interested in the starting and ending times, t_{start} and t_{end}, when the packets come to shared buffer at the higher rate of μ_{lh}, as it is during this period the shared buffer begins to fill and possibly overflows. Clearly, leading edge of packets at the higher rate reaches shared buffer at $t_{start} = t_0 + T$. Now for ending time, one of the following three possible cases have to be considered:

Case 1: The link buffer runs empty before the leading edge of packet flow at higher rate reaches shared buffer. Clearly for this case, after the bandwidth switch at $t = t_0$, $\frac{dq_l(t)}{dt} = \mu_{ll} - \mu_{lh}$. The time required for link queue to become empty is $\frac{0-q_l(t_0)}{\mu_{ll}-\mu_{lh}} = \frac{W-\mu_{ll}\cdot T}{\mu_{lh}-\mu_{ll}}$. So, the governing condition for this case to happen is $T \geq \frac{W-\mu_{ll}\cdot T}{\mu_{lh}-\mu_{ll}}$, or $\mu_{lh} \geq \frac{W}{T}$. Clearly, this change in state of link queue from being non-empty to empty will reach TCP sender after a delay T, when it stops sending the packets to shared buffer at the higher rate μ_{lh}. Thus,

$$t^*_{end} = t_0 + \frac{W-\mu_{ll}\cdot T}{\mu_{lh}-\mu_{ll}} + T \quad \text{if } \mu_{lh} \geq \frac{W}{T} \tag{1}$$

Case 2: The leading edge of packet flow at higher rate reaches the shared buffer before the link buffer runs empty. In this case, for the first T seconds after t_0, the shared buffer's inflow rate is μ_{ll}, for $t_0 \leq t \leq (t_0 + T)$, afterwards it becomes μ_{lh}, for $t_0 + T \leq t$ and while $q_l(t - T) \geq 0$. Consequently outflow rate of shared buffer is μ_{ll} till $t < t_0 + T$ and μ_s, afterwards. The draining rates of link buffer are, $\frac{dq_l(t)}{dt} = \mu_{lh} - \mu_{ll}$ for $t_0 < t < t_0+T$ and $(\mu_{lh} - \mu_s)$ afterwards while $q_l(t) \geq 0$. Thus at $t = t_0 + T$, the content of link buffer is, $q_l(t_0+T) = q_l(t_0) - (\mu_{lh} - \mu_{ll}) \cdot T = W - \mu_{lh} \cdot T$. So the total time after t_0, required for the link buffer to run empty is $T + \frac{q_l(t_0+T)}{\mu_{lh}-\mu_s} = \frac{W-\mu_s \cdot T}{\mu_{lh}-\mu_s}$. Since there is a delay of T seconds between link buffer and TCP source, shared buffer will stop receiving packets at high rate at,

$$t^{**}_{end} = t_0 + \frac{W-\mu_s \cdot T}{\mu_{lh}-\mu_s} + T \quad \text{if } \mu_{lh} < \frac{W}{T}. \tag{2}$$

Case 3: The leading edge of packet flow at higher rate reaches the shared buffer before the link buffer runs empty and the TCP sender becomes aware of a drop event in shared buffer before it stops receiving packets at higher rate. Clearly, as described earlier, if a TCP sender receives first trail of packet flow after a drop event, it stops sending any further packets irrespective of arrival rate of ACKs. In practice, it waits till it gets three duplicate ACKs, after which it retransmits the lost segment. Now for our model, the shared buffer , for $t > t_0 + T$, begins to fill at a rate $\frac{dq_s(t)}{dt} = \mu_{lh} - \mu_s$. It gets filled after another $\frac{B}{\mu_{lh}-\mu_s}$ seconds and the buffered packets in shared buffer at this instant take another $\frac{B}{\mu_s}$ seconds to clear the shared buffer after which the trail reaches the link buffer. The remaining content of link buffer at this instant is $W - \mu_{ll} \cdot T - (\mu_{lh} - \mu_{ll}) \cdot T - (\mu_{lh} - \mu_s) \cdot (\frac{B}{\mu_{lh}-\mu_s} + \frac{B}{\mu_s})$. This remaining data in the link buffer ahead of the trail behind first dropped segment drains at rate μ_{lh} and after another T seconds the trail behind first drop event reaches the TCP sender and at that point TCP sender stops transmission. So, for this case:

