A TCP-Friendly Congestion Control Algorithm for 1XEV-DV Forward Link Packet Data

Patrick Hosein
Ericsson Wireless Communications, Inc.
6455 Lusk Blvd, San Diego, CA 92121, USA
Phone: 858 332-6405, Fax: 858 332-7468, Email: hosein@ericsson.com

Abstract—The 1XEV-DV 3G Standard is designed to support high-speed data services. It is also expected to provide Quality of Service (QoS) guarantees for these high speed data connections. On the radio interface, voice and data connections are code multiplexed. However, it is also possible to time share multiple data users on a single forward link data channel. An important aspect of this option is the scheduler used to serve users on this shared, slotted forward link data channel. If QoS guarantees are provided to these users then the scheduler must ensure that these are met at all times. Proper admission and congestion control algorithms are needed to ensure that accepted voice and data users are served at their guaranteed QoS levels. In this paper we present a congestion control algorithm for the air interface that is aware of its effects on the transport layer protocol.

I. INTRODUCTION

1XEV-DV is being developed as a 3G solution for mixed voice and high-speed data services. It will also provide QoS guarantees for these data connections. On the forward link, voice and data connections are code multiplexed. However, the Forward Packet Data Channel (F-PDCH), can be used to time share multiple data users. The F-PDCH is divided into slots and data frames are scheduled over these time slots (see [5] for details). If, in addition, QoS guarantees are provided to these connections then the scheduler must ensure that these are met for all users. An admission control algorithm is needed to ensure that accepted voice and data users can be served at their guaranteed QoS levels. However, due to the dynamic resource demands of both voice and data users and the stochastic nature of the bandwidth available to each user (because of the fluctuations in their radio conditions due to fading, interference and noise), a congestion control algorithm is also needed to properly allocate resources during periods of congestion until the admission control algorithm can act to reduce the offered load. If we assume that voice users are given higher priority than data users then during these overload periods the performance of the data users will degrade. This performance degradation must be controlled in such a way to maintain sufficiently high throughput at the application layer and avoid congestion collapse.

In this paper we present a congestion control algorithm that is aware of the effects of congestion on the upper layers (in particular the TCP layer). During overload periods, the RLP (Radio Link Protocol) buffers of the data connections will tend to overflow due to insufficient radio capacity. This in turn results in lost IP packets which in turn causes TCP to back-off which can lead to a drastic reduction in TCP throughput. If this is allowed to happen to all data connections then the congestion period (which may be short) results in a drastic reduction in throughput at the application layer. Furthermore, the TCP slow start mechanism will result in the inefficient utilization of the radio link following this congestion period. The net result is poor performance of the data connections (i.e. cycling between overload and light loading).

A similar phenomenon occurs in IP routers within wire-line networks. During congestion, the router buffer overflows and several TCP connections simultaneously reduce their congestion windows resulting in a bursty, inefficient data transportation service. Several algorithms, generically called Adaptive Queue Management (AQM) algorithms, have been proposed for reducing this effect. The most widely known of these is Random Early Detection (RED). In the case of RED, IP packets are randomly discarded when the router buffer occupancy reaches a sufficiently high level (see [1]). These random discards cause a subset of TCP connections to reduce their congestion windows which results in a reduction in the traffic load. In this way the reduction in traffic is small enough to maintain high link utilization (i.e., non-zero packet buffer occupancies) resulting in better total TCP throughput.

