Wireless Personal Communications (2006)
DOI: 10.1007/s11277-006-9049-2

Congestion due to Rate Variations in cdma2000 Data Networks

VIKAS PALIWAL, PARSA LARIJANI, IOANNIS LAMBADARIS
and BISWAJIT NANDY

Broadband Networks Laboratory, Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, K1S 5B6
E-mail: vpaliwal@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 at a rate 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. We also try to identify the boundary for stable operation of RED and finally present some possible methods for improving the performance.

Keywords: bandwidth oscillation, AQM, cdma2000, continuous flow models

1. 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, 3]. 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 [4], 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) [18] 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 [3] 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

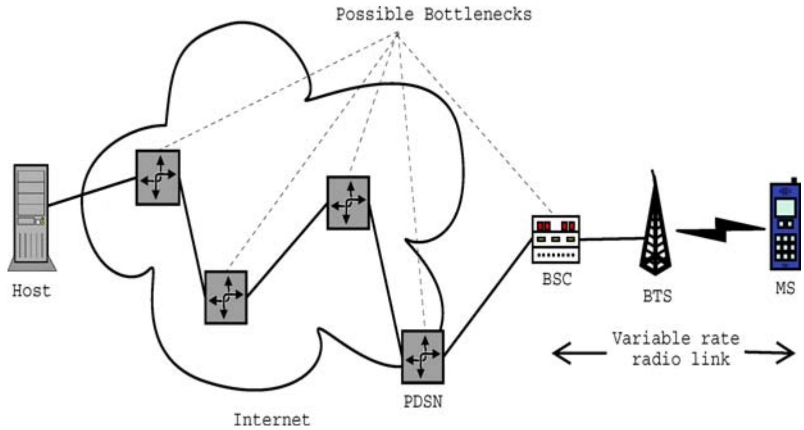


Figure 1. A network view of a mobile's TCP flow over an IP/cdma2000 [1] interconnection.

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.

Congestion control in 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 caused due to 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 it learns its lesson the hard way – multiple packets dropped at one of intermediate routers and a possible prolonged timeout afterwards.

Further, we show that the most widely used RED mechanism for network routers is unable to check the 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%.

2. Background

A typical buffering scheme based on the cdma2000 standard [3] is shown in Figure 2. Radio link rate variations are quite frequent in wireless networks [4]. 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 to the mobile stations for specific duration 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

Table 1. 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		
	b^s_{ij}	100 Mbps, for all i, j
	b^c	100 Mbps
	b^t_i	100 Mbps, for all i
	b^m_{ij}	R_{ij} kb/s, for all i, j
Delays	d^s_{ij}, d^m_{ij}, d^t_i, d^c	37.5 ms each
BSC's buffer	B_{cum}	Service rate[1] = $m \cdot n \cdot 38.4\text{kb/s}$ Droptail(cap. = 5 kB) RED(cap. = 300 kB, $\text{min}_{\text{th}} = 20\text{kB}$, $\text{max}_{\text{th}} = 60\text{kB}$).
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.

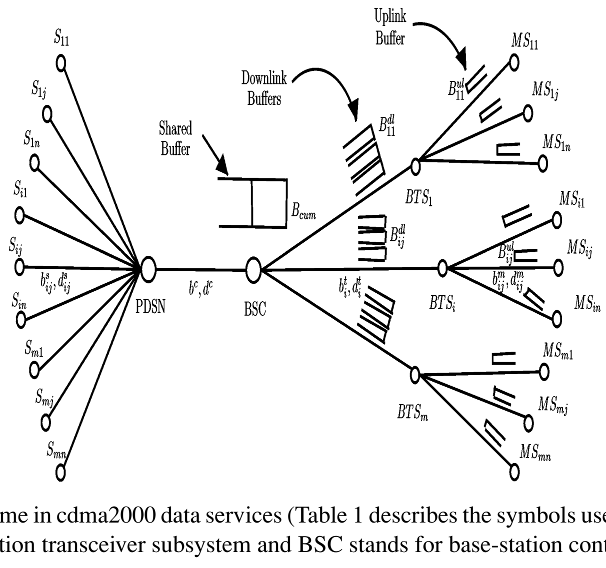


Figure 2. Buffering scheme in cdma2000 data services (Table 1 describes the symbols used). MS stands for mobile station, BTS for base-station transceiver subsystem and BSC stands for base-station controller.

allocations can be done in a variety of schemes, and one such scheme is *finite-burst* mode [11] (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), [7] 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 band-widths 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 agents 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 [11]. 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 [11] 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 [1] that aim at providing higher data rates to the level of 2 Mbps, 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 [10] on current ongoing research on wireless links considers a study of QM issues in shifting bottleneck scenarios as highly desirable. 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 [9] 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.

As an example to illustrate our point, Figure 3 shows the TCP traces (obtained using *ns*2 simulator [21]) for a mobile whose rate is switched from base FCH rate of 9.6 Kbps to 163.2 Kbps(FCH + 16X SCH) at 100s. The intermediate buffer is assumed to have a service rate of mere 40Kbps and a buffer capacity of 5 packets for this flow. The traces of dropping at this buffer are also shown. It is clear that sudden allocation of higher bandwidth leads to ACK compression and resultant prolonged timeout-based recovery.

Table 2. Comparison of link QM and Network QM

Link QM	Network QM
QM scheme for link buffers e.g. B_{ij}^{dl}, B_{ij}^{ul}.	Scheme for other network buffers that may overflow, like B_{cum} in Figure 2.
Stores packets for/from a mobile only.	Stores packets for/from many mobiles.
Usually handles just 1, at most 3-4 TCP flows.	Handles lots of TCP flows, several hundreds of them.
Very low statistical multi-plexing.	High statistical multiplex–ing.
Deterministic QM schemes like PDPC [9] preferable.	Probabilistic QM schemes like RED preferable, how-ever possibly with some modifications

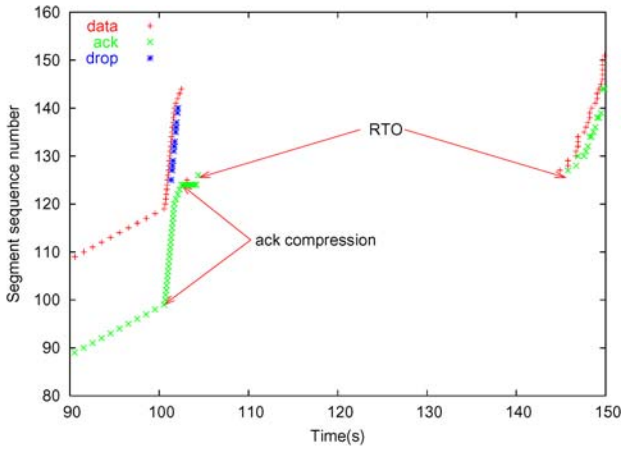


Figure 3. TCP trace for SCH allocation (RTO stands for retransmission timeout).