$$\begin{aligned} t^{***}_{end} = \quad & t_0 + T + \frac{B}{\mu_{lh}-\mu_s} + \frac{B}{\mu_s} \\ & + \frac{W-(\mu_{ll} \cdot T)-((\mu_{lh}-\mu_{ll}) \cdot T)-((\mu_{lh}-\mu_s) \cdot (\frac{B}{\mu_{lh}-\mu_s}+\frac{B}{\mu_s}))}{\mu_{lh}} \\ & +T \\ & = t_0 + T + \frac{B}{\mu_{lh}-\mu_s} + \frac{W}{\mu_{lh}} \end{aligned} \tag{3}$$

The governing condition for this case can be obtained by using the values of t_{end} in equation 2 and 3 and noting that the value in latter should be smaller than the one in former to let the TCP sender become aware of first drop before it stops receiving packet flow at higher rate. This leads to a condition,

$$t^{***}_{end} \leq t^{**}_{end} \Rightarrow \mu_{lh} \leq \frac{W}{T + \frac{B}{\mu_s}} \tag{4}$$

Based on equations 1-4, the time at which the shared buffer stops receiving packets at higher rate of μ_{lh} either due to link buffer becoming empty or TCP sender recieving information about first drop event is given by:

$$t_{end} = \begin{cases} t_0 + T + \frac{B}{\mu_{lh}-\mu_s} + \frac{W}{\mu_{lh}} & \text{if } \mu_{lh} \leq \frac{W}{T+\frac{B}{\mu_s}} \\ t_0 + T + \frac{W-\mu_s \cdot T}{\mu_{lh}-\mu_s} & \text{if } \frac{W}{T+\frac{B}{\mu_s}} < \mu_{lh} < \frac{W}{T} \\ t_0 + T + \frac{W-\mu_{ll} \cdot T}{\mu_{lh}-\mu_{ll}} & \text{if } \mu_{lh} \geq \frac{W}{T} \end{cases} \tag{5}$$

To calculate the volume of packets dropped from the shared buffer we note that no losses occur while the buffer gets filled and losses occur afterwards. So from time t_{start} to $t_{start} + \frac{B}{\mu_{lh}-\mu_s}$, no losses occur and afterwards losses occur at a rate $(\mu_{lh} - \mu_s)$ until time t_{end}. These concepts together with equation 5 can be used to calculate the loss volume (L_V) that comes out as:

$$L_V(\mu_{ll}, \mu_s, \mu_{lh}, W, B, T) =$$

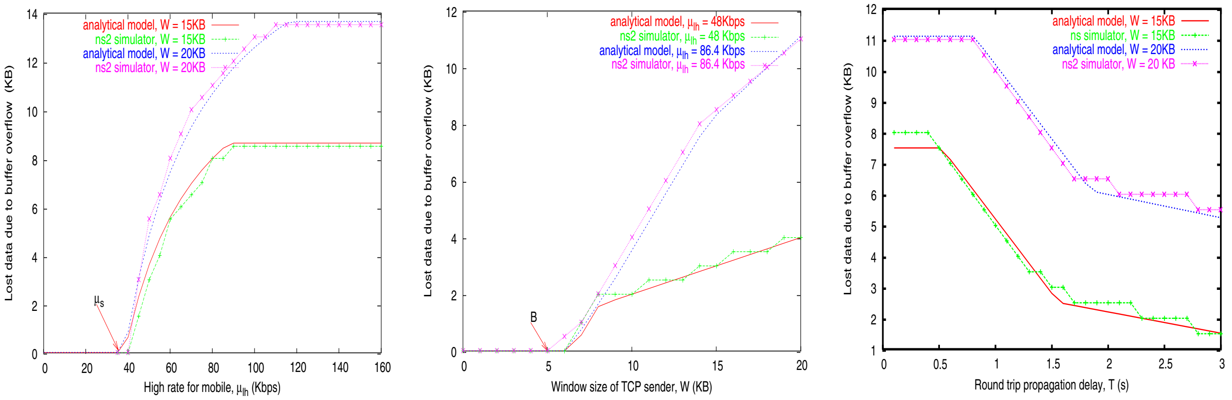
$$\begin{cases} \frac{W \cdot (\mu_{lh}-\mu_s)}{\mu_{lh}} & \text{if } \mu_{lh} \leq \frac{W}{T+\frac{B}{\mu_s}} \\ max[(W - B - \mu_s \cdot T), 0] & \text{if } \frac{W}{T+\frac{B}{\mu_s}} < \mu_{lh} < \frac{W}{T} \\ max[(\frac{(W-\mu_{ll} \cdot T)(\mu_{lh}-\mu_s)}{(\mu_{lh}-\mu_{ll})} - B), 0] & \text{if } \mu_{lh} \geq \frac{W}{T} \end{cases} \tag{6}$$

