Network Working Group
Request for Comments: 4828
Category: Experimental

S. Floyd
ICIR
E. Kohler
UCLA

April 2007

TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document proposes a mechanism for further experimentation, but not for widespread deployment at this time in the global Internet.

TCP-Friendly Rate Control (TFRC) is a congestion control mechanism for unicast flows operating in a best-effort Internet environment (RFC 3448). TFRC was intended for applications that use a fixed packet size, and was designed to be reasonably fair when competing for bandwidth with TCP connections using the same packet size. This document proposes TFRC-SP, a Small-Packet (SP) variant of TFRC, that is designed for applications that send small packets. The design goal for TFRC-SP is to achieve the same bandwidth in bps (bits per second) as a TCP flow using packets of up to 1500 bytes. TFRC-SP enforces a minimum interval of 10 ms between data packets to prevent a single flow from sending small packets arbitrarily frequently.

Flows using TFRC-SP compete reasonably fairly with large-packet TCP and TFRC flows in environments where large-packet flows and small-packet flows experience similar packet drop rates. However, in environments where small-packet flows experience lower packet drop rates than large-packet flows (e.g., with Drop-Tail queues in units of bytes), TFRC-SP can receive considerably more than its share of the bandwidth.

Table of Contents

1. Introduction

This document specifies TFRC-SP, an experimental, Small-Packet variant of TCP-friendly Rate Control (TFRC) [RFC3448].

TFRC was designed to be reasonably fair when competing for bandwidth with TCP flows, but to avoid the abrupt changes in the sending rate characteristic of TCP's congestion control mechanisms. TFRC is intended for applications such as streaming media applications where a relatively smooth sending rate is of importance. Conventional TFRC measures loss rates by estimating the loss event ratio as described in [RFC3448], and uses this loss event rate to determine the sending rate in packets per round-trip time. This has consequences for the rate that a TFRC flow can achieve when sharing a bottleneck with large-packet TCP flows. In particular, a low-bandwidth, small-packet TFRC flow sharing a bottleneck with high-bandwidth, large-packet TCP flows may be forced to slow down, even though the TFRC application's nominal rate in bytes per second is less than the rate achieved by the TCP flows. Intuitively, this would be "fair" only if the network limitation was in packets per second (such as a routing lookup), rather than bytes per second (such as link bandwidth). Conventional wisdom is that many of the network limitations in today's Internet are in bytes per second, even though the network limitations of the future might move back towards limitations in packets per second.

TFRC-SP is intended for flows that need to send frequent small packets, with less than 1500 bytes per packet, limited by a minimum interval between packets of 10 ms. It will better support applications that do not want their sending rates in bytes per second to suffer from their use of small packets. This variant is restricted to applications that send packets no more than once every 10 ms (the Min Interval or minimum interval). Given this restriction, TFRC-SP effectively calculates the TFRC fair rate as if the bottleneck restriction was in bytes per second. Applications using TFRC-SP could have a fixed or naturally-varying packet size, or could vary their packet size in response to congestion. Applications that are not willing to be limited by a minimum interval of 10 ms between packets, or that want to send packets larger than 1500 bytes, should not use TFRC-SP. However, for applications with a minimum interval of at least 10 ms between packets and with data packets of at most 1500 bytes, the performance of TFRC-SP should be at least as good as that from TFRC.

RFC 3448, the protocol specification for TFRC, stated that TFRC-PS (for TFRC-PacketSize), a variant of TFRC for applications that have a fixed sending rate but vary their packet size in response to congestion, would be specified in a later document. This document instead specifies TFRC-SP, a variant of TFRC designed for applications that send small packets, where applications could either have a fixed or varying packet size or could adapt their packet size in response to congestion. However, as discussed in Section 6 of this document, there are many questions about how such an adaptive application would use TFRC-SP that are beyond the scope of this document, and that would need to be addressed in documents that are more application-specific.

TFRC-SP is motivated in part by the approach in RFC 3714, which argues that it is acceptable for VoIP flows to assume that the network limitation is in bytes per second (Bps) rather in packets per second (pps), and to have the same sending rate in bytes per second as a TCP flow with 1500-byte packets and the same packet drop rate. RFC 3714 states the following:

> "While the ideal would be to have a transport protocol that is able to detect whether the bottleneck links along the path are limited in Bps or in pps, and to respond appropriately when the limitation is in pps, such an ideal is hard to achieve. We would not want to delay the deployment of congestion control for telephony traffic until such an ideal could be accomplished. In addition, we note that the current TCP congestion control mechanisms are themselves not very effective in an environment where there is a limitation along the reverse path in pps. While the TCP mechanisms do provide an incentive to use large data packets, TCP does not include any effective congestion control mechanisms for the stream of small acknowledgement packets on the reverse path. Given the arguments above, it seems acceptable to us to assume a network limitation in Bps rather than in pps in considering the minimum sending rate of telephony traffic."

Translating the discussion in [RFC3714] to the congestion control mechanisms of TFRC, it seems acceptable to standardize a variant of TFRC that allows low-bandwidth flows sending small packets to achieve a rough fairness with TCP flows in terms of the sending rate in Bps, rather than in terms of the sending rate in pps. This is accomplished by TFRC-SP, a small modification to TFRC, as described below.

Maintaining incentives for large packets: Because the bottlenecks in the network in fact can include limitations in pps as well as in Bps, we pay special attention to the potential dangers of encouraging a large deployment of best-effort traffic in the Internet consisting entirely of small packets. This is discussed in more detail in Section 4.3. In addition, as again discussed in Section 4.3, TFRC-SP includes the limitation of the Min Interval between packets of 10 ms.

Packet drop rates as a function of packet size: TFRC-SP essentially assumes that the small-packet TFRC-SP flow receives roughly the same packet drop rate as a large-packet TFRC or TCP flow. As we show, this assumption is not necessarily correct for all environments in the Internet.

Initializing the Loss History after the First Loss Event: Section 6.3.1 of RFC 3448 specifies that the TFRC receiver initializes the loss history after the first loss event by calculating the loss interval that would be required to produce the receive rate measured over the most recent round-trip time. In calculating this loss interval, TFRC-SP uses the segment size of 1460 bytes, rather than the actual segment size used in the connection.

Calculating the loss event rate for TFRC-SP: TFRC-SP requires a modification in TFRC's calculation of the loss event rate, because a TFRC-SP connection can send many small packets when a standard TFRC or TCP connection would send a single large packet. It is not possible for a standard TFRC or TCP connection to repeatedly send multiple packets per round-trip time in the face of a high packet drop rate. As a result, TCP and standard TFRC only respond to a single loss event per round-trip time, and are still able to detect and respond to increasingly heavy packet loss rates. However, in a highly-congested environment, when a TCP connection might be sending, on average, one large packet per round-trip time, a corresponding TFRC-SP connection might be sending many small packets per round-trip time. As a result, in order to maintain fairness with TCP, and to be able to detect changes in the degree of congestion, TFRC-SP needs to be sensitive to the actual packet drop rate during periods of high congestion. This is discussed in more detail in the section below.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. TFRC-SP Congestion Control

TFRC uses the TCP throughput equation given in Section 3.1 of [RFC3448], which gives the allowed sending rate X in bytes per second as a function of the loss event rate, segment size, and round-trip time. [RFC3448] specifies that the segment size s used in the throughput equation should be the segment size used by the application, or the estimated mean segment size if there are variations in the segment size depending on the data. This gives rough fairness with TCP flows using the same segment size.

TFRC-SP changes this behavior in the following ways.

- The nominal segment size: The nominal segment size s defaults to 1460 bytes. Following [RFC3714], this provides a goal of fairness, in terms of the sending rate in bytes per second, with a TCP flow with 1460 bytes of application data per packet but with the same packet drop rate. If the endpoint knows the MTU (Maximum Transmission Unit) of the path and the derived MSS (Maximum Segment Size) is less than 1460 bytes, then the endpoint SHOULD set the nominal segment size s to MSS bytes. In addition, if the endpoint knows the MTU of the path and the resulting MSS is less than 536 bytes, then the endpoint MUST set the nominal segment size s to MSS bytes.

  However, this document does not require that TFRC-SP endpoints determine the path MTU. While most paths allow an MSS of 1460 bytes, some paths have a slightly smaller MSS due to tunnels (e.g., IPv6 over IPv4). In some specific cases, IPv4 paths may experience a much smaller path MTU. Due to the complications of estimating the path MTU, and to the fact that most paths support an MSS of at least 536 bytes, TFRC-SP as a default uses a nominal segment size of 1460 bytes. The nominal segment size is discussed in more detail in Section 4.5.3.

- Taking packet headers into account: The allowed transmit rate X in bytes per second is reduced by a factor that accounts for packet header size. This gives the application some incentive, beyond the Min Interval, not to use unnecessarily small packets. In particular, we introduce a new parameter H, which represents the expected size in bytes of network and transport headers to be used on the TFRC connection's packets. This is used to reduce the allowed transmit rate X as follows:

  X := X * s_true / (s_true + H),

  where s_true is the true average data segment size for the connection in bytes, excluding the transport and network headers. Section 4.1 of RFC 3448 states that where the packet size varies naturally with the data, an estimate of the mean segment size can be used for s_true. As suggested in Section 4.1 of [RFC3448bis], when an estimate of the mean segment size is used for s_true, the estimate SHOULD be calculated over at least the last four loss intervals. However, this document does not specify a specific algorithm for estimating the mean segment size.