In this reference, it would be worth while to mention that the queue management(QM) problem of wireless domain is twofold – individual link QM and collective QM for other network nodes. The heart of this argument lies in the fact that not only the individual link buffers, $B_{ij}^{dl}s$, can overflow, instead situations may be there when other collective buffers like B_{cum} feeding these individual link buffers can also overflow. It has to be understood that these two problems are of entirely different natures and have to be tackled differently. Table 2 compares these two. Notably enough, some research efforts have been done towards the study of problem of link QM [9] 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 are tiny queues 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 instrumental in improving the performance. In brief, we would like to say that a unified study of QM techniques involving wired links together with wireless links is essential for developing an effective approach.

3. Modeling Losses Due to Bandwidth Change

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

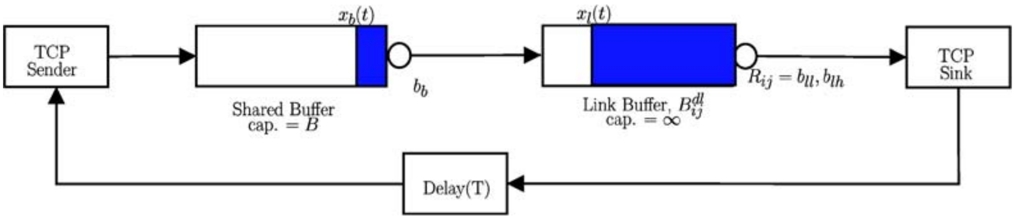


Figure 4. Model for losses due to rate change in a radio link.

as used in some previous works [13, 16, 17]. Figure 4 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 b_b, 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, $R_{ij} = b_{ll}$ corresponding to the base FCH rate, say 9.6 kb/s and the higher rate of b_{lh}, equal to sum of rates of an FCH and a SCH, say $R_{ij} = \mathrm{FCH}_{1X} + \mathrm{SCH}_{16X} = (9.6 + 153.6)\,kb/s = 163.2\,kb/s$. Shared buffers are normally designed for nominal loads and we are specifically interested in scenarios when b_b falls in between these two values (say a rate of mere $SCH_{4X} = 38.4$ kb/s for these values of b_{ll} and b_{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, $x_l(t)$ and $x_b(t)$, respectively and the round-trip propagation delay, T, lumped between the TCP sender and the sink.

To illustrate the impact of a single switching, let us assume that the mobile was operating with the lower link rate of b_{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 $x_l(t) = W - b_{ll} \cdot T$ and $x_b(t) = 0$. Now if at instant t_0, the link rate is switched to the higher value of b_{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 more traffic 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 b_{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 $t_{\text{start}} = t_0 + T$. 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 $t = t_0$, $\frac{dx_l(t)}{dt} = b_{ll} - b_{lh}$. The time required for link queue to become empty is $\frac{0-x_l(t_0)}{b_{ll}-b_{lh}} = \frac{W-b_{ll}\cdot T}{b_{lh}-b_{ll}}$. So, the governing condition for this case to happen is $T \geq \frac{W-b_{ll}\cdot T}{b_{lh}-b_{ll}}$, or $b_{lh} \geq \frac{W}{T}$. 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 b_{lh}. Thus,

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

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 b_{ll}, for $t_0 \leq t \leq (t_0 + T)$, afterwards it becomes b_{lh}, for $t_0 + T \leq t$ and while $x_l(t - T) \geq 0$. Consequently outflow rate of shared buffer is b_{ll} till $t < t_0 + T$ and b_b, afterwards. The draining rates of link buffer are, $\frac{dx_l}{dt} = b_{lh} - b_{ll}$ for $t_0 < t < t_0 + T$ and $(b_{lh} - b_b)$ afterwards while $x_l(t) \geq 0$. Thus at $t = t_0 + T$, the content of link buffer is, $x_l(t_0 + T) = x_l(t_0) - (b_{lh} - b_{ll}) \cdot T = W - b_{lh} \cdot T$. So the total time after t_0, required for the link buffer to run empty is $T + \frac{x_l(t_0+T)}{b_{lh}-b_b} = \frac{W - b_b \cdot T}{b_{lh}-b_b}$. Since there is a delay of T seconds between link buffer and TCP agents, shared buffer will stop receiving packets at high rate at,

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

Case 3: The leading edge of packet flow at higher rate reaches the shared buffer before the link buffer runs empty and the TCP_1 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 $\frac{dx_b}{dt}/ = b_{lh} - b_b$. It gets filled after another $\frac{B}{b_{lh}-b_b}$ seconds and the buffered packets in shared buffer at this instant take another $\frac{B}{b_b}$ seconds to clear the shared buffer after which the trail reaches the link buffer. The remaining content of link buffer at this instant is $W - b_{ll} \cdot T - (b_{lh} - b_{ll}) \cdot T - (b_{lh} - b_b) \cdot (\frac{B}{b_{lh}-b_b} + \frac{B}{b_b})$. This remaining data in the link buffer ahead of the trail behind first dropped segment drains at rate b_{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:

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

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,

$$t^{***}_{\text{end}} \leq t^{**}_{\text{end}} \Rightarrow b_{lh} \leq \frac{W}{T + \frac{B}{b_b}} \tag{4}$$

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

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

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 $t_{\text{start}} + \frac{B}{b_{lh} - b_b}$, no losses occur and afterwards losses occur at a rate $(b_{lh} - b_b)$ until time t_{end}. These concepts together with Equation (5) can be used to calculate the loss volume (L_V) that comes out as:

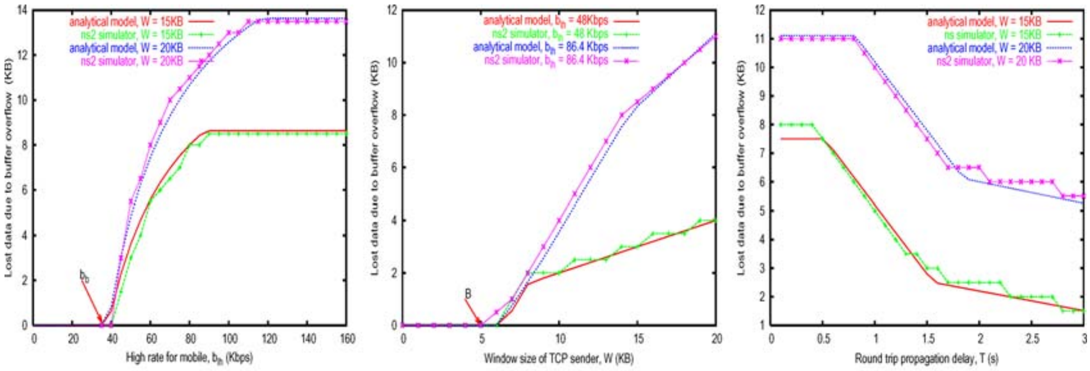
$$\begin{aligned} &L_V(b_{ll}, b_b, b_{lh}, W, B, T) \\ &\quad = \begin{cases} \dfrac{W \cdot (b_{lh} - b_b)}{b_{lh}} & \text{if } b_{lh} \leq \dfrac{W}{T + \frac{B}{b_b}} \\ \max[(W - B - b_b \cdot T), 0] & \text{if } \dfrac{W}{T + \frac{B}{b_b}} < b_{lh} < \dfrac{W}{T} \\ \max\left[\left(\dfrac{(W - b_{ll} \cdot T)(b_{lh} - b_b)}{(b_{lh} - b_{ll})} - B\right), 0\right] & \text{if } b_{lh} \geq \dfrac{W}{T} \end{cases} \end{aligned} \tag{6}$$

4. Simulations

For the purposes of simulations, we used an implementation of cdma2000's link layer protocol, RLP [4], in *ns*2 [21] simulator. This RLP module made available at [20] can be used to model the SCH allocations and wireless losses. In a companion report [20] 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 1.

4.1. 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



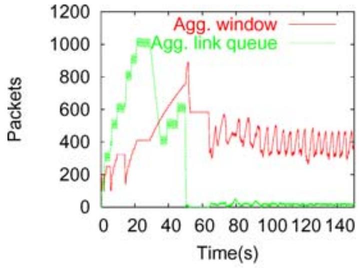
(a) $T = 0.3$s, $W = 15$ kB, 20 kB (b) $T = 0.3$s, $b_{lh} = 48$ kb/s, 86.4 kb/s (c) $b_{lh} = 86.4$ kb/s, $W = 15$ kB, 20 kB

Figure 5. Variation of losses due to bandwidth change with, (a) mobile's higher switched rate, b_{lh}, (b) TCP sender's window size, W, and, (c) round trip propagation delay, T. (Other parameters are kept constant at $b_{ll} = 9.6$ kb/s, $b_b = 38.4$ kb/s, $B = 5$ kB, packet size for *ns*2 simulations = 500 bytes).

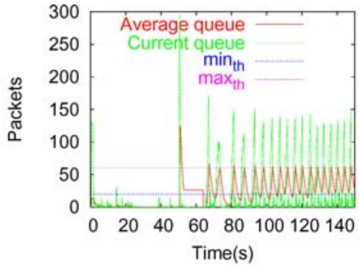
for packets destined for the mobile. The parameters are shown in Table 1. Figure 5 shows the comparison between the two models. It can be clearly seen that, except for the slight inaccuracy due to continuous flow approximation, the two models are in close agreement. Figure 5(a) shows that, for fixed W, T, b_b, b_{ll} and B, packet loss volume, L_V, first increases and then remains nearly constant. The limiting value of losses for increasing b_{lh} will be, $\lim_{b_{lh}} \rightarrow \infty L_V(b_{ll}, b_b, b_{lh}, W, B, T) = (W - B - b_{ll} \cdot T)$. So, for very sharp bandwidth swing, the losses can be as high as total buffered data at low bandwidth, $W - b_{ll}T$, minus the buffer space available at shared buffer, B. Figure 5(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 5(c), which shows that, keeping other parameters fixed, losses decrease with increased propagation delay.

4.2. 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 6 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 16 × (153.6 kb/s) SCH allocation at 50s. This means a change in aggregate link bandwidth from 0.96Mbps to 16.32 Mbps. 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



(a) Aggregate window, aggregate link queue.



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

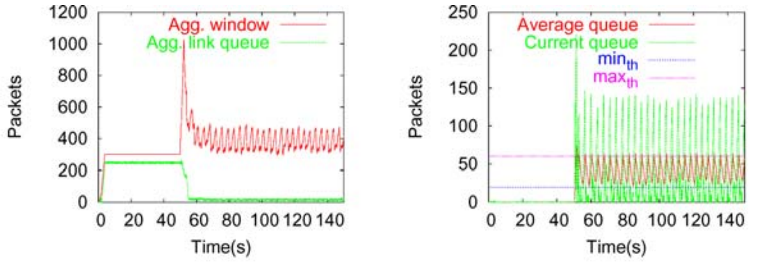
Figure 6. Plots for simultaneous *16X* SCH allocation to all 100 mobiles at 50 s.

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 [7], 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 50 s, 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 9.6 kb/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 6(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 [7].

4.3. Comparison with a Scenario with Variable Number of Users

To establish our claim of bandwidth changes as a severe source of congestion, we benchmark the variable bandwidth scenario against the cases when number of users changes as in [8], keeping the aggregate link bandwidth and shared buffer's service rate constant. To simulate an aggregate bandwidth swing from 0.96 Mbps to 16.32 Mbps, we change the number of active users from 6 to 100 at 50s, keeping the link rate for all the mobiles at 163.2 Kbps throughout. The plots for this scenario are shown in Figure 7. 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



(a) Aggregate window, aggregate link queue.

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

Figure 7. Plots for more users joining simultaneously at 50 s.