IV. Simulations

For the purposes of simulations, we used an implementation of cdma2000's link layer protocol, RLP[3], in *ns2* [17] simulator. This RLP module made available at [16] can be used to model the SCH allocations and wireless losses. In a companion report [16] on study of impact of RF errors, it was found that moderate level of wireless frame error rates (FERs) and fading-induced correlation produce only a nominal jitter in round trip times (RTTs) for the TCP flows and such jitters are easily absorbed in long-term performance. On the contrary, rate variation produces sharp changes in RTTs and hence is the main contributor to degraded system performance. Thus, keeping in mind that impact of RF errors is minimal and to focus on rate variations, we assumed error-free wireless links. A long FTP session is assumed to be conducted between fixed hosts in external internet and mobile stations. The setup is exactly as shown in figure 2 and other parameters that will be used (unless otherwise specified) are presented in table I.

A. Dependence on window size, bandwidth swing and round-trip delay

Equation 6 shows the relation of losses due to bandwidth switching for a single mobile under the assumptions of fixed resources for it at the shared buffer. We simulate the similar scenario in the simulation environment by using a droptail buffer of fixed size and a fixed service rate for packets destined for the mobile. The parameters are shown in table I. Figure 4 shows the comparison between the two models. It can be clearly seen that, except for the slight inaccuracy due to continuous flow approximation, the two models are in close agreement. Figure 4(a) shows that, for fixed W, T, μ_s, μ_{ll} and B, packet loss volume, L_V, first increases and then remains nearly constant. The limiting value of losses for increasing μ_{lh} will be, $\lim_{\mu_{lh} \to \infty} L_V(\mu_{ll}, \mu_s, \mu_{lh}, W, B, T) = (W - B - \mu_{ll} \cdot T)$. So, for very sharp bandwidth swing, the losses can be as high as total buffered data at low bandwidth, $W - \mu_{ll}T$, minus the buffer space available at shared buffer, B. Figure 4(b) shows the loss variation with window sizes. It can be seen that losses increase linearly with window size and the slope is governed by the conditions in equation 6. Loss variation with round-trip delay is shown in figure 4(c), which shows that, keeping other parameters fixed, losses decrease with increased propagation delay.



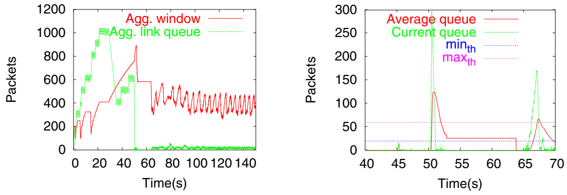
(a) $T = 0.3$s, W = 15 kB, 20 kB (b) $T = 0.3$s, μ_{lh} = 48 kb/s, 86.4 kb/s (c) μ_{lh} = 86.4 kb/s, W = 15 kB, 20 kB

Fig. 4. Variation of losses due to bandwidth change with, (a) mobile's higher switched rate, μ_{lh} , (b) TCP sender's window size, W, and, (c) round trip propagation delay, T. (Other parameters are kept constant at μ_{ll} = 9.6 kb/s, μ_s = 38.4 kb/s, B= 5 kB, packet size for *ns2* simulations = 500 bytes)