  The H parameter is set to the constant 40 bytes. Thus, if the TFRC-SP application used 40-byte data segments, the allowed transmit rate X would be halved to account for the fact that half of the sending rate would be used by the headers. Section 4.2 justifies this definition. However, a connection using TFRC-SP MAY instead use a more precise estimate of H, based on the actual network and transport headers to be used on the connection's packets. For example, a Datagram Congestion Control Protocol (DCCP) connection [RFC4340] over IPv4, where data packets use the DCCP-Data packet type, and there are no IP or DCCP options, could set H to 20 + 12 = 32 bytes. However, if the TFRC implementation knows that the IP layer is using IPv6 instead of IPv4, then the connection using TFRC-SP MAY still use the default estimate of 40 bytes for H instead of the actual size of 60 bytes, for simplicity of implementation.

- Measuring the loss event rate in times of high loss: During short loss intervals (those at most two round-trip times), the loss rate is computed by counting the actual number of packets lost or marked, not by counting at most one loss event per loss interval. Without this change, TFRC-SP could send multiple packets per round-trip time even in the face of heavy congestion, for a steady-state behavior of multiple packets dropped each round-trip time.

  In standard TFRC, the TFRC receiver estimates the loss event rate by calculating the average loss interval in packets, and inverting to get the loss event rate. Thus, for a short loss interval with N packets and K losses, standard TFRC calculates the size of that loss interval as N packets, contributing to a loss event rate of 1/N. However, for TFRC-SP, for small loss intervals of at most two round-trip times, if the loss interval consists of N packets including K losses, the size of the loss interval is calculated as N/K, contributing to a loss event rate of K/N instead of 1/N.

  Section 5.4 of RFC 3448 specifies that the calculation of the average loss interval includes the most recent loss interval only if this increases the calculated average loss interval, as in the pseudocode below. However, in TFRC-SP the calculated loss interval size for a short loss interval varies as a function of the number of packet losses that have been detected, allowing either increases or decreases in the calculated loss interval size for the current short loss interval as new packets are received. Therefore, TFRC-SP adds the restriction that the calculation of the average loss interval can include the most recent loss interval only if more than two round-trip times have passed since the beginning of that loss interval.

Let the most recent loss intervals be I_0 to I_n, with I_0 being the interval including the most recent loss event, with the corresponding weights w_i as defined in RFC 3448. In RFC 3448 (Section 5.4), the average loss interval I_mean is calculated as follows:

```
I_tot0 = 0;
I_tot1 = 0;
W_tot = 0;
for (i = 0 to n-1) {
  I_tot0 = I_tot0 + (I_i * w_i);
  W_tot = W_tot + w_i;
}
for (i = 1 to n) {
  I_tot1 = I_tot1 + (I_i * w_(i-1));
}
I_tot = max(I_tot0, I_tot1);
I_mean = I_tot/W_tot;
```

In TFRC-SP, the average loss interval I_mean is instead calculated as follows:

```
I_tot0 = 0;
I_tot1 = 0;
W_tot = 0;
for (i = 0 to n-1) {
  I_tot0 = I_tot0 + (I_i * w_i);
  W_tot = W_tot + w_i;
}
for (i = 1 to n) {
  I_tot1 = I_tot1 + (I_i * w_(i-1));
}
If the current loss interval I_0 is "short"
  then I_tot = I_tot1;
  else I_tot = max(I_tot0, I_tot1);
I_mean = I_tot/W_tot;
```

o A minimum interval between packets: TFRC-SP enforces a Min Interval between packets of 10 ms. A flow that wishes its transport protocol to exceed this Min Interval MUST use the conventional TFRC equations, rather than TFRC-SP. The motivation for this is discussed below.

4. TFRC-SP Discussion

4.1. Response Functions and Throughput Equations

TFRC uses the TCP throughput equation given in [RFC3448], with the sending rate X in bytes per second as follows:

$$X = \frac{s}{R*sqrt(2*p/3) + (4*R* (3*sqrt(3*p/8) * p * (1+32*p^2)))} ,$$

where:

s is the packet size in bytes;

R is the round-trip time in seconds;

p is the loss event rate, between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.

This equation uses a retransmission timeout (RTO) of 4*R, and assumes that the TCP connection sends an acknowledgement for every data packet.

This equation essentially gives the response function for TCP as well as for standard TFRC (modulo TCP's range of sender algorithms for setting the RTO). As shown in Table 1 of [RFC3714], for high packet drop rates, this throughput equation gives rough fairness with the most aggressive possible current TCP: a SACK TCP flow using timestamps and Explicit Congestion Notification (ECN). Because it is not recommended for routers to use ECN-marking in highly-congested environments with high packet dropping/marking rates (Section 7 of [RFC3168]), we note that it would be useful to have a throughput equation with a somewhat more moderate sending rate for packet drop rates of 40% and above.

The effective response function of TFRC-SP can be derived from the TFRC response function by using a segment size s of 1460 bytes, and using the loss event rate actually experienced by the TFRC-SP flow. In addition, for loss intervals of at most two round-trip times, the loss event rate for TFRC-SP is estimated by counting the actual number of lost or marked packets, rather than by counting loss events. In addition, the sending rate for TFRC-SP is constrained to be at most 100 packets per second.

For an environment with a fixed packet drop rate p, regardless of packet size, the response functions of TCP, TFRC, and TFRC-SP are illustrated as follows, given in KBps (kilobytes per second), for a flow with a round-trip time of 100 ms:

	<-- TCP and Standard TFRC -->		
Packet DropRate	14-byte Segments	536-byte Segments	1460-byte Segments
0.00001	209.25	2232.00	5812.49
0.00003	120.79	1288.41	3355.24
0.00010	66.12	705.25	1836.58
0.00030	38.10	406.44	1058.45
0.00100	20.74	221.23	576.12
0.00300	11.76	125.49	326.79
0.01000	6.07	64.75	168.61
0.03000	2.99	31.90	83.07
0.10000	0.96	10.21	26.58
0.20000	0.29	3.09	8.06
0.30000	0.11	1.12	2.93
0.40000	0.05	0.48	1.26
0.50000	0.02	0.24	0.63

Table 1: Response Function for TCP and TFRC.
Sending Rate in KBps, as a Function of Packet Drop Rate.

	<---------- TFRC-SP -------->		
Packet DropRate	14-byte Segments	536-byte Segments	1460-byte Segments
0.00001	5.40	57.60	150.00
0.00003	5.40	57.60	150.00
0.00010	5.40	57.60	150.00
0.00030	5.40	57.60	150.00
0.00100	5.40	57.60	150.00
0.00300	5.40	57.60	150.00
0.01000	5.40	57.60	150.00
0.03000	5.40	57.60	83.07
0.10000	5.40	26.58	26.58
0.20000	5.40	8.06	8.06
0.30000	2.93	2.93	2.93
0.40000	1.26	1.26	1.26
0.50000	0.63	0.63	0.63

Table 2: Response Function for TFRC-SP.
Sending Rate in KBps, as a Function of Packet Drop Rate.
Maximum Sending Rate of 100 Packets per Second.

The calculations for Tables 1 and 2 use the packet loss rate for an approximation for the loss event rate p. Scripts and graphs for the tables are available from [VOIPSIMS]. As the well-known TCP response function in Table 1 shows, the sending rate for TCP and standard TFRC varies linearly with segment size. The TFRC-SP response function shown in Table 2 reflects the maximum sending rate of a hundred packets per second; when not limited by this maximum sending rate, the TFRC-SP flow receives the same sending rate in KBps as the TCP flow with 1460-byte segments, given the same packet drop rate. Simulations showing the TCP, standard TFRC, and TFRC-SP sending rates in response to a configured packet drop rate are given in Tables 7, 8, and 9, and are consistent with the response functions shown here.

	<-- TCP and Standard TFRC -->		
Byte DropRate	14-byte Segments	536-byte Segments	1460-byte Segments
0.0000001	284.76	929.61	1498.95
0.0000003	164.39	536.17	863.16
0.0000010	90.01	292.64	468.49
0.0000030	51.92	167.28	263.68
0.0000100	28.34	88.56	132.75
0.0000300	16.21	46.67	61.70
0.0001000	8.60	19.20	16.25
0.0003000	4.56	4.95	1.70
0.0010000	1.90	0.37	0.15
0.0030000	0.52	0.05	0.06
0.0100000	0.04	0.02	0.06
0.0300000	0.00	0.02	0.06

Table 3: Response Function for TCP and TFRC.
Sending Rate in KBps, as a Function of Byte Drop Rate.

	TFRC-SP		
Byte DropRate	14-byte Segments	536-byte Segments	1460-byte Segments
0.0000001	5.40	57.60	150.00
0.0000003	5.40	57.60	150.00
0.0000010	5.40	57.60	150.00
0.0000030	5.40	57.60	150.00
0.0000100	5.40	57.60	132.75
0.0000300	5.40	57.60	61.70
0.0001000	5.40	50.00	16.25
0.0003000	5.40	12.89	1.70
0.0010000	5.40	0.95	0.15
0.0030000	5.40	0.12	0.06
0.0100000	1.10	0.06	0.06
0.0300000	0.13	0.06	0.06

Table 4: Response Function for TFRC-SP.
Sending Rate in KBps, as a Function of Byte Drop Rate.
Maximum Sending Rate of 100 Packets per Second.

For Tables 3 and 4, the packet drop rate is calculated as 1-(1-b)^N, for a byte drop rate of b, and a packet size of N bytes. These tables use the packet loss rate as an approximation for the loss event rate p. The TCP response functions shown in Table 3 for fixed byte drop rates are rather different from the response functions shown in Table 1 for fixed packet drop rates; with higher byte drop rates, a TCP connection can have a higher sending rate using *smaller* packets. Table 4 also shows that with fixed byte drop rates, the sending rate for TFRC-SP can be significantly higher than that for TCP or standard TFRC, regardless of the TCP segment size. This is because in this environment, with small packets, TFRC-SP receives a small packet drop rate, but is allowed to send at the sending rate of a TCP or standard TFRC flow using larger packets but receiving the same packet drop rate.