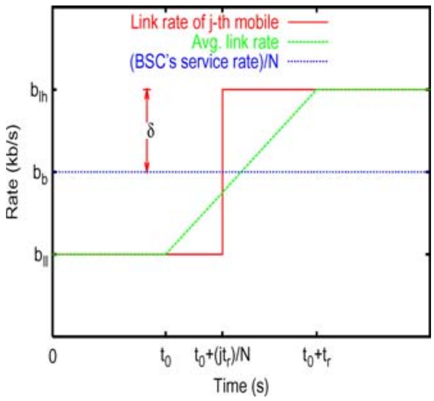


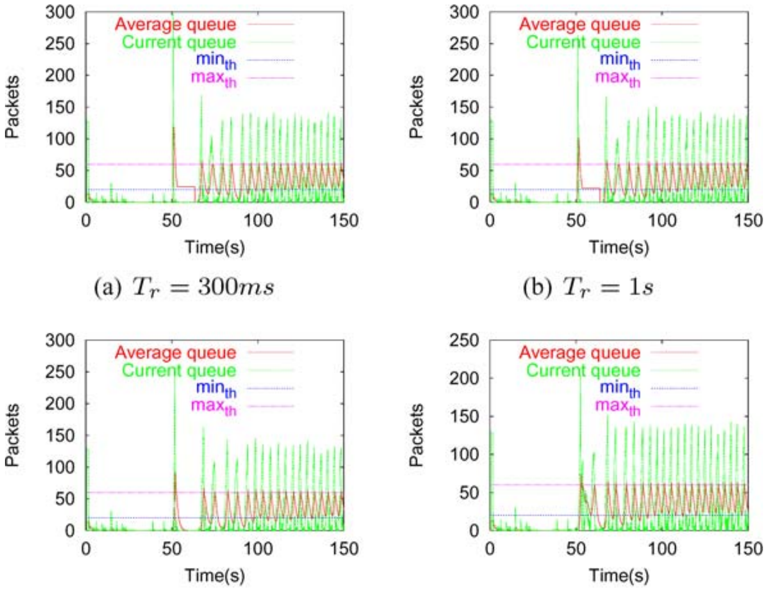
Figure 8. Scheme for increasing the radio links' rates.

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 6, it is very clear that RED's random dropping mechanism is capable to handle congestion due to change in number of users.

4.4. 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 allocation are usually are *on-demand* 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 as shown in Figure 8. In this configuration, 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/N, where N is total number of mobiles ($N = (m \cdot n)$). This scheme allows for aggregate output links' bandwidth, as seen by BSC, to change from 0.96Mbps to 16.32Mbps over T_r seconds if each of the mobile's rate is changed from 9.6 Kbps to 163.2 Kbps.

Figure 9 shows the queueing behavior for BSC's buffer for an up-switching of the system beginning at 50 s. The round-trip propagation delay is 300 ms, as before. We plot for T_r values of 0.3 ms, 1 s, 2 s and 5 s. It can be clearly seen that in all the first three cases, severe under-



(a) $T_r = 300ms$ (b) $T_r = 1s$

(c) $T_r = 2s$ (d) $T_r = 5s$

Figure 9. Impact of rise time of aggregate bandwidth on queue behavior at shared buffer.

utilization of system resources will occur as all the TCP senders have suffered multiple losses and are waiting to timeout for a long duration. Even for $T_r = 5$ s, 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, such changes in aggregate bandwidths lead to empty queues at shared buffers and system under-utilization.

4.5. 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[11]. This mode of operation involves assignment of an SCH of rate 153.6 Kbps for duration T_B seconds at inter-allocation spacing of T_D seconds. We look at the performance of RED for various values of burst-delay durations in this finite-burst mode of operation for the mobiles. To analyze the worst-case behavior, 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. First, we present a performance metric that is used for analyzing the load scenario that captures the under-utilization of the whole wireless system. A good metric for QM behavior could have been the aggregate throughput for the mobiles, but reductions in throughputs can not be benchmarked against a standard as in [11]. There, the authors measured througputs against average channel rate. However in our study where the wireless channels are not the only bottleneck at all times, benchmarking against average channel rate makes little sense. Utilization of BSC's shared buffer could have been the other option but since BSC's queue is also not the bottleneck at all the times and when mobiles' aggregate rate is less than BSC's output

Table 3. Percentage dead-periods (100% underutilization definition) versus burst-delay durations

	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	18.84	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

rate, the output queue of BSC remaining empty is very natural. Essentially we would like to get a metric that simultaneously captures underutilization of both the wireless channels and BSC's output buffer. Since these are the only two possible locations of bottleneck in scenario we are considering, it suffices to analyze the utilization of these two. However, instead of looking at utilization of these two, we instead look at periods during which both are not utilized. These periods are representative of the underutilization of the whole system as both the potential bottlenecks are not being utilized during these periods. We call these times as *dead-periods*.

So, 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 output queue on sudden rate increase. During these periods, all the link queues as well as the BSC's output queue remain empty meaning enormous 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 3.

In Figure 10, a looser definition of *dead-period* is used wherein fractional time when the BSC's output queue is empty and 90% of link queues are empty. Clearly a scenario with BSC's queue being empty and only 10 percent of links in use is also a strong indication of underutilization of total available resources. As can be seen in Table 3, 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 pump in data packets very fast to cause any havoc. However, at larger values, dead-periods are longer because now TCP senses this bandwidth change and pumps 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 simulation time, this effect is

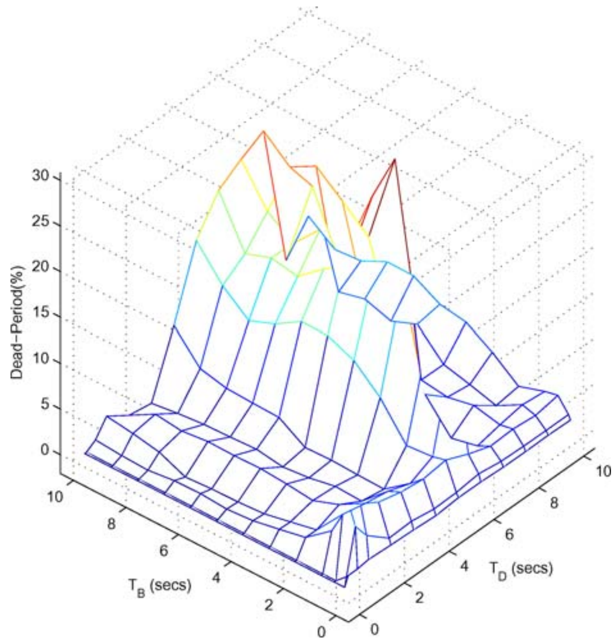


Figure 10. Fractional dead-period (90% underutilization definition) for various values of burst-delay duration.

less pronounced.