One can attempt to use the RED approach for the problem being addressed in this paper. However, there are several differences that make the RED approach ineffective, (a) an IP packet typically consists of several RLP frames and hence even if each connection loses a single RLP frame this results in an IP packet loss per connection and (b) each connection has a separate RLP buffer which means that random packet discards will have to be coordinated by a central process to ensure the appropriate number of packets are being discarded. Furthermore, in the router scenario, all TCP connections were treated equally and so random discards were appropriate. In our case, connections may be distinguished based on (a) the User Class of the connection owner, (b) the QoS demands of the application being run on the connection and (c) the radio resource requirements of the connection (e.g., more power is needed to support users far from the Base Station (BS) antenna). Therefore it is preferable to use this information in making IP packet discard decisions. In the next section we outline the scheduling framework, then we discuss the effects that different schedulers have on TCP congestion control then present the algorithm.
II. THE 1xEV-DV F-PDCH SCHEDULING FRAMEWORK

In 1xEV-DV (see [5] for details), the Forward Packet Data Channel (F-PDCH) is time shared among several users. This can be done on a Round-Robin basis or more sophisticated scheduling algorithms can be used. This shared channel uses a slotted transmission system with a slot size of 1.25 ms. On a periodic basis (every 1.25 ms), each Access Terminal (AT) informs the Base Station of the Carrier to Interference (C/I) level of its received signal. Whenever a time slot is free, the Base Station chooses an AT from among the ATs that have data for transmission on the Forward Link. The Base Station then transmits a single physical layer frame to the chosen AT. However, due to imperfect radio conditions, the frame may need to be repeated over multiple time slots. The repetition rate and the number of data bits that are inserted into the frame are determined by the estimated C/I value of the chosen AT as well as the power available to the F-PDCH. The choice of the AT (decided by the scheduling algorithm) can be based on various factors. In the case of 1xEV-DV, the supportable transmission rate (determined by the AT’s C/I ratio and the available F-PDCH power) as well as the average throughput achieved so far by the AT are used to make the determination. Once transmission to the AT is complete then the Base Station picks the next AT and repeats. NAKs are used to determine correct delivery of frames to the AT. If a frame is received in error then it is re-transmitted (up to two additional times). The Proportional Fair (PF) scheduling algorithm is typically used for 1xEV-DV. However other algorithms, such as the Barrier function algorithm (see [2]) may also be used.

A. Formulation of the Scheduling Problem

Kelly [3] provides a formulation of this shared resource allocation problem as follows:

\[
\begin{align*}
\text{maximize} & \quad F(\vec{r}) = \sum_{i=1}^{k} U_i(r_i) \\
\text{subject to} & \quad \sum_{i=1}^{k} r_i < C \\
& \quad \text{over } r_i \geq 0, \ 1 \leq i \leq k.
\end{align*}
\]

where,

- \(k\) = the number of active users competing for the channel,
- \(r_i\) = the average throughput of user \(i\),
- \(C\) = the channel capacity,
- \(U_i(r_i)\) = the utility function of user \(i\).

If we assume the utility function of each user is strictly concave and differentiable, then the same also holds for the objective function \(F\). Since the feasible region is compact then an optimal solution exists, is unique and can be found by Lagrangian Methods. The optimal set of throughputs, which we denote by \(\{r^*_i\}\), has the property that \(U_i'(r^*_i)\) is the same for all users. Note that this optimal point can be computed explicitly but cannot be instantly achieved because user throughputs are indirectly varied by serving users. The maximum gradient ascent algorithm can be applied to this problem by always serving the user for which the corresponding objective function gradient is maximum.

Assume that user throughputs are computed as follows:

\[
\begin{align*}
r_i(n+1) &= \left\{ \begin{array}{ll} 
(1 - \frac{1}{\tau})r_i(n) + \frac{d_i(n)}{\tau} & \text{if } i \text{ served in slot } n, \\
(1 - \frac{1}{\tau})r_i(n) & \text{otherwise}
\end{array} \right.
\end{align*}
\]

where \(d_i(n)\) is the bit rate of user \(i\) if served during the \(n\)th period, \(\tau\) is the time constant of the smoothing filter and \(r_i(n)\) denotes the average throughput that was previously computed.

Denote the gradient of the utility function of user \(i\) as follows:

\[
\begin{align*}
\vec{r}_i(n) &= \text{gradient of the objective function in the direction of serving user } j.
\end{align*}
\]

The resulting scheduler picks the user such that:

\[
\begin{align*}
\vec{r}^*_n &= \arg \max_{j} \{ F_j'(\vec{r}(n)) \}
\end{align*}
\]

since the summation term is common to all directions and so can be removed.