Simulations showing TCP, standard TFRC, and TFRC-SP sending rates in response to a configured byte drop rate are given in Appendix B.2.

4.2. Accounting for Header Size

[RFC3714] makes the optimistic assumption that the limitation of the network is in bandwidth in bytes per second (Bps), and not in CPU cycles or in packets per second (pps). However, some attention must be paid to the load in pps as well as to the load in Bps. Even aside from the Min Interval, TFRC-SP gives the application some incentive to use fewer but larger packets, when larger packets would suffice, by including the bandwidth used by the packet header in the allowed sending rate.

As an example, a sender using 120-byte packets needs a TCP-friendly rate of 128 Kbps to send 96 Kbps of application data. This is because the TCP-friendly rate is reduced by a factor of s_true/(s_true + H) = 120/160, to account for the effect of packet headers. If the sender suddenly switched to 40-byte data segments, the allowed sending rate would reduce to 64 Kbps of application data; and the use of one-byte data segments would reduce the allowed sending rate to 3.12 Kbps of application data. (In fact, the Min Interval would prevent senders from achieving these rates, since applications using TFRC-SP cannot send more than 100 packets per second.)

Unless it has a more precise estimate of the header size, TFRC-SP assumes 40 bytes for the header size, although the header could be larger (due to IP options, IPv6, IP tunnels, and the like) or smaller (due to header compression) on the wire. Requiring the use of the exact header size in bytes would require significant additional complexity, and would have little additional benefit. TFRC-SP's default assumption of a 40-byte header is sufficient to get a rough estimate of the throughput, and to give the application some incentive not to use an excessive amount of small packets. Because we are only aiming at rough fairness, and at a rough incentive for applications, the default use of a 40-byte header in the calculations of the header bandwidth is sufficient for both IPv4 and IPv6.

4.3. The TFRC-SP Min Interval

The header size calculation provides an incentive for applications to use fewer, but larger, packets. Another incentive is that when the path limitation is in pps, the application using more small packets is likely to cause higher packet drop rates, and to have to reduce its sending rate accordingly. That is, if the congestion is in terms of pps, then the flow sending more pps will increase the packet drop rate, and have to adjust its sending rate accordingly. However, the increased congestion caused by the use of small packets in an environment limited by pps is experienced not only by the flow using the small packets, but by all of the competing traffic on that congested link. These incentives are therefore insufficient to provide sufficient protection for pps network limitations.

TFRC-SP, then, includes a Min Interval of 10 ms. This provides additional restrictions on the amount of small packets used.

One practical justification for the Min Interval is that the applications that currently want to send small packets are the VoIP applications that send at most one packet every 10 ms, so this restriction does not affect current traffic. A second justification is that there is no pressing need for best-effort traffic in the current Internet to send small packets more frequently than once every 10 ms (rather than taking the 10 ms delay at the sender, and merging the two small packets into one larger one). This 10 ms delay for merging small packets is likely to be dominated by the network propagation, transmission, and queueing delays of best-effort traffic in the current Internet. As a result, our judgment would be that the benefit to the user of having less than 10 ms between packets is outweighed by the benefit to the network of avoiding an excessive amount of small packets.

The Min Interval causes TFRC-SP not to support applications sending small packets very frequently. Consider a TFRC flow with a fixed packet size of 100 bytes, but with a variable sending rate and a fairly uncongested path. When this flow is sending at most 100 pps, it would be able to use TFRC-SP. If the flow wishes to increase its sending rate to more than 100 pps, but keep the same packet size, it would no longer be able to achieve this with TFRC-SP, and would have to switch to the default TFRC, receiving a dramatic, discontinuous decrease in its allowed sending rate. This seems not only acceptable, but desirable for the global Internet.

What is to prevent flows from opening multiple connections, each with a 10 ms Min Interval, thereby getting around the limitation of the Min Interval? Obviously, there is nothing to prevent flows from doing this, just as there is currently nothing to prevent flows from using UDP, or from opening multiple parallel TCP connections, or from using their own congestion control mechanism. Of course, implementations or middleboxes are also free to limit the number of parallel TFRC connections opened to the same destination in times of congestion, if that seems desirable. And flows that open multiple parallel connections are subject to the inconveniences of reordering and the like.

4.4. Counting Packet Losses

It is not possible for a TCP connection to persistently send multiple packets per round-trip time in the face of high congestion, with a steady-state with multiple packets dropped per round-trip time. For TCP, when one or more packets are dropped each round-trip time, the sending rate is quickly dropped to less than one packet per round-trip time. In addition, for TCP with Tahoe, NewReno, or SACK congestion control mechanisms, the response to congestion is largely independent of the number of packets dropped per round-trip time.

As a result, standard TFRC can best achieve fairness with TCP, even in highly congested environments, by calculating the loss event rate rather than the packet drop rate, where a loss event is one or more packets dropped or marked from a window of data.

However, with TFRC-SP, it is no longer possible to achieve fairness with TCP or with standard TFRC by counting only the loss event rate [WBL04]. Instead of sending one large packet per round-trip time, TFRC-SP could be sending N small packets (where N small packets equal one large 1500-byte packet). The loss measurement used with TFRC-SP has to be able to detect a connection that is consistently receiving multiple packet losses or marks per round-trip time, to allow TFRC-SP to respond appropriately.

In TFRC-SP, the loss event rate is calculated by counting at most one loss event in loss intervals longer than two round-trip times, and by counting each packet lost or marked in shorter loss intervals. In particular, for a short loss interval with N packets, including K lost or marked packets, the loss interval length is calculated as N/K, instead of as N. The average loss interval I_mean is still averaged over the eight most recent loss intervals, as specified in Section 5.4 of RFC 3448. Thus, if eight successive loss intervals are short loss intervals with N packets and K losses, the loss event rate is calculated as K/N, rather than as 1/N.

4.5. The Nominal Packet Size

4.5.1. Packet Size and Packet Drop Rates

The guidelines in Section 3 above say that the nominal segment size s is set to 1460 bytes, providing a goal of fairness, in terms of the sending rate in bytes per second, with a TCP flow with 1460 bytes of application data per packet but with the same packet drop rate. This follows the assumption that a TCP flow with 1460-byte segments will have a higher sending rate than a TCP flow with smaller segments. While this assumption holds in an environment where the packet drop rate is independent of packet size, this assumption does not necessarily hold in an environment where larger packets are more likely to be dropped than are small packets.

The table below shows the results of simulations with standard (SACK) TCP flows, where, for each *byte*, the packet containing that byte is dropped with probability p. Consider the approximation for the TCP response function for packet drop rates up to 10% or so; for these environments, the sending rate in bytes per RTT is roughly 1.2 s/sqrt(p), for a packet size of s bytes and packet drop rate p.

Given a fixed *byte* drop rate p1, and a TCP packet size of s bytes, the packet drop rate is roughly s*p1, producing a sending rate in bytes per RTT of roughly 1.2 sqrt(s)/sqrt(p1). Thus, for TCP in an environment with a fixed byte drop rate, the sending rate should grow roughly as sqrt(s), for packet drop rates up to 10% or so.

Each row of Table 5 below shows a separate simulation with ten TCP connections and a fixed byte drop rate of 0.0001, with each simulation using a different segment size. For each simulation, the TCP sending rate and goodput are averaged over the ten flows. As one would expect from the paragraph above, the TCP sending rate grows somewhat less than linearly with an increase in packet size, up to a packet size of 1460 bytes, corresponding to a packet drop rate of 13%. After that, further increases in the packet size result in a *decrease* in the TCP sending rate, as the TCP connection enters the regime of exponential backoff of the retransmit timer.

Segment Size (B)	Packet DropRate	TCP Rates (Kbps) SendRate	Goodput
14	0.005	6.37	6.34
128	0.016	30.78	30.30
256	0.028	46.54	44.96
512	0.053	62.43	58.37
1460	0.134	94.15	80.02
4000	0.324	35.20	21.44
8000	0.531	15.36	5.76

Table 5: TCP Median Send Rate vs. Packet Size I:
Byte Drop Rate 0.0001

Table 6 below shows similar results for a byte drop rate of 0.001. In this case, the TCP sending rate grows with increasing packet size up to a packet size of 128 bytes, corresponding to a packet drop rate of 16%. After that, the TCP sending rate decreases and then increases again, as the TCP connection enters the regime of exponential backoff of the retransmit timer. Note that with this byte drop rate, with packet sizes of 4000 and 8000 bytes, the TCP sending rate increases but the TCP goodput rate remains essentially zero. This makes sense, as almost all packets that are sent are dropped.

Segment Size (B)	Packet DropRate	TCP Rates (Kbps) SendRate	Goodput
14	0.053	1.68	1.56
128	0.159	7.66	6.13
256	0.248	6.21	4.32
512	0.402	1.84	1.11
1460	0.712	1.87	0.47
4000	0.870	3.20	0.00
8000	0.890	5.76	0.00

Table 6: TCP Median Send Rate vs. Packet Size II: Byte Drop Rate 0.001

The TCP behavior in the presence of a fixed byte drop rate suggests that instead of the goal of a TFRC-SP flow achieving the same sending rate in bytes per second as a TCP flow using 1500-byte packets, it makes more sense to consider an ideal goal of a TFRC-SP flow achieving the same sending rate as a TCP flow with the optimal packet size, given that the packet size is at most 1500 bytes. This does not mean that we need to change the TFRC-SP mechanisms for computing the allowed transmit rate; this means simply that in evaluating the fairness of TFRC-SP, we should consider fairness relative to the TCP flow using the optimal packet size (though still at most 1500 bytes) for that environment.