Also from Figure 10, it can be clearly seen that for certain values, fractional dead-period can be as large as nearly 25 percent. This is a waste of resources as during one-fourth 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 75%.

After having clearly identified the queue management problem that comes up on sudden allocation of high bandwidth. It has been shown 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 roundtrip 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 and very slow rise times, RED performs well.

Nonetheless the significance of these short-term under-utilizations can not be ruled out. The reasons for it are 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 impact on long-term aggregate throughput for mobiles in a typical load scenario.

4.6. Threshold for Stable Operation of RED

Earlier it was shown that rise time has major impacts on the performance after bandwidth changes. The other parameters which impact the performance that are introduced in this section are the fractional rate overload, $\delta (= \frac{b_{\text{th}} - b_b}{b_b}$, refer Figure 8), and queue overload in link buffers prior to increase in aggregate link bandwidth, β. The latter can be quantified as ratio of aggregate link queue length prior to onset of channel allocations to the queue limit of BSC's shared input buffer. We discuss each of them now in detail.

4.6.1. *Queueing Overload*

Channel allocations are mostly based on data backlog in link queues. It is important to note that if excessively larger size of queue is allowed to build up in link buffer, it will result in longer duration of *ack* compression and greater losses. This factor is dependent on channel allocation policy, e.g. the inter-burst separation in finite burst mode [11]. A larger spacing between two consecutive SCH allocations results in a large queue in the link buffer and it can be alleviated by means of more frequent channel allocations based on fast signaling. However, signalling can not be done at very high frequency due to system limitations. Even though cdma2000 systems support faster signaling than IS-95 and IS-95B by means of mini-messages and 5ms frames, still signalling can not be fast enough to achieve any desired level of parity between aggregate link queues and BSC's *limit*. Since the present practice is to set BSC's buffer size based on delay-bandwidth product (DBP) and does not account for burstiness due to channel allocations, this is often insufficient to absorb the sudden transfer of workload from links to BSC's input buffer after channel allocation. Still, it is possible to choose a suitable design so that the mismatch of queueing workload between shared buffer and link queues is not too large. We identify the thresholds for stable operation of traditional DBP based sizing of shared buffer, and beyond the threshold, buffers of sizes larger than those governed by DBP principle are necessary.

4.6.2. *Rate Overload*

If the aggregate rate at which links can drain is greater than the service rate of input buffer of BSC, this leads to transfer of entire queued workload in links to the shared input buffer. If the value of aggregate bandwidth after channel allocations is too larger than BSC's service rate, b_{ll}, it leads to faster filling of shared buffer and greater losses afterwards. As seen before, by means of continuous flow approximations, for very sharp bandwidth switching, amount of data lost can be as high as the difference of queued workload in link queues prior to channel allocations, $W - b_{ll} \cdot T$ (W is aggregate window size of senders and T is round trip propagation delay) and the buffer space available at shared buffer, B. Rate overload can be mitigated by reducing the mismatch between shared buffer's service rate and aggregate link rate, but is often limited by processing powers available at BSC (considering that lots of processing of IP packets into RLP frames and vice versa needs to be done at the BSC).

In determining the thresholds for stable operation of queueing behavior we use the following rule. The threshold for a variable, keeping other parameters constant, is the minimum value of the variable for which either of the following two things occur: (1) the queue length of shared buffer just hits its maximum value equal to the buffer space, *limit*, or (2) the EWMA (Exponential Weighted Moving Average) averaged queue length of RED algorithm just hits the maximum threshold parameter, $\max_{\text{th}}$. We argue that these two conditions define the threshold by observing that a stable queue management would be one that preserves its probabilistic dropping to maintain good statistical multiplexing even during congestion. Since, conditions (1) and (2) are the ones that trigger undesirable droptail behavior, the prime target of a stable design would involve avoiding these two as much as possible. Although some researchers have proposed the gentle mode of RED beyond max_{th}, still we would like to maintain that average queues greater than max_{th} are the regions that should be avoided and aim at operating the average queue in a linear probabilistic dropping mode between min_{th} and max_{th}.

We simulated the congestion scenario for a variety of parameter settings. Figure 11 shows the impact of varying the rate overload, δ, keeping the values of queueing overload constant at $\beta = 30\%$ and rise time at $t_r = 1$ s. It can be clearly seen that transfer of workload from links to shared buffer is smooth for lower values of δ and is very sharp at higher values leading to prolonged timeouts and drops. For each of the overloads, the number of packets drops in the 5 seconds after the channel allocation are also shown. It is during this period that most of drops are only due to channel allocations and not because of long term probabilistic dropping of RED mechanism.

We now proceed on to identifying the threshold for stable operation for this setup. A closer examination of the Figure reveals that the threshold condition (1) occurs for some value between 40% and 45% and condition (2) first occurs for some value between 15% and 20%. Clearly, in this case, average queue hits max_{th} value before queue size reaches the buffer limit and hence condition (2) governs the threshold value of δ. But this is not always true as, for some simulations, it was observed that condition (1) is the determining factor. After running several simulations in the range 15% to 20%, it was observed that 18.5% is the minimum value at which condition (2) just occurs and hence this seems to be the threshold value for stable operation. This may seem a too low value for stable operation, but this is expected for a small value of rise time of 1 s.