B. A worst-case scenario

Next we focus on scenarios with multiple mobile users and simulate a worst-case scenario where all the mobiles under a BSC simultaneously change their rates, so that the aggregate service rate of link buffers becomes greater than the service rate of shared buffer B_{cum} instantly. This scenario translates to shifting of bottleneck from wireless links to the shared buffer. Figure 5 shows this case when the rates of all the 100 mobiles are switched from 9.6 kb/s to 163.2 kb/s due to a 16X (153.6 kb/s) SCH allocation at 50s. This means a change in aggregate link bandwidth from 0.96 Mb/s to 16.32 Mb/s. In this scenario, all the queues of link buffers are shifted to shared buffer, which has insufficient buffer space and hence results in excessive dropping of packets. Our example with a shared buffer following RED discipline shows that RED algorithm is unable to check the sharp growth of queues and drops packets like an ordinary droptail gateway after hitting its buffer limit of 300 packets. The low pass filter characteristics employed in RED algorithm create lot of inertia in it so that it takes a long time for it to raise its average queue length above the threshold to perform any useful congestion indication action. Even when it begins to do so, it takes almost a round trip time for the congestion indication to reach the senders so that they can resort to any reduction in sending rates by window-halving mechanisms and by this time, lots of packets are lost in a droptail-like fashion. Packet traces show that the total dropped packets, because of this droptail like behavior, are 729, close to the aggregate window size of 915 packets of TCP senders. This is due to the fact that the shared buffer has very little space of 300 packets (set as per recommendations in [5], which does not account for transient bursts due to bandwidth changes), and on top of that, being a RED gateway, it begins dropping all incoming packets as soon as average queue reaches the max_{th} value of 60 packets.

It is also notable that subsequent timeout based recovery is also very prolonged. The reason for this is that before 50s, all the mobiles were operating at a lower bandwidth of 9.6 kb/s and larger link queues and hence their retransmission timeout (RTO) will be highly inflated (RTT for such low bandwidth networks is highly dominated by current bandwidth, e.g. each 1000 byte packet in link queue of base service rate of $9.6kb/s$ adds 1.1 seconds to RTT) and after multiple packets are dropped, the TCP sender will wait for a long time for the lost packets before it begins to retransmit. Thus, owing to high rate allocation, the link buffer drains fast and soon runs empty. Also, during the timeout based recovery after multiple losses at the shared buffer it remains empty as almost all the TCP senders keep on waiting for their outstanding packets that have been dropped by the shared buffer. This is exemplified in figure 5(b), where between 51s and 62s, both the shared buffer and the link queues are empty. Since both potential bottlenecks are empty, this is a period of under-utilization of the entire system. We call such periods as *dead-periods* and use them as a performance metric in our typical load scenarios later. Note that the average queue of RED remains constant at nearly 21 packets during 51s-62s even though actual queue is zero and no packets arrive. This is due to its implementation of not changing the average queue value when no packet arrives and changing it only when the first packet arrives, by an exponential decay mechanism based on link rate [5].

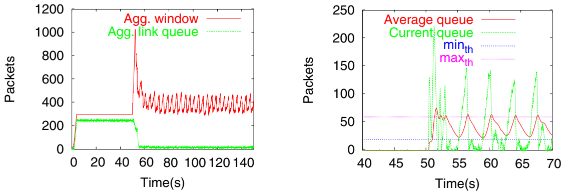


(a) Aggregate window, aggregate link queue. (b) Queue at shared buffer, B_{cum}

Fig. 5. Plots for simultaneous $16X$ SCH allocation to all 100 mobiles at $50s$.

C. Comparison with a scenario with variable number of users

To establish our claim of bandwidth changes as a more severe source of congestion, we benchmark the variable bandwidth scenario against the cases when number of users changes, as in [6], keeping the aggregate link bandwidth and shared buffer's service rate values as in section 4.2. To simulate an aggregate bandwidth swing from 0.96 Mb/s to 16.32 Mb/s, we change the number of active users from 6 to 100 at 50s, keeping the link rate for all the mobiles at 163.2 kb/s throughout. The plots for



(a) Aggregate window, aggregate link queue.