4.5.2. Fragmentation and the Path MTU

This document doesn't specify TFRC-SP behavior in terms of packet fragmentation and Path MTU Discovery (PMTUD). That is, should the transport protocol using TFRC-SP use PMTUD information to set an upper bound on the segment size? Should the transport protocol allow packets to be fragmented in the network? We leave these as questions for the transport protocol. As an example, we note that DCCP requires that endpoints keep track of the current PMTU, and says that fragmentation should not be the default (Section 14 of [RFC4340]).

4.5.3. The Nominal Segment Size and the Path MTU

When TFRC-SP is used with a nominal segment size s of 1460 bytes on a path where the TCP MSS is in fact only 536 bytes, the TFRC-SP flow could receive almost three times the bandwidth, in bytes per second, as that of a TCP flow using an MSS of 536 bytes. Similarly, in an environment with an MSS close to 4000 bytes, a TCP flow could receive almost three times the bandwidth of a TFRC-SP flow with its nominal segment size s of 1460 bytes. In both cases, we feel that these levels of "unfairness" with factors of two or three are acceptable; in particular, they won't result in one flow grabbing all of the available bandwidth, to the exclusion of the competing TCP or TFRC-SP flow.

All IPv4 *end hosts* are required to accept and reassemble IP packets of size 576 bytes [RFC791], but IPv4 *links* do not necessarily have to support this packet size. In slow networks, the largest possible packet may take a considerable amount of time to send [RFC3819], and a smaller MTU may be desirable, e.g., hundreds of bytes. If the first-hop link had a small MTU, then TCP would choose an appropriately small MSS [RFC879]. [RFC1144] quotes cases of very low link speeds where the MSS may be tens of bytes (and notes this is an extreme case). We note that if TFRC-SP is used over a path with an MTU considerably smaller than 576 bytes, and the TFRC-SP flow uses a nominal segment size s of 1460 bytes, then the TFRC-SP flow could receive considerably more than three times the bandwidth of competing TCP flows.

If TFRC-SP is used with a nominal segment size s of less than 536 bytes (because the path MTU is known to be less than 576 bytes), then TFRC-SP is likely to be of minimal benefit to applications. If TFRC-SP was to be used on paths that have a path MTU of considerably less than 576 bytes, and the transport protocol was not required to keep track of the path MTU, this could result in the TFRC-SP flow using the default nominal segment size of 1460 bytes, and as a result receiving considerably more bandwidth than competing TCP flows. As a result, TFRC-SP is not recommended for use with transport protocols that don't maintain some knowledge of the path MTU.

4.6. The Loss Interval Length for Short Loss Intervals

For a TFRC-SP receiver, the guidelines from Section 6 of RFC 3448 govern when the receiver should send feedback messages. In particular, from [RFC3448], "a feedback packet should ... be sent whenever a new loss event is detected without waiting for the end of an RTT". In addition, feedback packets are sent at least once per RTT.

For a TFRC-SP connection with a short current loss interval (less than two round-trip times), it is possible for the loss interval length calculated for that loss interval to change in odd ways as additional packet losses in that loss interval are detected. To prevent unnecessary oscillations in the average loss interval, Section 3 specifies that the current loss interval can be included in the calculation of the average loss interval only if the current loss interval is longer than two round-trip times.

5. A Comparison with RFC 3714

RFC 3714 considers the problems of fairness, potential congestion collapse, and poor user quality that could occur with the deployment of non-congestion-controlled IP telephony over congested best-effort networks. The March 2004 document cites ongoing efforts in the IETF, including work on TFRC and DCCP. RFC 3714 recommends that for best-effort traffic with applications that have a fixed or minimum sending rate, the application or transport protocol should monitor the packet drop rate, and discontinue sending for a period if the steady-state packet drop rate significantly exceeds the allowed threshold for that minimum sending rate.

In determining the allowed packet drop rate for a fixed sending rate, RFC 3714 assumes that an IP telephony flow should be allowed to use the same sending rate in bytes per second as a 1500-byte packet TCP flow experiencing the same packet drop rate as that of the IP telephony flow. As an example, following this guideline, a VoIP connection with a round-trip time of 0.1 sec and a minimum sending rate of 64 Kbps would be required to terminate or suspend when the persistent packet drop rate significantly exceeded 25%.

One limitation of the lack of fine-grained control in the minimal mechanism described in RFC 3714 is that an IP telephony flow would not adapt its sending rate in response to congestion. In contrast, with TFRC-SP, a small-packet flow would reduce its sending rate somewhat in response to moderate packet drop rates, possibly avoiding a period with unnecessarily-heavy packet drop rates.

Because RFC 3714 assumes that the allowed packet drop rate for an IP telephony flow is determined by the sending rate that a TCP flow would use *with the same packet drop rate*, the minimal mechanism in RFC 3714 would not provide fairness between TCP and IP telephony traffic in an environment where small packets are less likely to be dropped than large packets. In such an environment, the small-packet IP telephony flow would make the optimistic assumption that a large-packet TCP flow would receive the same packet drop rate as the IP telephony flow, and as a result the small-packet IP telephony flow would receive a larger fraction of the link bandwidth than a competing large-packet TCP flow.

6. TFRC-SP with Applications that Modify the Packet Size

One possible use for TFRC-SP would be with applications that maintain a fixed sending rate in packets per second, but modify their packet size in response to congestion. TFRC-SP monitors the connection's packet drop rate, and determines the allowed sending rate in bytes per second. Given an application with a fixed sending rate in packets per second, the TFRC-SP sender could determine the data packet size that would be needed for the sending rate in bytes per second not to exceed the allowed sending rate. In environments where the packet drop rate is affected by the packet size, decreasing the packet size could also result in a decrease in the packet drop rate experienced by the flow.

There are many questions about how an adaptive application would use TFRC-SP that are beyond the scope of this document. In particular, an application might wish to avoid unnecessary reductions in the packet size. In this case, an application might wait for some period of time before reducing the packet size, to avoid an unnecessary reduction in the packet size. The details of how long an application might wait before reducing the packet size can be addressed in documents that are more application-specific.

Similarly, an application using TFRC-SP might only have a few packet sizes that it is able to use. In this case, the application might not reduce the packet size until the current packet drop rate has significantly exceeded the packet drop rate threshold for the current sending rate, to avoid unnecessary oscillations between two packet sizes and two sending rates. Again, the details will have to be addressed in documents that are more application-specific.

Because the allowed sending rate in TFRC-SP is in bytes per second rather than in packets per second, there is little opportunity for applications to manipulate the packet size in order to "game" the system. This differs from TFRC in CCID 3 (Section 5.3 of [RFC4342]), where the allowed sending rate is in packets per second. In particular, a TFRC-SP application that sends small packets for a while, building up a fast sending rate, and then switches to large packets, would not increase its overall sending rate by the change of packet size.

7. Simulations

This section describes the performance of TFRC-SP in simulation scenarios with configured packet or byte drop rates, and in scenarios with a range of queue management mechanisms at the congested link. The simulations, described in detail in Appendix B, explore environments where standard TFRC significantly limits the throughput of small-packet flows, and TFRC-SP gives the desired throughput. The simulations also explore environments where standard TFRC allows small-packet flows to receive good performance, while TFRC-SP is overly aggressive.

The general lessons from the simulations are as follows.

- o In scenarios where large and small packets receive similar packet drop rates, TFRC-SP gives roughly the desired sending rate (Appendix B.1, B.2).

- o In scenarios where each *byte* is equally likely to be dropped, standard TFRC gives reasonable fairness between small-packet TFRC flows and large-packet TCP flows (Appendix B.2).

- o In scenarios where small packets are less likely to be dropped than large packets, TFRC-SP does not give reasonable fairness between small-packet TFRC-SP flows and large-packet TCP flows; small-packet TFRC-SP flows can receive considerably more bandwidth than competing large-packet TCP flows, and in some cases large-packet TCP flows can be essentially starved by competing small-packet TFRC-SP flows (Appendix B.2, B.3, B.4).

- o Scenarios where small packets are less likely to be dropped than large packets include those with Drop-Tail queues in bytes, and with AQM mechanisms in byte mode (Appendix B.3, B.4). It has also been reported that wireless links are sometimes good enough to let small packets through, while larger packets have a much higher error rate, and hence a higher drop probability [S05].

8. General Discussion

Dropping rates for small packets: The goal of TFRC-SP is for small-packet TFRC-SP flows to have rough fairness with large-packet TCP flows in the sending rate in bps, in a scenario where each packet receives roughly the same probability of being dropped. In a scenario where large packets are more likely to be dropped than small packets, or where flows with a bursty sending rate are more likely to have packets dropped than are flows with a smooth sending rate, small-packet TFRC-SP flows can receive significantly more bandwidth than competing large-packet TCP flows.

The accuracy of the TCP response function used in TFRC: For applications with a maximum sending rate of 96 Kbps or less (i.e., packets of at most 120 bytes), TFRC-SP only restricts the sending rate when the packet drop rate is fairly high, e.g., greater than 10%. [Derivation: A TFRC-SP flow with a 200 ms round-trip time and a maximum sending rate with packet headers of 128 Kbps would have a sending rate in bytes per second equivalent to a TCP flow with 1460-byte segments sending 2.2 packets per round-trip time. From Table 1 of RFC 3714, this sending rate can be sustained with a packet drop rate slightly greater than 10%.] In this high-packet-drop regime, the performance of TFRC-SP is determined in part by the accuracy of the TCP response function in representing the actual sending rate of a TCP connection.