For example, if an operator charges on a per delivered bit basis then a natural utility function is given by \(U(r) = \alpha r\) for some constant \(\alpha\). The resulting scheduler will then pick the user for which:

\[
\begin{align*}
\vec{r}^*_n &= \arg \max_{j} \{ d_j(n) \}
\end{align*}
\]

(i.e., the one with the highest achievable bit rate). This scheduler is referred to as the Max C/I scheduler and provides the maximum sector throughput. However, it provides unfair allocation of resources (those in good radio conditions are preferred over those in bad radio conditions).

In order to address this unfairness, the PF utility function \(U(r) = \log(r)\) is used (see [3] for an explanation of why this function provides fairness) and hence the resulting scheduler picks the user such that:

\[
\begin{align*}
\vec{r}^*_n &= \arg \max_{j} \left\{ \frac{d_j(n)}{r_j(n)} \right\}
\end{align*}
\]

This provides some degree of fairness at the expense of reduced sector throughput (since users in bad conditions must also be served). In order to provide QoS guarantees (which neither Max C/I nor PF provides) while at the same time taking advantage of user diversity gains a Barrier Function scheduler may be used. These can also take into account different user classes and uses additional functions to provide “soft constraints” for each of the desired QoS guarantees.
III. EFFECTS OF SCHEDULING ON TCP PERFORMANCE

In this section we address the effects of various frame schedulers on TCP performance. In particular we focus on the performance when the system gets congested. This discussion provides the basis for our proposed congestion control method.

A. Proportional Fair Scheduling

Recall that for the PF scheduler, the next user to be served is the one with the highest supportable rate to throughput ratio. Secondly note that the average supportable rate of a user is dependent on the user’s average radio conditions. We assume a stationary system in which the average SINR (Signal to Interference and Noise Ratio) of user \( i \) is given by \( \Gamma_i \). The stationarity assumption is reasonable for a slowly moving propagation environments. For multi-path propagation environments, the received signal envelope has a Rayleigh distribution. Assuming additive Gaussian noise, the instantaneous SINR of AT \( i \), denoted by \( \gamma_i \), is distributed exponentially with probability density function

\[
p(\gamma) = \frac{1}{\Gamma_i} e^{-\gamma/\Gamma_i}, \quad \gamma \geq 0.
\]

We assume that the supportable rate of an AT, denoted by \( d_i \), is proportional to its instantaneous SINR so that \( d_i = \alpha \gamma_i \) where \( \alpha \) is the same for all ATs. Here we assume a continuous range of supportable rates as opposed to the finite number that are supported in practice. We also assume that each AT has reached its steady state throughput \( r_i \). The PF scheduler chooses the AT for which \( d_i/r_i \) is maximum. The chosen AT is then served at its supportable rate. We can compute the average throughput of user \( i \) by integrating over all rates, the probability that the AT is chosen to be served at that rate. The probability that \( i \) is served is the probability that \( d_i/r_i > d_j/r_j \) for all \( j \neq i \) which is the probability that \( \gamma_j < \gamma_i r_j/r_i \). In steady state the expected value of \( d_i/r_i \) is the same for all ATs hence \( \alpha \Gamma_i/r_i \) is the same for all ATs. Therefore \( r_j/r_i = \Gamma_j/\Gamma_i \) and hence the probability that \( i \) is served is the probability that \( \gamma_j < \gamma_i \Gamma_j/\Gamma_i \). Assuming \( N \) ATs, we can compute the average throughput for user \( i \) as follows,

\[
r_i = \int_0^\infty \alpha \frac{\gamma}{\Gamma_i} e^{-\gamma/\Gamma_i} \left( \prod_{j \neq i}^{N} \frac{\gamma \Gamma_j}{\Gamma_i} e^{-\gamma/\Gamma_j} \right) d\gamma
\]