(b) Queue at shared buffer, B_{cum}

Fig. 6. Plots for more users joining simultaneously at $50s$.

this scenario are shown in figure 6. The new incoming users begin in slow start phase with an initial window of size one and quickly learn the allowable rates at shared bottleneck link. The mobile stations that were switched on from the start at 0s and were operating at larger windows are penalized by the RED algorithm and are forced to conform to the new changed network dynamics. All this happens very fast due to probabilistic dropping by RED wherein all the senders are not shut simultaneously as in previous case of variable bandwidth and flow of data across the shared buffer keeps on going at all times. Comparing this with figure 5, it is very clear that RED's random dropping mechanism is capable to handle congestion due to change in number of users.

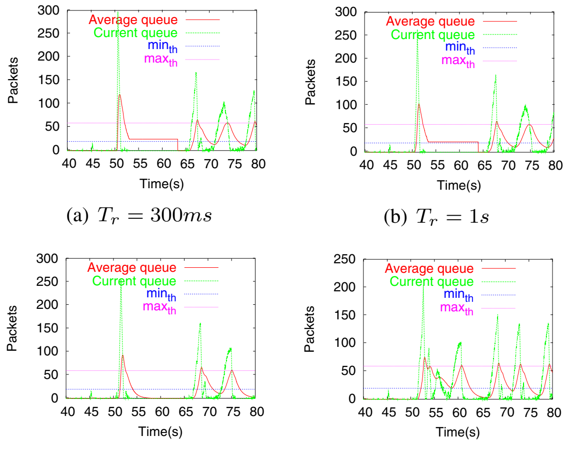
D. Impact of rise time for aggregate bandwidth switching

So far we considered a worst-case simultaneous allocation of SCH to all the mobile stations under a BSC. In a realistic scenario, the bandwidth allocations are usually *on demand* basis with little or no regulation. To simulate more gradual increases and to analyze the impact of rate of rise of aggregate bandwidth, we create a simulation setup in which mobiles are switched to higher rate sequentially over a time span of T_r seconds, so that time between switching to higher rate of two mobiles is $T_r/(m \cdot n)$. This scheme allows for aggregate output links' bandwidth, as seen by BSC, to change from 0.96 Mb/s to 16.32 Mb/s over T_r seconds when each of the mobile's rate is changed from 9.6 kb/s to 163.2 kb/s one after another in a sequential manner. In the down-switching, the rates are decreased sequentially over a span of fall time of T_f seconds.

Figure 7 shows the queueing behavior for BSC's buffer for an up-switching of the system beginning at 50s. The round-trip propagation delay is 300ms, as before. We plot for T_r values of 0.3ms, 1s, 2s and 5s. It can be clearly seen that in all the first three cases, severe under-utilization of system resources occurs as all the TCP senders have suffered multiple losses and are waiting to timeout. For $T_r = 5s$, the packet flow is not halted after bandwidth allocations but even then the instantaneous queue at BSC's shared buffer is often zero, leading to under-utilization. This indicates that the extent of degradation due to congestion phenomenon after bandwidth change is larger for rise-times that are not significantly larger than round-trip times. In most cases, changes in aggregate bandwidths lead to empty queues at shared buffers and system under-utilization.

E. Typical load scenario

A typical scheme for assigning higher data rate to mobiles is to allocate and de-allocate SCH in a *finite-burst* mode [9]. In



(a) $T_r = 300ms$

(b) $T_r = 1s$

(c) $T_r = 2s$

(d) $T_r = 5s$

Fig. 7. Impact of rise time of aggregate bandwidth on queue behavior at shared buffer.

this mode of operation, an SCH of 153.6 kb/s is assigned for duration T_B seconds at inter-allocation spacing of T_D seconds. These durations are in the range of 20ms to 5.12s. The allocation decisions are made based on the service requirements, radio conditions or data backlogs of the mobile. In the final part of this section, we look at the performance of RED for various values of burst-delay durations in this finite-burst mode of operation for the mobiles. We have chosen the rise and fall times for the 100 mobiles to be 10% of the burst and delay durations respectively, i.e. in each burst, the aggregate link rate reaches its maximum $0.1T_B$ seconds after the burst was given to the first mobile and vice versa for down-switching.