In the regime of high packet drop rates, TCP performance is also affected by the TCP algorithm (e.g., SACK or not), the minimum RTO, the use of Limited Transmit (or lack thereof), the use of ECN, and the like. It is good to ensure that simulations or experiments exploring fairness include the exploration of fairness with the most aggressive TCP mechanisms conformant with the current standards. Our simulations use SACK TCP with Limited Transmit and with a minimum RTO of 200 ms. The simulation results are largely the same with or without timestamps; timestamps were not used for simulations reported in this paper. We didn't use TCP with ECN in setting the target sending rate for TFRC, because, as explained in Appendix B.1, our expectation is that in high packet drop regimes, routers will drop rather than mark packets, either from policy or from buffer overflow.

Fairness with different packet header sizes: In an environment with IPv6 and/or several layers of network-layer tunnels (e.g., IPsec, Generic Routing Encapsulation (GRE)), the packet header could be 60, 80, or 100 bytes instead of the default 40 bytes assumed in Section 3. For an application with small ten-byte data segments, this means that the actual packet size could be 70, 90, or 110 bytes, instead of the 50 bytes assumed by TFRC-SP in calculating the allowed sending rate. Thus, a TFRC-SP application with large headers could receive more than twice the bandwidth (including the bandwidth used by headers) than a TFRC-SP application with small headers. We do not expect this to be a problem; in particular, TFRC-SP applications with large headers will not significantly degrade the performance of competing TCP applications, or of competing TFRC-SP applications with small headers.

General issues for TFRC: The congestion control mechanisms in TFRC and TFRC-SP limit a flow's sending rate in packets per second. Simulations by Tom Phelan [P04] explore how such a limitation in sending rate can lead to unfairness for the TFRC flow in some scenarios, e.g., when competing with bursty TCP web traffic, in scenarios with low levels of statistical multiplexing at the congested link.

9. Security Considerations

There are no new security considerations introduced in this document.

The issues of the fairness of TFRC-SP with standard TFRC and TCP in a range of environments, including those with byte-based queue management at the congested routers, are discussed extensively in the main body of this document.

General security considerations for TFRC are discussed in RFC 3448. As with TFRC in RFC 3448, TFRC-SP is not a transport protocol in its own right, but a congestion control mechanism that is intended to be used in conjunction with a transport protocol. Therefore, security primarily needs to be considered in the context of a specific transport protocol and its authentication mechanisms. As discussed for TFRC in RFC 3448, any transport protocol that uses TFRC-SP needs to protect against spoofed feedback, and to protect the congestion control mechanisms against incorrect information from the receiver. Again as discussed for TFRC in RFC 3448, we expect that protocols incorporating ECN with TFRC-SP will want to use the ECN nonce [RFC3540] to protect the sender from the accidental or malicious concealment of marked packets

Security considerations for DCCP's Congestion Control ID 3, TFRC Congestion Control, the transport protocol that uses TFRC, are discussed in [RFC4342]. That document extensively discussed the mechanisms the sender can use to verify the information sent by the receiver, including the use of the ECN nonce.

10. Conclusions

This document has specified TFRC-SP, a Small-Packet (SP) variant of TFRC, designed for applications that send small packets, with at most a hundred packets per second, but that desire the same sending rate in bps as a TCP connection experiencing the same packet drop rate but sending packets of 1500 bytes. TFRC-SP competes reasonably well with large-packet TCP and TFRC flows in environments where large-packet flows and small-packet flows experience similar packet drop rates, but receives more than its share of the bandwidth in bps in environments where small packets are less likely to be dropped or marked than are large packets. As a result, TFRC-SP is experimental, and is not intended for widespread deployment at this time in the global Internet.

In order to allow experimentation with TFRC-SP in the Datagram Congestion Control Protocol (DCCP), an experimental Congestion Control IDentifier (CCID) will be used, based on TFRC-SP but specified in a separate document.

11. Acknowledgements

We thank Tom Phelan for discussions of TFRC-SP and for his paper exploring the fairness between TCP and TFRC-SP flows. We thank Lars Eggert, Gorry Fairhurst, Mark Handley, Miguel Garcia, Ladan Gharai, Richard Nelson, Colin Perkins, Juergen Quittek, Pete Sholander, Magnus Westerlund, and Joerg Widmer for feedback on earlier versions of this document. We also thank the DCCP Working Group for feedback and discussions.

Appendix A. Related Work on Small-Packet Variants of TFRC

Other proposals for variants of TFRC for applications with variable packet sizes include [WBL04] and [V00]. [V00] proposed that TFRC should be modified so that flows are not penalized by sending smaller packets. In particular, [V00] proposes using the path MTU in the TCP-friendly equation, instead of the actual packet size used by TFRC, and counting the packet drop rate by using an estimation algorithm that aggregates both packet drops and received packets into MTU-sized units.

[WBL04] also argues that adapting TFRC for variable packet sizes by just using the packet size of a reference TCP flow in the TFRC equation would not suffice in the high-packet-loss regime; such a modified TFRC would have a strong bias in favor of smaller packets, because multiple lost packets in a single round-trip time would be aggregated into a single packet loss. [WBL04] proposes modifying the loss measurement process to account for the bias in favor of smaller packets.

The TFRC-SP variant of TFRC proposed in our document differs from [WBL04] in restricting its attention to flows that send at most 100 packets per second. The TFRC-SP variant proposed in our document also differs from the straw proposal discussed in [WBL04] in that the allowed bandwidth includes the bandwidth used by packet headers.

[WBL04] discusses four methods for modifying the loss measurement process, "unbiasing", "virtual packets", "random sampling", and "Loss Insensitive Period (LIP) scaling". [WBL04] finds only the second and third methods sufficiently robust when the network drops packets independently of packet size. They find only the second method sufficiently robust when the network is more likely to drop large packets than small packets. Our method for calculating the loss event rate is somewhat similar to the random sampling method proposed in [WBL04], except that randomization is not used; instead of randomization, the exact packet loss rate is computed for short loss intervals, and the standard loss event rate calculation is used for longer loss intervals. [WBL04] includes simulations with a Bernoulli loss model, a Bernoulli loss model with a drop rate varying over time, and a Gilbert loss model, as well as more realistic simulations with a range of TCP and TFRC flows.

[WBL04] produces both a byte-mode and a packet-mode variant of the TFRC transport protocol, for connections over paths with per-byte and per-packet dropping respectively. We would argue that in the absence of transport-level mechanisms for determining whether the packet dropping in the network is per-packet, per-byte, or somewhere in between, a single TFRC implementation is needed, independently of the packet-dropping behaviors of the routers along the path. It would of course be preferable to have a mechanism that gives roughly acceptable behavior, to the connection and to the network as a whole, on paths with both per-byte and per-packet dropping (and on paths with multiple congested routers, some with per-byte dropping mechanisms, some with per-packet dropping mechanisms, and some with dropping mechanisms that lie somewhere between per-byte and per-packet).

An important contribution would be to investigate the range of behaviors actually present in today's networks, in terms of packet dropping as a function of packet size.

Appendix B. Simulation Results

This appendix reports on the simulation results outlined in Section 7. TFRC-SP has been added to the NS simulator, and is illustrated in the validation test "./test-all-friendly" in the directory tcl/tests. The simulation scripts and graphs for the simulations in this document are available at [VOIPSIMS].

B.1. Simulations with Configured Packet Drop Rates

In this section we describe simulation results from simulations comparing the throughput of standard (SACK) TCP flows, TCP flows with timestamps and ECN, TFRC-SP flows, and standard TFRC (Stnd TFRC) flows. In these simulations we configure the router to randomly drop or mark packets at a specified rate, independently of the packet size. For each specified packet drop rate, we give a flow's average sending rate in Kbps over the second half of a 100-second simulation, averaged over ten flows.

Packet DropRate	Send Rates (Kbps, 1460B seg) TCP	ECN TCP	TFRC
0.001	2020.85	1904.61	982.09
0.005	811.10	792.11	878.08
0.01	515.45	533.19	598.90
0.02	362.93	382.67	431.41
0.04	250.06	252.64	284.82
0.05	204.48	218.16	268.51
0.1	143.30	148.41	146.03
0.2	78.65	93.23*	55.14
0.3	26.26	59.65*	32.87
0.4	9.87	47.79*	25.45
0.5	3.53	32.01*	18.52

* ECN scenarios marked with an asterisk are not realistic, as routers are not recommended to mark packets when packet drop/mark rates are 20% or higher.

Table 7: Send Rate vs. Packet Drop Rate I:
1460B TFRC Segments
(1.184 Kbps Maximum TFRC Data Sending Rate)

Table 7 shows the sending rate for a TCP and a standard TFRC flow for a range of configured packet drop rates, when both flows have 1460-byte data segments, in order to illustrate the relative fairness of TCP and TFRC when both flows use the same packet size. For example, a packet drop rate of 0.1 means that 10% of the TCP and TFRC packets are dropped. The TFRC flow is configured to send at most 100 packets per second. There is good relative fairness until the packet drop percentages reach 40 and 50%, when the TFRC flow receives three to five times more bandwidth than the standard TCP flow. We note that an ECN TCP flow would receive a higher throughput than the TFRC flow at these high packet drop rates, if ECN-marking was still being used instead of packet dropping. However, we don't use the ECN TCP sending rate in these high-packet-drop scenarios as the target sending rate for TFRC, as routers are advised to drop rather than mark packets during high levels of congestion (Section 7 of [RFC3168]). In addition, there is likely to be buffer overflow in scenarios with such high packet dropping/marking rates, forcing routers to drop packets instead of ECN-marking them.