\[
= \int_0^\infty \alpha \frac{\gamma}{\Gamma_i} e^{-\gamma/\Gamma_i} \left( \prod_{j \neq i}^{N} (1 - e^{-\gamma/\Gamma_j}) \right) d\gamma
\]

\[
= \int_0^\infty \alpha \frac{\gamma}{\Gamma_i} e^{-\gamma/\Gamma_i} (1 - e^{-\gamma/\Gamma_i})^{N-1} d\gamma
\]

\[
= \alpha \Gamma_i \int_0^\infty xe^{-x} (1 - e^{-x})^{N-1} dx
\]

\[
= D_i g(N)
\]

where \( D_i = \alpha \Gamma_i \) is the average supportable rate of AT \( i \) and \( g(N) \) is the attenuation factor due to the other ATs and is defined as

\[
g(N) = \int_0^\infty xe^{-x} (1 - e^{-x})^{N-1} dx.
\]

Note that the attenuation factor \( g(N) \) is independent of the radio conditions of the other ATs. Essentially the PF scheduler provides each user with the same number of service opportunities but these are provided to an AT when it is in good radio conditions. Unfortunately \( g(N) \) cannot be evaluated analytically and has to be evaluated numerically. Note that if a Round Robin scheduler is used then the average throughput of user \( i \) is given simply by \( D_i/N \). Therefore we can compare the attenuation factors for the PF scheduler, \( g(N) \) and the Round Robin Scheduler \( 1/N \) (see Fig. 1) to illustrate their throughput differences.

In 1XEV-DV, the power available to the F-PDCH depends on how many voice users are in the sector. Therefore, the power (and hence the sector throughput) available to data users will vary with time. Furthermore, as we see above, the throughput of a user varies with the number of active users in the sector which also varies over time. Consider the case of two users G and B with average throughputs of 100 and 50 kbps respectively. If the power available to data users drops by 20% then the average throughputs of these users will drop to 80 and 40 kbps respectively. The rate decrease for user G is 20 kbps while that for user B is 10 kbps. Hence user G’s buffer utilization increases at twice the rate at which user B’s utilization increases. The probability of buffer overflow and hence the probability that TCP times out for the corresponding connection is greater for user G than for user B. However since user G has a higher throughput then it is preferable to maintain user G at its high rate at the expense of an increased probability of buffer overflow for the lower throughput user B. This will result in better utilization of available resources. Our proposed congestion control mechanism attempts to achieve the objective of maintaining high throughputs for the high data rate users.
B. Max C/I Scheduler

We now consider the Max C/I scheduler. In this case the user to be served next is the one with the highest supportable rate. Using the same notation as in the previous section, the average rate of user $i$ can be computed in a similar manner as we did for the PF scheduler. In this case the probability that user $i$ is served is simply the probability that $\gamma_j < \gamma_i$ for all other users $j$. Hence we get

$$ r_i = \int_0^\infty \frac{\alpha}{\Gamma_i} \Gamma_i e^{-\gamma_i/a} \left( \prod_{j \neq i} \int_0^{\gamma_j} \frac{1}{\Gamma_j} e^{-x/\Gamma_j} dx \right) \gamma_j d\gamma_j $$

$$ = \int_0^\infty \frac{\alpha}{\Gamma_i} \Gamma_i e^{-\gamma_i/a} \left( \prod_{j \neq i} (1 - e^{-\gamma_j/\Gamma_j}) \right) d\gamma_j. $$

Given the average SINR for each user one can expand the above product and solve the associated integral. For brevity we only consider the case of two users for which we have

$$ r_1 = \int_0^\infty \frac{\alpha}{\Gamma_1} \Gamma_1 e^{-\gamma_1/a} \left(1 - e^{-\gamma_2/\Gamma_2}\right) d\gamma_1 $$

$$ = \alpha \Gamma_1 \left(1 - \frac{\Gamma_2}{(\Gamma_1 + \Gamma_2)^2}\right). $$