The *dead-period* performance metric that we used corresponds to periods when shared buffers and link buffers are all empty and captures under-utilization of both the possible bottlenecks in the cdma2000 data system. Based on this idea, the fractional *dead-period* of 100% system under-utilization, ρ_{dead}, could be defined as the ratio of total time, T_{dead}, during which the queue for BSC's output link, link queue of all mobiles - are all empty and the total simulation time T_{sim}. Clearly this metric quantifies the fraction of time during which the TCP senders are waiting to timeout after their packets are being dropped by BSC's shared queue on sudden rate increase. During these periods, all the link queues as well as the BSC's output queue remain empty meaning waste of resources due to shared buffer's droptail-like behavior. The length of fractional dead-period versus various values of burst, delay durations is shown in Table. II.

As can be seen in Table. II, for smaller values of T_B and T_D, the fractional *dead-periods* are negligible. This is because TCP does not sense bandwidth changes at high frequency of allocation/de-allocation (i.e $1/(T_B + T_D)$), ACK compression does not occur for long and hence TCP senders do not send data packets very fast to cause any havoc. However, at larger values, dead-periods are longer because now TCP senses this bandwidth change and puts in data at an excessive rate for BSC's shared buffer leading to loss of many packets and subsequent inefficient timeout-based recovery. For very large values of T_B and T_D, since not many bandwidth changes occur during simu-

	T_B(secs)												
T_D(secs)	0.02	0.1	0.5	1	2	3	4	5	6	7	8	9	10
0.02	0.26	0.66	0.66	0.54	0.71	0.70	0.62	0.62	0.70	0.62	0.62	0.59	0.59
0.1	0.0	0.59	3.62	2.26	1.24	1.34	0.95	1.04	1.22	0.94	0.90	0.97	0.97
0.5	0.0	0.0	4.5	4.78	2.80	3.02	2.68	2.70	1.52	1.76	1.90	1.56	1.93
1	0.0	0.0	1.9	1.7	2.3	2.46	2.16	2.85	7.74	1.36	1.67	1.72	1.74
2	0.0	0.0	0.76	0.38	1.90	0.92	0.52	1.78	6.56	0.57	1.5	0.78	0.83
3	0.0	0.04	0.78	0.59	0.97	0.82	0.56	3.85	9.43	2.0	1.48	1.34	1.58
4	0.0	0.14	0.38	0.38	1.10	0.5	0.59	6.25	12.04	14.88	1.38	1.52	2.48
5	0.0	0.0	0.38	0.28	1.15	0.35	0.82	7.10	13.9	16.18	19.84	3.56	3.39
6	0.0	0.0	0.70	0.14	0.70	0.95	1.8	7.02	10.6	8.98	9.0	2.43	3.52
7	0.0	0.0	0.41	0.17	1.48	0.88	2.28	6.70	8.72	11.6	11.84	2.83	3.56
8	0.0	0.0	0.10	0.52	1.7	0.64	2.16	6.41	9.09	8.90	15.1	2.30	2.66
9	0.0	0.0	0.34	1.10	1.84	0.47	3.04	8.30	9.33	10.16	14.16	8.91	3.2
10	0.0	0.0	0.23	1.28	0.22	0.20	3.62	9.25	15.3	16.2	17.4	2.93	4.90

TABLE II

PERCENTAGE DEAD-PERIODS(100% UNDERUTILIZATION DEFINITION) VERSUS BURST-DELAY DURATIONS.

lation time, this effect is less pronounced. It can be clearly seen that for certain values, fractional dead-period can be as large as nearly 20 percent. This is a waste of resources as during one-fifth of the time the whole wireless systems is not being used in the sense that both BSC's output queue and all link queues are empty and translates to a dismal value of overall system utilization at mere 80%.