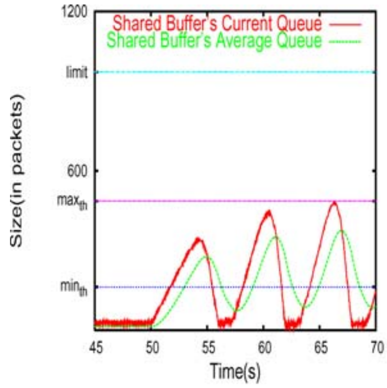
We generalize this concept of obtaining the values of rate overload threshold, δ^*, for various values of queue overloads and rise times as shown in Figure 12. The system operates smoothly for values of overload rates below the threshold and beyond that performance will be degraded due to channel allocation induced congestion. A lower value of queueing overload and a higher value of rise time yield a broader range of operation. Thus, by means of simulations, range of stable operation can easily be found and used to enhance the performance. If the setup parameters can not be compromised for strict adherence to stable operation, larger buffers are the only option to mitigate the problem. This might come as a little surprise because larger buffers would mean greater queueing delays, especially since some latest work [19] claim that even buffer sizes much smaller than DBP yield same performance, but our suggestions are based on specific needs of wireless networks to absorb the excessive burstiness due to rate variations.

5. Possible Methods to Improve Performance

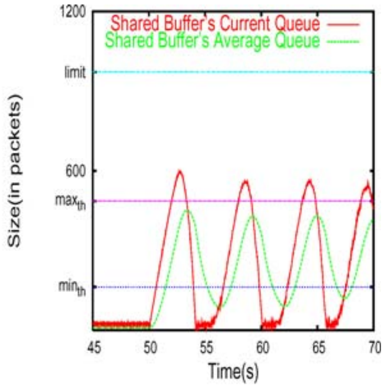
Now after having understood the queue management problem due to bandwidth oscillations, in this section, we try to explore the solution space for the problem of queue management in cellular networks. We examine the various levels at which this issue can be resolved. In doing so, we also recognize the possible pitfalls in terms of applicability of these approaches.

5.1. Modifications to Queue Management

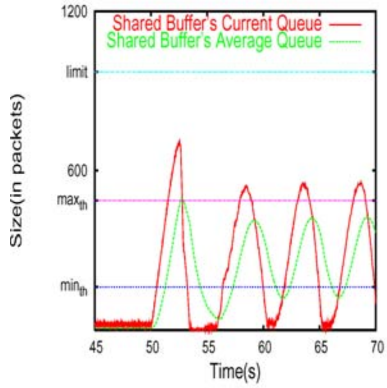
If somehow RED algorithm is able to track the sharp enormous growth in queue size on sudden allocation of bandwidth and distinguish it from usual transient spikes and additionally inform



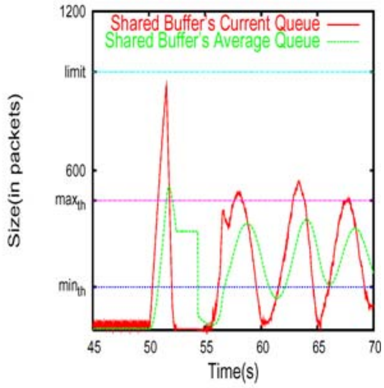
(a) $\delta = 5\%$, drops = 12



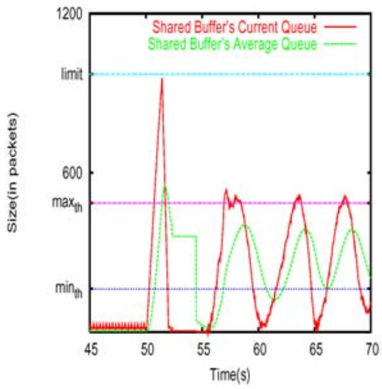
(b) $\delta = 15\%$, drops = 33



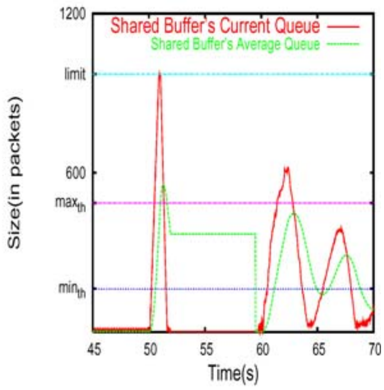
(c) $\delta = 20\%$, drops = 55



(d) $\delta = 40\%$, drops = 167



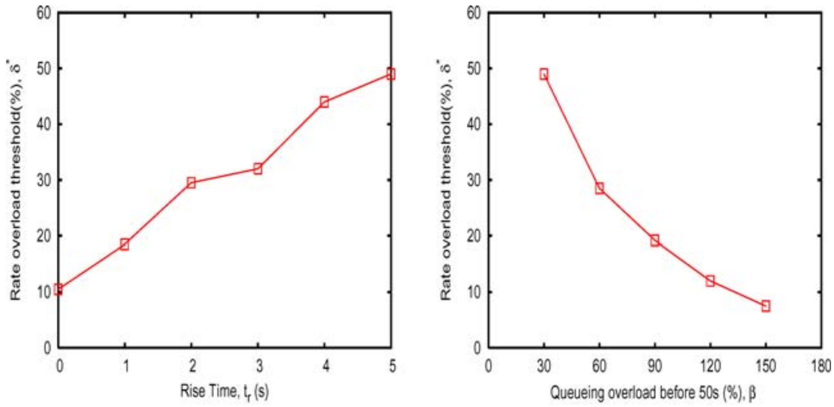
(e) $\delta = 45\%$, drops = 192



(f) $\delta = 75\%$, drops = 291

Figure 11. Queueing behavior in shared buffer and link queues with varying levels of overload. For each of the figures, the number of packets dropped in 5 seconds after the channel allocations (i.e. between 50–55 s) are also shown. The values of t_r and β are constant at 1s and 30% respectively.

the senders without significant dropping in a droptail fashion, it could mitigate this problem. However, these are competing demands and it might be hard to find the decision boundary for them. On one hand, filtering out transients helps in allowing short-lived bursts but on the other hand this leads to sluggish response of router to bandwidth-induced sudden incipient congestion leading to loss of many packets in a droptail-like fashion. These contrasting



(a) Constant queueing overload, β = 30%

(b) Constant rise time, t_r = 5s

Figure 12. Threshold for stable operation with default settings.

demands make it difficult to understand how much a low-pass filter(LPF) behavior is desirable in calculating average queue size. However, if there were a way by which BSC had an idea of current aggregate bandwidth, it could make decision based on them so that it knows when it needs to react fast and when not to. It can alternatively allow for larger queue sizes when aggregate bandwidth is high and vice versa as our ultimate QM target is bounding the packet delays and not the queue sizes. In either of the cases even if a BSC can adapt its QM discipline to aggregate bandwidth changes, still other intermediate routers have no way of determining the changes in aggregate bandwidth, and problem for other network nodes still persists.