Similarly the throughput for the second user is given by,

$$ r_2 = \alpha \Gamma_2 \left(1 - \frac{\Gamma_1}{(\Gamma_1 + \Gamma_2)^2}\right). $$

Note that, unlike the PF and Round Robin cases, the throughput of a user is dependent on the radio conditions of the other users. If we solve the two user case for the PF scheduler we obtain $r_i = 0.75a \Gamma_i$. For Round Robin scheduling we get $r_i = a \Gamma_i / 2$ since the service opportunities for a user is random and equally divided between the two users. Suppose that user 2 is in worse radio conditions than user 1 then $0 \leq \Gamma_2 / \Gamma_1 \leq 1$. We fix the average SINR for user 1 and vary that of user 2 and plot the user throughputs normalized by the maximum achievable by user 1. This plot is presented in Fig. 2 for both PF and Max C/I schedulers.

Note that for a given ratio, the user in better radio conditions has a higher throughput using the Max C/I scheduler than with the PF scheduler. The opposite is true for the low throughput user. Consider again the above example of users G and B with average rates 100 and 50 kbps using a PF scheduler. In this case the average supportable rates are given by $D_1 = 100/0.75 = 133.3$ kbps and $D_2 = 66.67$ kbps. Hence if Max C/I was instead used then, using 2 and 3, their average throughputs would have instead been 118.5 kbps and 37.0 kbps. With a 20% drop in power the Max C/I rates would become 94.8 kbps for user G and 29.6 kbps for user B. Therefore, if immediately following the power reduction the scheduler switched from PF to Max C/I then the rate at which user G’s buffer increases would be 5.2 kbps while that for user B would be 20.4 kbps. Hence user G would maintain a high throughput while user B would quickly overflow its buffers causing TCP to either perform fast retransmit or time-out. However the net result is that the overall sector TCP throughput remains high.

C. Barrier Function Schedulers

The utility function of the Barrier Function scheduler consists of a base function (e.g., Max C/I or PF function) plus functions used to maintain the QoS guarantees for the connection. The parameters of both these functions may be dependent on the Class of the user. For example, assume that an operator provides three service levels (Gold, Silver and Bronze). If resources are available then all users are provided with a minimum average throughput of 64 kbps. However, the tightness of this constraint should be highest for the Gold users followed by the Silver users and then the Bronze users. As long as all users achieve their minimum rates then the operator would like to provide proportionally fair throughputs. In this case the base function is $\log(r)$ (i.e., the PF utility function) and we use an exponential barrier function for the minimum throughput constraint resulting in the following utility function,

$$ U(r) = \log(r) + (1 - e^{-\beta_j(r - 64)}). $$

The subscript $j$ is used to denote the User Class with $\beta_{\text{gold}} > \beta_{\text{silver}} > \beta_{\text{bronze}}$. The resulting scheduler chooses the user for which $d(1/r + \beta_j e^{-\beta_j(r - 64)})$ is maximum. As we did for the Max C/I and PF schedulers, one can compute the average throughput for each user. As long as resources are available then essentially the PF criterion is used but as soon as a user’s throughput drops below the minimum value then she is scheduled more often. When the radio capacity is reduced because of a drop in available power, the users in the lowest class (Bronze) are the first to be starved and hence the first to experience buffer overflow and TCP time outs. This further releases resources for higher class users and allows them to maintain their throughput levels. Therefore no additional congestion control procedures are needed for this scheduler.
IV. THE CONGESTION CONTROL ALGORITHM

Assuming that the radio link forms the throughput bottleneck then the total buffering (for IP packets and RLP frames) should ideally be set to the bandwidth-time product of the connection. If the buffer is made too large then large delays are incurred and this can result in poor performance of interactive applications if they share the same RLP connection as a bandwidth intensive application such as FTP. If the buffer is too small then the radio link is inefficiently utilized since the RLP queue may become zero for extended periods and, although the forward link is shared, the average number of users with data for transmission is reduced thus reducing the user diversity gains. The throughput of a user can vary between a few kbps to Mbps and depends on the user’s radio conditions and the number of active users in the sector. Also the round trip time may vary from a few milliseconds to several seconds depending on the location of the server. Hence it not possible to choose the optimal RLP buffer size. This means that in practice these buffers may overflow causing TCP to either time-out or invoke fast retransmits.