Until this point we have seen how a rise in aggregate bandwidth poses problems for buffers at intermediate nodes. We have also seen that only a rise in a time period not significantly larger than round-trip times(RTTs) produces erratic behavior. A RED gateway is effectively able to handle a slow, gradual rise in bandwidth. Also, in the long-run, RED is able to handle the extra bandwidth and under-utilization periods are only after the sudden allocation of bandwidth. So we can say that target problem's domain is limited to only short-term under-utilization after a sharp rise in aggregate bandwidth in a period not very large than RTT. For long-term buffer management with contant aggregate link rates or cases with very large rise times, RED performs reasonably well. Nonetheless the significance of these short-term under-utilizations can not be ruled out. The reason for it is twofold. First, the dead-periods of under-utilization are not really that short. In our simulations we have shown that these periods are roughly 10-12s which is by no means a small duration to be neglected that easily. Second, the allocations/deallocations are done quite frequently and during a typical load period this will be done several times, as a consequence of which, each allocation will create a dead-period of underutilization. These multiple dead-periods will have pronounced impacts on long-term aggregate throughput for mobiles in a typical load scenario.

V. CONCLUSIONS

In this work, we introduced a new problem of congestion due to variable rate links in wireless data networks. By means of analysis and simulations, we identified and characterized this problem. We analyzed the system performance under variety of configurations and based on them, we provide extensive results to stimulate further research to develop a solution to this problem. Such a solution would involve either some modifications in existing queue management schemes to address the greater burstiness introduced due to rate variations or some suitable scheme for channel allocations that avoids sudden changes in aggregate bandwidth for outgoing links at the shared buffer. It is our belief that the insight into the nature and causes of the congestion problem developed in this work will be useful to develop a solution to this problem.

ACKNOWLEDGEMENTS

The authors would like to thank Gamini Senarath for the help and support.

REFERENCES

[1] TIA/EIA/IS-2000.1. Introduction to cdma2000 standards for spread spectrum systems, March 1999.

[2] P. Bender *et al*. CDMA/HDR: A Bandwidth Efficient High Speed Wireless Data Service for Nomadic Users. In *IEEE Communications Magazine*, pages 70–77, July 2000.

[3] TIA/EIA/IS-707-A-1. Data services options for spread spectrum systems - Radio Link Protocol type 3. December 1999.

[4] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. *RFC 2582, IETF*, April 1999.

[5] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. In *ACM Trans. on Networking(ToN)*, pages 397–413, August 1993.

[6] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. *under submission*, August 2001.

[7] M. Sagfors, R. Ludwig, M. Meyer, and J. Peisa. Queue management for TCP traffic over 3G links. In *Proc. of IEEE Wireless Communications and Networking Conference(WCNC'03)*, March 2003.

[8] A. Gurtov and S. Floyd. Modeling Wireless Links for Transport Protocols. In *ACM Comp. Comm. Review(CCR)*, pages 85–96, April 2004.

[9] M. Yavuz and F. Khafizov. TCP over wireless links with variable bandwidth. In *Proc. of Vehicular Technology Conference(VTC'02)*, pages 1322–1327, Sept. 2002.

[10] M. Yavuz and F. Khafizov. Running TCP over IS-2000. In *Proc. of Intl. Conf. on Comm.(ICC'02)*, pages 3444–3448, May 2002.

[11] Y. Wardi and B. Melamed. Loss volume in continuous flow models: fast simulation and sensitivity analysis. In *Proc. of IEEE MED*, July 17-19, 2000.

[12] M. Chan and R. Ramjee. TCP/IP performance over 3G wireless links with rate and delay variation. In *Proc. of ACM Mobicom*, September 2002.

[13] B. Liu, D. Figueiredo, Y. Guo, J. Kurose, and D. Towsley. A study of network simulation efficiency: fluid simulation vs. packet-level simulation. In *Proc. of IEEE Infocom 2001*.

[14] V. Misra, W. Gong and, D. Towsley. A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proc. of ACM SIGCOMM'00*, September 2000.

[15] W. Stevens. *TCP/IP Illustrated*, Addison Wesley, 1993.

[16] *http://www.sce.carleton.ca/∼vpaliwal/index_files/pub.html*

[17] *http://www.isi.edu/nsnam/ns/*