Another approach would be to regulate the flow of returning ACKs at the intermediate shared buffer in some fashion so that TCP senders do not suddenly start pushing data at an excessive rate on seeing ACKs returning at a very fast rate due to high bandwidth allocation. This means that on sudden allocation, the BSC delays the returning ACKs for a while so that RED routers have time to respond to it and multiple drops of packets do not happen. In the long run, ACKs will be forwarded without any delay, so that we are not unnecessarily increasing the RTTs. Basically in this scheme, on sudden allocation, we simply delay the ACKs for a while so that ACK compression does not happen and a sharp bandwidth change reflects in a smooth change in TCP sender's sending rate. Such an approach is presented in [15]. Therein the authors use a ACK-regulator mechanism by which TCP is adjusted for variable bandwidth, variable delay scenarios.

5.2. Fixing TCP

Another way of looking at the problem is to somehow fix the TCP in some manner so that its window mechanism responds smoothly to fastly arriving ACKs. In the long-run it should work with the fast rate but for periods just after allocation it should not change its sending rate abruptly. Such an approach would be difficult in practice for two reasons: first all the TCP senders can not be enforced to conform to this new approach, and second this assumes that rate of returning ACKs is totally dependent on wireless conditions. By not allowing TCP to increase its sending rate sharply for many ACKs in a short duration, we might lead to fewer packets at bottleneck resulting in underutilization.

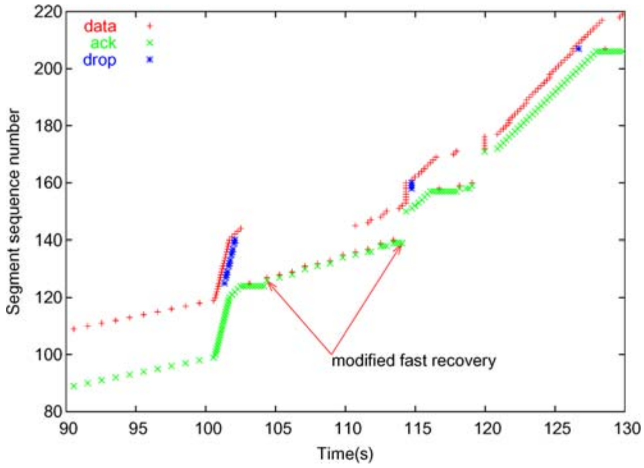


Figure 13. Enhancement offered by Newreno TCP's fast recovery algorithm as against Reno TCP in Figure 3.

The other working direction is to fasten the recovery process after multiple packet losses. Fortunately, a solution to this exists in form of Newreno TCP's modified fast recovery algorithm to react to multiple packet losses. Basically, a Newreno TCP sender retransmits packets based on partial ACKs that arrive after detection of losses and subsequent partial recovery. This mechanism is very robust in handling multiple losses and resumes the normal flow of data in a shorter span of time. Figure 13 shows that the recovery time after multiple losses has been reduced to a mere 16,s which, in case of a Reno sender in Figure 3, was 50 s due to the latter's dependence on timeout mechanism after multiple losses. Not all the TCP implementation in existing Internet use Newreno modification, and our results supply another reason for enforcing the vendors to conform to Newreno modifications.

5.3. Scheduling Bandwidth Allocation/Deallocation

Another approach that will solve most of the problems but lacks greater applicability is scheduling the allocation/deallocation of higher-bandwidth supplemental channels. If the aggregate links' bandwidth remains roughly the same, much of the QM problems will not occur. One way of doing this is by using some scheduling mechanism at BSC to do the allocation/dealloaction uniformly spread over the time. For instance, in our setups with burst-delay durations of T_B, T_D, whereby we give SCH to all mobiles at say t_0, remove it at $t_0 + T_B$ and again allocate it at $t_0 + (T_B + T_D)$, which results in oscillation of aggregate bandwidth between (mnb_{ll}) and (mnb_{lh}). If the BSC allotted these bandwidths in a sequential fashion like $k - th$ mobile is assigned SCH at time $t_0 + ((k-1)/(mn))(T_B + T_D)$, removed at $t_0 + T_B + ((k-1)/(mn))(T_B + T_D)$ and reallocated at $t_0 + (T_B + T_D) + ((k-1)/(mn))(T_B + T_D)$ for $1 \leq k \leq (mn)$, such an approach, while keeping the burst-delay durations of T_B and T_D for a mobile constant, also keeps the cumulative bandwidth at all times tied down to a value of $mn(b_{ll}(T_D/(T_B + T_D)) + b_{lh}(T_B/(T_B + T_D)))$. Under this scheme, individual bandwidths rise and fall abruptly but aggregate bandwidth remains constant and it can be shown that this poses no problem in terms of queue management. However since channel assignment is done in a completely unregulated fashion, such a smooth behavior will be difficult to be enforced at BSC. Instead, what could be done is to implement an algorithm at BSC by which, based on a value of maximum tolerable rise rate of aggregate bandwidth, other mobiles requesting additional bandwidth, when aggregate bandwidth rise rate has hit its maximum, are denied the SCH for a while so that queue management can be done smoothly. Such an approach requires only an estimate of current aggregate bandwidth at BSC and pre-determined thresholds of rise-rate for SCH denial at current juncture and of delay period after which this SCH can be allocated. This can be an effective approach if BSC is completely aware of current status of channels allocated to all mobiles under it.

6. 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 and possible working directions to stimulate further research to develop a solution to this problem.