Our objective is to maintain high TCP throughput levels even after sudden reductions in capacity. If no control is put in place then all connections may suffer time-outs leading to TCP congestion collapse and inefficient resource utilization. As we saw in previous sections, for the same amount of transmission energy more bits can be transmitted to a user in good radio conditions than to one in bad radio conditions. Hence under overload (during which efficient utilization of resources is more important than fairness) one can maintain high TCP throughput by sacrificing those users in bad conditions for those in good conditions. This is the philosophy and approach we follow in the algorithm described below.

Recall that the Max C/I scheduler provides our desired objective under overload (high efficiency at the expense of fairness). The proposed algorithm simply shifts the scheduler utility function closer to that of the Max C/I scheduler function as the congestion increases. As we saw in earlier sections this maintains the throughput of high data rate users and accelerates the congestion of the low data rate users. Eventually the congestion subsides and the low data rate users will gradually increase their throughputs. Note that the congestion period will typically be short since admission control (which we do not discuss in this paper) will also detect the congestion condition and will block new users so that user departures will result in a reduction of the offered load. Next we outline possible methods for detecting the onset of congestion, controlling the congestion to achieve our stated objective and detecting the abatement of the congestion.

A. Congestion Detection

Arriving IP packets are split over multiple RLP frames for transmission over the air. This is performed at the Base Station Controller and is assumed not to be a bottleneck. We assume that almost all queuing is performed at the RLP frame buffers in the Base Station. Any sudden drop in capacity or sudden increase in the offered load is reflected in a sudden increase in the RLP buffer utilization. Hence we use the average (over all connections) RLP buffer utilization, \( \rho \), as the metric to be monitored for congestion detection. Note that if flow control is used between the Base Station Controller and the Base station then the buffer overflow will instead occur at the Base Station Controller. Therefore the detection will have to be performed at the Base Station Controller.

B. Congestion Control

Let \( U(r) \) denote the utility function used by the scheduler under normal conditions. Under congestion we propose switching to the following overload utility function,

\[
V(r) = \theta r + (1 - \theta) U(r),
\]

where \( \theta \) will be made congestion dependent (as explained below). As \( \theta \) is increased to unity and \( V(r) \) approaches the Max C/I function the following happens. The system capacity increases thus reducing the backlog of RLP frames and reducing the average buffer occupancy. Furthermore, users in poor radio conditions are served less often resulting in a rapid overflow of their buffers which in turn triggers TCP congestion controls. Hence the offered load for users in poor radio conditions decreases but it remains high for those in good radio conditions (hence avoiding the TCP congestion collapse cycling problem). As \( \theta \) is increased, \( \rho(n) \) will decrease. As \( \theta \) is moved back to zero then, if the system is still congested the average buffer occupancy once again increases. However, if congestion has subsided then the buffer occupancy will continue to remain small. One simple method for adjusting \( \theta \) is as follows:

At control interval \( nT \), do the following:

if \( \rho(n) > \rho_{up} \) set \( \theta = 1 \)
else if \( \rho(n) < \rho_{lo} \) set \( \theta = 0 \)
else set \( \theta = (\rho(n) - \rho_{lo})/(\rho_{up} - \rho_{lo}) \)

C. Congestion Abatement

Congestion can subside on its own (users run less demanding applications or more power becomes available to the F-PDCH) or due to the actions of the admission control algorithm (which reduces the connection load). In either case, the buffer utilization will eventually drop to its normal value even with \( \theta = 0 \). Hence, if \( \theta = 0 \) for some number of consecutive control intervals the algorithm is switched off.

REFERENCES