	< - - - - - - - Send Rates (Kbps) - - - - - - >			
Packet DropRate	TCP (1460B seg)	ECN TCP (1460B seg)	TFRC-SP (14B seg)	Stnd TFRC (14B seg)
0.001	1787.54	1993.03	17.71	17.69
0.005	785.11	823.75	18.11	17.69
0.01	533.38	529.01	17.69	17.80
0.02	317.16	399.62	17.69	13.41
0.04	245.42	260.57	17.69	8.84
0.05	216.38	223.75	17.69	7.63
0.1	142.75	138.36	17.69	4.29
0.2	58.61	91.54*	17.80	1.94
0.3	21.62	63.96*	10.26	1.00
0.4	10.51	41.74*	4.78	0.77
0.5	1.92	19.03*	2.41	0.56

* ECN scenarios marked with an asterisk are not realistic, as routers are not recommended to mark packets when packet drop/mark rates are 20% or higher.

Table 8: Send Rate vs. Packet Drop Rate II:
14B TFRC Segments
(5.6 Kbps Maximum TFRC Data Sending Rate)

Table 8 shows the results of simulations where each TFRC-SP flow has a maximum data sending rate of 5.6 Kbps, with 14-byte data packets and a 32-byte packet header for DCCP and IP. Each TCP flow has a receive window of 100 packets and a data packet size of 1460 bytes, with a 40-byte packet header for TCP and IP. The TCP flow uses SACK TCP with Limited Transmit, but without timestamps or ECN. Each flow has a round-trip time of 240 ms in the absence of queueing delay.

The TFRC sending rate in Table 8 is the sending rate for the 14-byte data packet with the 32-byte packet header. Thus, only 30% of the TFRC sending rate is for data, and with a packet drop rate of p, only a fraction 1-p of that data makes it to the receiver. Thus, the TFRC data receive rate can be considerably less than the TFRC sending rate in the table. Because TCP uses large packets, 97% of the TCP sending rate is for data, and the same fraction 1-p of that data makes it to the receiver.

Table 8 shows that for the 5.6 Kbps data stream with TFRC, Standard TFRC (Stnd TFRC) gives a very poor sending rate in bps, relative to the sending rate for the large-packet TCP connection. In contrast, the sending rate for the TFRC-SP flow is relatively close to the desired goal of fairness in bps with TCP.

Table 8 shows that with TFRC-SP, the 5.6 Kbps data stream doesn't reduce its sending rate until packet drop rates are greater than 20%, as desired. With packet drop rates of 30-40%, the sending rate for the TFRC-SP flow is somewhat less than that of the average large-packet TCP flow, while for packet drop rates of 50% the sending rate for the TFRC-SP flow is somewhat greater than that of the average large- packet TCP flow.

	< - - - - - - - Send Rates (Kbps) - - - - - - >			
Packet DropRate	TCP (1460B seg)	ECN TCP (1460B seg)	TFRC-SP (200B seg)	Stnd TFRC (200B seg)
0.001	1908.98	1869.24	183.45	178.35
0.005	854.69	835.10	185.06	138.06
0.01	564.10	531.10	185.33	92.43
0.02	365.38	369.10	185.57	62.18
0.04	220.80	257.81	185.14	45.43
0.05	208.97	219.41	180.08	39.44
0.1	141.67	143.88	127.33	21.96
0.2	62.66	91.87*	54.66	9.40
0.3	16.63	65.52*	24.50	4.73
0.4	6.62	42.00*	13.47	3.35
0.5	1.01	21.34*	10.51	2.92

* ECN scenarios marked with an asterisk are not realistic, as routers are not recommended to mark packets when packet drop/mark rates are 20% or higher.

Table 9: Sending Rate vs. Packet Drop Rate III:
200B TFRC Segments
(160 Kbps Maximum TFRC Data Sending Rate)

Table 9 shows results with configured packet drop rates when the TFRC flow uses 200-byte data packets, with a maximum data sending rate of 160 Kbps. As in Table 8, the performance of Standard TFRC is quite poor, while the performance of TFRC-SP is essentially as desired for packet drop rates up to 30%. Again as expected, with packet drop rates of 40-50% the TFRC-SP sending rate is somewhat higher than desired.

For these simulations, the sending rate of a TCP connection using timestamps is similar to the sending rate shown for a standard TCP connection without timestamps.

B.2. Simulations with Configured Byte Drop Rates

In this section we explore simulations where the router is configured to drop or mark each *byte* at a specified rate. When a byte is chosen to be dropped (or marked), the entire packet containing that byte is dropped (or marked).

		< - - - - - - Send Rates (Kbps) - - - - - - >			
Byte DropRate	TCP SegSize	TCP	ECN TCP	TFRC-SP (14B seg)	Stnd TFRC (14B seg)
0.00001	1460	423.02	431.26	17.69	17.69
0.0001	1460	117.41	114.34	17.69	17.69
0.001	128	10.78	11.50	17.69	8.37
0.005	14	1.75	2.89	18.39	1.91
0.010	1460	0.31	0.26	7.07	0.84
0.020	1460	0.29	0.26	1.61	0.43
0.040	1460	0.12	0.26*	0.17	0.12
0.050	1460	0.15	0.26*	0.08	0.06

* ECN scenarios marked with an asterisk are not realistic, as routers are not recommended to mark packets when packet drop/mark rates are 20% or higher.

TFRC's maximum data sending rate is 5.6 Kbps.

Table 10: Sending Rate vs. Byte Drop Rate

Table 10 shows the TCP and TFRC send rates for various byte drop rates. For each byte drop rate, Table 10 shows the TCP sending rate, with and without ECN, for the TCP segment size that gives the highest TCP sending rate. Simulations were run with TCP segments of 14, 128, 256, 512, and 1460 bytes. Thus, for a byte drop rate of 0.00001, the table shows the TCP sending rate with 1460-byte data segments, but with a byte drop rate of 0.001, the table shows the TCP sending rate with 128-byte data segments. For each byte drop rate, Table 10 also shows the TFRC-SP and Standard TFRC sending rates. With configured byte drop rates, TFRC-SP gives an unfair advantage to the TFRC-SP flow, while Standard TFRC gives essentially the desired performance.

Byte DropRate	TCP Pkt DropRate (1460B seg)	TFRC Pkt DropRate (14B seg)	TCP/TFRC Pkt Drop Ratio
0.00001	0.015	0.0006	26.59
0.0001	0.13	0.0056	24.94
0.001	0.77	0.054	14.26
0.005	0.99	0.24	4.08
0.01	1.00	0.43	2.32
0.05	1.00	0.94	1.05

Table 11: Packet Drop Rate Ratio vs. Byte Drop Rate

Table 11 converts the byte drop rate p to packet drop rates for the TCP and TFRC packets, where the packet drop rate for an N-byte packet is 1-(1-p)^N. Thus, a byte drop rate of 0.001, with each byte being dropped with probability 0.001, converts to a packet drop rate of 0.77, or 77%, for the 1500-byte TCP packets, and a packet drop rate of 0.054, or 5.4%, for the 56-byte TFRC packets.

The right column of Table 11 shows the ratio between the TCP packet drop rate and the TFRC packet drop rate. For low byte drop rates, this ratio is close to 26.8, the ratio between the TCP and TFRC packet sizes. For high byte drop rates, where even most small TFRC packets are likely to be dropped, this drop ratio approaches 1. As Table 10 shows, with byte drop rates, the Standard TFRC sending rate is close to optimal, competing fairly with a TCP connection using the optimal packet size within the allowed range. In contrast, the TFRC-SP connection gets more than its share of the bandwidth, though it does reduce its sending rate for a byte drop rate of 0.01 or more (corresponding to a TFRC-SP *packet* drop rate of 0.43.

Table 10 essentially shows three separate regions. In the low-congestion region, with byte drop rates less than 0.0001, the TFRC-SP connection is sending at its maximum sending rate. In this region the optimal TCP connection is the one with 1460-byte segments, and the TCP sending rate goes as 1/sqrt(p), for packet drop rate p. This low-congestion region holds for packet drop rates up to 10-15%.

In the middle region of Table 10, with byte drop rates from 0.0001 to 0.005, the optimal TCP segment size is a function of the byte drop rate. In particular, the optimal TCP segment size seems to be the one that keeps the packet drop rate at most 15%, keeping the TCP connection in the regime controlled by a 1/sqrt(p) sending rate, for packet drop rate p. For a TCP packet size of S bytes (including headers), and a *byte* drop rate of B, the packet drop rate is roughly B*S. For a given byte drop rate in this regime, if the optimal TCP performance occurs with a packet size chosen to give a packet drop rate of at most 15%, keeping the TCP connection out of the regime of exponential backoffs of the retransmit timer, then this requires B*S = 0.15, or S = 0.15/B.

In the high-congestion regime of Table 10, with high congestion and with byte drop rates of 0.01 and higher, the TCP performance is dominated by the exponential backoff of the retransmit timer regardless of the segment size. Even a 40-byte packet with a zero-byte data segment would have a packet drop rate of at least 33%. In this regime, the optimal TCP *sending* rate comes with a large, 1460-byte data segment, but the TCP sending rate is low with any segment size, considerably less than one packet per round-trip time.

We note that in this regime, while a larger packet gives a higher TCP *sending* rate, a smaller packet gives a better *goodput* rate.

In general, Tables 8 and 9 show good performance for TFRC-SP in environments with stable packet drop rates, where the decision to drop a packet is independent of the packet size. However, in some environments the packet size might affect the likelihood that a packet is dropped. For example, with heavy congestion and a Drop Tail queue with a fixed number of bytes rather than a fixed number of packet-sized buffers, small packets might be more likely than large packets to find room at the end of an almost-full queue. As a further complication, in a scenario with Active Queue Management, the AQM mechanism could either be in packet mode, dropping each packet with equal probability, or in byte mode, dropping each byte with equal probability. Sections B.3 and B.4 show simulations with packets dropped at Drop-Tail or AQM queues, rather that from a probabilistic process.