Acknowledgements

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

References

1. 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*, Vol. 38, No. 7, July, pp. 70–77 2000.
2. "Proposed 1XTREME Physical Layer Delta Specification", *Source: Nokia, Motorola, LSI Logic, Texas Instruments and Dot Wireless, Contribution to 3GPP2*, August 2000.
3. TIA/EIA/IS-2000.1, "Introduction to cdma2000 Standards for Spread Spectrum Systems", March 1999.
4. TIA/EIA/IS-707-A-1, "Data Services Options for Spread Spectrum Systems – Radio Link Protocol Type 3", December 1999.
5. 3GPP TS 03.60: Digital Cellular Telecommunications System (Phase 2+), "General Packet Radio Service (GPRS): Service Description, Stage 2".
6. S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, *Internet Engineering Task Force*, April 1999.
7. S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *ACM Trans. on Networking(ToN)*, Vol. 1 No.4, pp. 397–413 August 1993.
8. S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", *Under Submission*, August 1, 2001.
9. 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)*, Mar. 2003.
10. A. Gurtov and S. Floyd, "Modeling Wireless Links for Transport Protocols", *in ACM Comp. Commm. Review(CCR)*, to appear.
11. M. Yavuz and F. Khafizov, "TCP Over Wireless Links with Variable Bandwidth", *In Proc. of Vehicular Technology Conference(VTC'02)*, Vol. 3, pp Sept. 2002. 1322–1327.
12. M. Yavuz and F. Khafizov,"Running TCP over IS-2000", *In Proc. of Intl. Conf. on Comm.(ICC'02)*, Vol. 2, pp. 3444–3448 May 2002.
13. Y. Wardi and B. Melamed, "Loss Volume in Continuous Flow Models: Fast Simulation and Sensitivity Snaly-

sis", *In Proc. of 8th IEEE Mediterranean Conference on Control and Automation (MED-2000)*, Patras, Greece, July 17–19, 2000.
14. B. Melamed. S. Pan, and Y. Wardi, "Hybrid Discrete-Continuous Flow Simulation", In *Proc. of the SPIE Intl. Symp. on Information Technologies and Communication (ITCOM'01)*, Aug. 2001.
15. M. Chan and R. Ramjee, "TCP/IP Performance Over 3G Wireless Links with Rate and Delay Variation", *In Proc of 8th Annual International Conference on Mobile Computing and Networking(Mobicom'02)*, Atlanta, Georgia, USA, 2002.
16. B. Liu, D. Figueiredo, Y. Guo, J. Kurose, and D. Towsley, "A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet-Level Simulation", *In Proc. of IEEE Infocom* 2001.
17. 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'OO*, Stockholm, Sweden, September 2000.
18. W.R. Stevens, *TCP/IP Illustrated*, Addison Wesley, 1993.
19. Guido Appenzeller, Isaac Keslassy, and Nick McKeown, "Sizing Router Buffers," in *Proc. of ACM Sigcomm*, August 2004.
20. Online. *http://www.sce.carleton.ca/vpaliwal/dist/*.
21. Online. *http://www.isi.edu/nsnam/ns/*.

Vikas Paliwal graduated with a Bachelor of Technology degree in Electrical Engineering from IIT (Indian Institute of Technology), Kanpur, India and an M.S. in Systems and Computer Engineering from Carleton University, Ottawa, Canada. He has five years of research and work experience in telecommunications area in a variety of work environments across the globe. During his under-graduation he worked on a research project in micro-electronics area with IMS Research Labs, Hanover, Germany, and developed some innovative algorithms for circuit schematics and layouts. After graduating from IIT, Vikas was the core Engineering and Marketing Consultant for GMG Telecom Consultants India Limited, New Delhi, India, and helped GMG in developing a level of technical expertise in the areas of security systems that resulted in key research projects from firms such as ADT and Honeywell. Later, as part of his Masters research at Carleton University, Ottawa, Canada, Vikas worked on performance analysis of CDMA data networks and queue management techniques that resulted in five research papers in various academic journals/conferences. His Masters research project also attracted the interest of Nortel Networks, Ottawa, Canada, and helped in starting joint research projects in the area of wireless systems simulation, which was of particular interest to Nortel. Vikas then worked with Solana Networks, Ottawa, on network topology detection and on products that improve the performance of mission-critical IP networks and wireless LANs. This was followed by a year of work with Qualcomm, UK, on modem software design and implementation for Qualcomm's 3G chipsets. Currently he works with Qualcomm Inc., San Diego, CA, on physical layer of WCDMA and HSxPA systems. At Qualcomm, he continues his research activities in advanced wireless systems design and implementation by actively participating in various technical forums and conducting several training/discussion sessions on UMTS systems.

Ioannis Lambadaris was born in Thessaloniki, Greece. He received a diploma in Electrical Engineering from the Polytechnic School of the Aristotle University of Thessaloniki in 1984. He was a recipient at a Fulbright Fellowship (1984–1985) for graduate studies in USA. He received a M.Sc. degree in Engineering from Brown University, Providence, RI, USA in 1985 and a Ph.D. degree in Electrical Engineering from the University of Maryland, College Park, MD, USA in 1991. He was employed as a research associate at Concordia University, Montreal, Quebec, Canada, in 1991–1992. Between September 1992 and July 1997 he has been an Assistant Professor in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Ontario, Canada. Currently he is an associate professor in the same department and a member of IEEE. While at Carleton he received the Premier %92 s Research Excellence Award (2000), and the Carleton University Research Excellence Award (2000–2001) for his research achievements in the area of modeling and performance analysis of computer networks. Professor Lambadaris' interests lie in the area of applied stochastic processes, stochastic control, queuing theory and their application for modeling and performance analysis of computer communication networks. His current research concentrates on quality of service (QoS) control for IP networks, resource allocation in optical networks, and optimal routing and flow control in ad-hoc wireless systems.

Biswajit, a co-founder of Solana Networks, has15 years of experience in the area of data communication. He has worked at Nortel Networks, Bell Northern Research, Tropic Networks and Belair Networks. The primary focus of his work has been in the area of IP networking protocols and architecture. As an Independent Contributor at Nortel Networks, he spearheaded various advance technology programs and R&D projects in IP networking. He worked as a Network Architect with the MPLS team and as a member of the architecture team at Tropic Networks to address complex optical networking issues. He worked as a System Architect on WLAN mesh networking products at BelAir Networks. Dr. Nandy has a PhD in Electrical and Computer Engineering from the University of Waterloo and has been an Adjunct Research Professor with Systems and Computer Engineering at Carleton University since 2001. He holds an M.Tech in Electrical Engineering from Indian Institute of Technology, Kanpur and a B.E. in Electronics and Telecommunication Engineering from Jadavpur University, Calcutta. He has authored and presented over 25 research papers in International conferences and holds 12 US patents. He has actively participated in various Work Groups in IETF and author of several IETF proposals in the form of Internet Drafts and RFC.