B.3. Packet Dropping Behavior at Routers with Drop-Tail Queues

One of the problems with comparing the throughput of two flows using different packet sizes is that the packet size itself can influence the packet drop rate [V00, WBL04].

The default TFRC was designed for rough fairness with TCP, for TFRC and TCP flows with the same packet size and experiencing the same packet drop rate. When the issue of fairness between flows with different packets sizes is addressed, it matters whether the packet drop rates experienced by the flows is related to the packet size. That is, are small packets just as likely to be dropped as large TCP packets, or are the smaller packets less likely to be dropped [WBL04]? And what is the relationship between the packet-dropping behavior of the path, and the loss event measurements of TFRC?

	< - - - - - - Send Rates in Kbps	- - - - >		
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.04	316.18	0.05	183.05
25	0.07	227.47	0.07	181.23
50	0.08	181.10	0.08	178.32
100	0.14	85.97	0.12	151.42
200	0.17	61.20	0.14	73.88
400	0.20	27.79	0.18	36.81
800	0.29	3.50	0.27	16.33
1600	0.37	0.63	0.33	6.29

Table 12: Drop and Send Rates for Drop-Tail Queues in Packets

Table 12 shows the results of the second half of 100-second simulations, with five TCP connections and five TFRC-SP connections competing with web traffic in a topology with a 3 Mbps shared link. The TFRC-SP application generates 200-byte data packets every 10 ms, for a maximum data rate of 160 Kbps. The five long-lived TCP connections use a data packet size of 1460 bytes, and the web traffic uses a data packet size of 512 bytes. The five TCP connections have round-trip times from 40 to 240 ms, and the five TFRC connections have the same set of round-trip times. The SACK TCP connections in these simulations use the default parameters in the NS simulator, with Limited Transmit, and a minimum RTO of 200 ms. Adding timestamps to the TCP connection didn't change the results appreciably. The simulations include reverse-path traffic, to add some small control packets to the forward path, and some queueing delay to the reverse path. The number of web sessions is varied to create different levels of congestion. The Drop-Tail queue is in units of packets, which each packet arriving to the queue requires a single buffer, regardless of the packet size.

Table 12 shows the average TCP and TFRC sending rates, each averaged over the five flows. As expected, the TFRC-SP flows see similar packet drop rates as the TCP flows, though the TFRC-SP flows receives higher throughput than the TCP flows with packet drop rates of 25% or higher.

	< - - - - - - Send Rates in Kbps - - - - >			
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.061	239.81	0.004	185.19
25	0.089	189.02	0.006	184.95
50	0.141	99.46	0.013	185.07
100	0.196	16.42	0.022	183.77
200	0.256	4.46	0.032	181.98
400	0.291	4.61	0.051	151.88
800	0.487	1.01	0.078	113.10
1600	0.648	0.67	0.121	65.17

Table 13: Drop and Send Rates for Drop-Tail Queues in Bytes I:
1460B TCP Segments

However, the fairness results can change significantly if the Drop-Tail queue at the congested output link is in units of bytes rather than packets. For a queue in packets, the queue has a fixed number of buffers, and each buffer can hold exactly one packet, regardless of its size in bytes. For a queue in bytes, the queue has a fixed number of *bytes*, and an almost-full queue might have room for a small packet but not for a large one. Thus, for a simulation with a Drop-Tail queue in bytes, large packets are more likely to be dropped than are small ones. The NS simulator doesn't yet have a more realistic intermediate model, where the queue has a fixed number of buffers, each buffer has a fixed number of bytes, and each packet would require one or more free buffers. In this model, a small packet would use one buffer, while a larger packet would require several buffers.

The scenarios in Table 13 are identical to those in Table 12, except that the Drop-Tail queue is in units of bytes instead of packets. Thus, five TCP connections and five TFRC-SP connections compete with web traffic in a topology with a 3 Mbps shared link, with each TFRC-SP application generating 200-byte data packets every 10 ms, for a maximum data rate of 160 Kbps. The number of web sessions is varied to create different levels of congestion. However, instead of Drop-Tail queues able to accommodate at most a hundred packets, the routers' Drop-Tail queues are each able to accommodate at most 50,000 bytes (computed in NS as a hundred packets times the mean packet size of 500 bytes).

As Table 13 shows, with a Drop-Tail queue in bytes, the TFRC-SP flow sees a much smaller packet drop rate than the TCP flow, and as a consequence receives a much larger sending rate. For the simulations in Table 13, the TFRC-SP flows use 200-byte data segments, while the long-lived TCP flows use 1460-byte data segments. For example, when the five TCP flows and five TFRC-SP flows share the link with 800 web sessions, the five TCP flows see an average drop rate of 49% in the second half of the simulation, while the five TFRC-SP flows receive an average drop rate of 8%, and as a consequence receive more than 100 times the throughput of the TCP flows. This raises serious questions about making the assumption that flows with small packets see the same packet drop rate as flows with larger packets. Further work will have to include an investigation into the range of realistic Internet scenarios, in terms of whether large packets are considerably more likely to be dropped than are small ones.

	< - - - - - - Send Rates in Kbps - - - - - >			
Web	TCP (512B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.02	335.05	0.00	185.16
25	0.02	289.94	0.00	185.36
50	0.04	139.99	0.01	184.98
100	0.06	53.50	0.01	184.66
200	0.10	16.14	0.04	167.87
400	0.16	6.36	0.07	114.85
800	0.24	0.90	0.11	67.23
1600	0.42	0.35	0.18	39.32

Table 14: Drop and Send Rates for Drop-Tail Queues in Bytes II: 512B TCP Segments

Table 14 shows that, in these scenarios, the long-lived TCP flows receive a higher packet drop rate than the TFRC-SP flows, and receive considerably less throughput, even when the long-lived TCP flows use 512-byte segments.

To show the potential negative effect of TFRC-SP in such an environment, we consider a simulation with N TCP flows, N TFRC-SP flows, and 10*N web sessions, for N ranging from 1 to 50, so that the demand increases from all types of traffic, with routers with Drop-Tail queues in bytes.

	< - - - - - - Send Rates in Kbps - - - - - >			
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.014	2085.36	0.001	180.29
20	0.040	788.88	0.004	183.87
30	0.074	248.80	0.006	182.94
40	0.113	163.20	0.008	185.33
50	0.127	94.70	0.011	185.14
60	0.177	53.24	0.015	185.30
70	0.174	35.31	0.016	185.07
80	0.221	19.38	0.019	183.91
90	0.188	15.63	0.019	180.42
100	0.265	7.08	0.023	176.71
200	0.324	0.38	0.042	139.48
300	0.397	0.32	0.076	93.53
400	0.529	0.40	0.100	70.40
500	0.610	0.37	0.121	57.59

Table 15: Drop and Send Rates for Drop-Tail Queues in Bytes III:
TFRC-SP, 1460B TCP Segments

	< - - - - - - Send Rates in Kbps - - - - - >			
Web	TCP (1460B seg)		TFRC (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.016	1926.00	0.002	178.47
20	0.030	805.20	0.003	178.23
30	0.062	346.24	0.005	161.19
40	0.093	219.18	0.007	136.28
50	0.110	132.77	0.010	83.02
60	0.170	88.88	0.014	69.75
70	0.149	70.73	0.015	55.59
80	0.213	43.17	0.020	42.82
90	0.188	36.19	0.017	43.61
100	0.233	24.77	0.026	35.17
200	0.311	5.46	0.034	24.85
300	0.367	3.74	0.049	20.19
400	0.421	2.59	0.055	17.71
500	0.459	1.10	0.069	15.95

Table 16: Drop and Send Rates for Drop-Tail Queues in Bytes IV:
Standard TFRC, 1460B TCP Segments

Table 15 shows simulations using TFRC-SP, while Table 16 shows simulations using TFRC instead of TFRC-SP. This is the only difference between the simulations in the two tables. Note that when TFRC-SP is used, the TCP flows and web traffic are essentially starved, while the TFRC-SP flows each continue to send 57 Kbps. In contrast, when standard TFRC is used instead of TFRC-SP for the flows sending 200-byte segments, the TCP flows are not starved (although they still don't receive their "share" of the link bandwidth when their packet drop rates are 30% or higher). These two sets of simulations illustrate the dangers of a widespread deployment of TFRC-SP in an environment where small packets are less likely to be dropped than large ones.

B.4. Packet-dropping Behavior at Routers with AQM

As expected, the packet-dropping behavior also can be varied by varying the Active Queue Management (AQM) mechanism in the router. When the routers use RED (Random Early Detection), there are several parameters than can affect the packet-dropping or marking behavior as a function of the packet size.

First, as with Drop-Tail, the RED queue can be in units of either packets or bytes. This can affect the packet-dropping behavior when RED is unable to control the average queue size, and the queue overflows.

Second, and orthogonally, RED can be configured to be either in packet mode or in byte mode. In packet mode, each *packet* has the same probability of being dropped by RED, while in byte mode, each *byte* has the same probability of being dropped. In packet mode, large-packet and small-packet flows receive roughly the same packet drop rate, while in byte mode, large-packet and small-packet flows with the same throughput in bps receive roughly the same *number* of packet drops. [EA03] assessed the impact of byte vs. packet modes on RED performance.

The simulations reported below show that for RED in packet mode, the packet drop rates for the TFRC-SP flows are similar to those for the TCP flows, with a resulting acceptable throughput for the TFRC-SP flows. This is true with the queue in packets or in bytes, and with or without Adaptive RED (discussed below). As we show below, this fairness between TCP and TFRC-SP flows does not hold for RED in byte mode.

The third RED parameter that affects the packet-dropping or marking behavior as a function of packet size is whether the RED mechanism is using Standard RED or Adaptive RED; Adaptive RED tries to maintain the same average queue size, regardless of the packet drop rate. The use of Adaptive RED allows the RED mechanism to function more effectively in the presence of high packet drop rates (e.g., greater than 10%). Without Adaptive RED, there is a fixed dropping threshold, and all arriving packets are dropped when the dropping or marking rate exceeds this threshold. In contrast, with Adaptive RED, the dropping function is adapted to accommodate high-drop-rate regimes. One consequence is that when byte mode is used with Adaptive RED, the byte mode extends even to high-drop-rate regimes. When byte mode is used with standard RED, however, the byte mode is no longer in use when the drop rate exceeds the fixed dropping threshold (set by default to 10% in the NS simulator).

In the simulations in this section, we explore the TFRC-SP behavior over some of this range of scenarios. In these simulations, as in Section B.3 above, the application for the TFRC-SP flow uses 200-byte data packets, generating 100 packets per second.

	< - - - - - - Send Rates in Kbps - - - - - >			
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.05	305.76	0.04	182.82
25	0.06	224.16	0.06	175.91
50	0.09	159.12	0.08	152.51
100	0.13	90.77	0.11	106.13
200	0.14	48.53	0.14	70.25
400	0.20	22.08	0.19	41.50
800	0.27	3.55	0.25	17.50
1600	0.42	1.87	0.34	8.81

Table 17: Drop and Send Rates for RED Queues in Packet Mode

For the simulations in Table 17, with a congested router with a RED queue in packet mode, the results are similar to those with a Drop-Tail queue in packets, as in Table 12 above. The TFRC-SP flow receives similar packet drop rates as the TCP flow, though it receives higher throughput in the more congested environments. The simulations are similar with a RED queue in packet mode with the queue in bytes, and with or without Adaptive RED. In these simulations, TFRC-SP gives roughly the desired performance.

	< - - - - - - Send Rates in Kbps - - - - >			
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.06	272.16	0.02	184.37
25	0.07	175.82	0.02	184.06
50	0.10	75.65	0.04	180.56
100	0.14	38.98	0.07	151.65
200	0.19	16.66	0.11	106.80
400	0.26	4.85	0.15	69.41
800	0.35	3.12	0.20	27.07
1600	0.42	0.67	0.29	10.68

Table 18: Drop and Send Rates for RED Queues in Byte Mode

Table 18 shows that with a standard RED queue in byte mode instead of packet mode, there is a somewhat greater difference between the packet drop rates of the TCP and TFRC-SP flows, particularly for lower packet drop rates. For the simulation in Table 18, the packet drop rates for the TCP flows can range from 1 1/2 to four times greater than the packet drop rates for the TFRC-SP flows. However, because the TFRC-SP flow has an upper bound on the sending rate, its sending rate is not affected in the lower packet-drop-rate regimes; its sending rate is only affected in the regimes with packet drop rates of 10% or more. The sending rate for TFRC-SP in the scenarios in Table 18 with higher packet drop rates are greater than desired, e.g., for the scenarios with 400 web sessions or greater. However, the results with TFRC-SP are at least better than that of small-packet flows with no congestion control at all.

	< - - - - - - Send Rates in Kbps - - - - >			
Web	TCP (512B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.01	337.86	0.01	184.06
25	0.02	258.71	0.01	184.03
50	0.02	184.71	0.01	183.99
100	0.04	63.63	0.03	184.43
200	0.08	28.95	0.06	149.80
400	0.12	17.03	0.10	88.21
800	0.24	5.94	0.15	36.80
1600	0.32	3.37	0.21	19.45

Table 19: Drop and Send Rates for RED Queues in Byte Mode

Table 19 shows that with a standard RED queue in byte mode and with long-lived TCP flows with 512-byte data segments, there is only a moderate difference between the packet drop rate for the 552-byte TCP packets and the 240-byte TFRC-SP packets. However, the sending rates for TFRC-SP in the scenarios in Table 19 with higher packet drop rates are still greater than desired, even given the goal of fairness with TCP flows with 1500-byte data segments instead of 512-byte data segments.

	< - - - - - - Send Rates in Kbps		- - - - - >	
Web	TCP (1460B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.04	318.10	0.02	185.34
25	0.08	175.34	0.03	184.38
50	0.10	81.60	0.04	181.95
100	0.12	28.51	0.05	178.79
200	0.20	3.65	0.06	173.78
400	0.27	1.44	0.08	161.41
800	0.40	0.58	0.06	159.62
1600	0.55	0.29	0.02	180.92

Table 20: Drop and Send Rates with Adaptive RED Queues in Byte Mode

For the simulations in Table 20, the congested router uses an Adaptive RED queue in byte mode.

For this router, the output queue is in units of bytes rather than of packets, and each *byte* is dropped with the same probability. Also, because of the use of Adaptive RED, this byte-dropping mode extends even for the high-packet-drop-rate regime.

As Table 20 shows, for a scenario with Adaptive RED in byte mode, the packet drop rate for the TFRC-SP traffic is *much* lower than that for the TCP traffic, and as a consequence, the sending rate for the TFRC-SP traffic in a highly congested environment is *much* higher than that of the TCP traffic. In fact, in these scenarios the TFRC-SP congestion control mechanisms are largely ineffective for the small-packet traffic.

The simulation with 1600 web servers is of particular concern, because the TCP packet drop rate increases, while the TFRC-SP packet drop rate decreases significantly. This is due to a detail of the Adaptive RED implementation in the NS simulator, and not about the dynamics of TFRC-SP. In particular, Adaptive RED is configured not to "adapt" to packet drop rates over 50%. When the packet drop rate is at most 50%, Adaptive RED is moderately successful in keeping the packet drop rate proportional to the packet size. TCP packets are six times larger than the TFRC-SP packets (including headers), so the TCP packets should see a packet drop rate roughly six times larger.

But for packet drop rates over 50%, Adaptive RED is no longer in this regime, so it is no longer successful in keeping the drop rate for TCP packets at most six times the drop rate of the TFRC-SP packets.

We note that the unfairness in these simulations, in favor of TFRC-SP, is even more severe than the unfairness shown in Table 13 for a Drop-Tail queue in bytes. At the same time, it is not known if there is any deployment in the Internet of any routers with Adaptive RED in byte mode, or of any AQM mechanisms with similar behavior; we don't know the extent of the deployment of standard RED, or of any of the proposed AQM mechanisms.

	< - - - - - - Send Rates in Kbps		- - - - - >	
Web	TCP (512B seg)		TFRC-SP (200B seg)	
Sessions	DropRate	SendRate	DropRate	SendRate
10	0.01	306.56	0.01	185.11
25	0.02	261.41	0.01	184.41
50	0.02	185.07	0.01	184.54
100	0.04	59.25	0.03	181.58
200	0.08	16.32	0.06	150.87
400	0.12	11.53	0.10	98.10
800	0.25	5.85	0.15	46.59
1600	0.32	1.43	0.22	19.40

Table 21: Drop and Send Rates for Adaptive RED Queues in Byte Mode

Table 21 shows that when TFRC-SP with 240-byte packets competes with TCP with 552-byte packets in a scenario with Adaptive RED in byte mode, the TFRC-SP flows still receive more bandwidth than the TCP flows, but the level of unfairness is less severe, and the packet drop rates of the TCP flows are at most twice that of the TFRC-SP flows. That is, the TFRC-SP flows still receive more than their share of the bandwidth, but the TFRC-SP congestion control is more effective than that in Table 20 above.

Appendix C. Exploring Possible Oscillations in the Loss Event Rate

TFRC-SP estimates the loss interval size differently for short and long loss intervals, counting only one loss event for long loss intervals, but counting all packet losses as loss events for the short loss intervals. One question that has been raised is whether this can lead to oscillations in the average loss event rate in environments where there are many packet drops in a single loss event, and loss events switch from short to long and vice versa. As an example, consider a loss interval with N packets, including N/4 losses. If this loss interval is short (at most two round-trip times), the loss interval length is measured as 4 packets. If this loss interval is long, then the loss interval length is measured as N packets.

If the loss interval switching from short to long and back leads to severe oscillations in the average packet drop rate, and therefore in the allowed sending rate, one solution would be to have a more gradual transition between the calculation of the loss interval length for short and long loss intervals. For example, one possibility would be to use all of the packet losses for a loss interval of one round-trip time in calculating the loss interval length, to use 2/3 of the packet losses from a loss interval of two round-trip times, to use 1/3 of the packet losses from a loss interval of three round-trip time, and to use only one packet loss from a loss interval of four or more round-trip times. This more gradual mechanism would give a transition to counting all losses for a loss interval of only one round-trip time, and counting only one loss event for a loss interval of four or more round-trip times.

However, our simulations so far have not shown a great difference in oscillations in the estimate loss event rate between the default mechanism for estimating the loss interval length for short loss interval and the gradual mechanism described above. Simulation scripts are available from [VOIPSIMS]. Therefore, for now we are staying with the simple default mechanism for estimating the loss event rate for short loss intervals described in this document.

Appendix D. A Discussion of Packet Size and Packet Dropping

The lists below give a general summary of the relative advantages of packet-dropping behavior at routers independent of packet size, versus packet-dropping behavior where large packets are more likely to be dropped than small ones.

Advantages of Packet Dropping Independent of Packet Size:

1. Adds another incentive for end nodes to use large packets.

2. Matches an environment with a limitation in pps rather than bps.

Advantages of Packet Dropping as a Function of Packet Size:

1. Small control packets are less likely to be dropped than are large data packets, improving TCP performance.

2. Matches an environment with a limitation in bps rather than pps.

3. Reduces the penalty of TCP and other transport protocols against flows with small packets (where the allowed sending rate is roughly a linear function of packet size).

4. A queue limited in bytes is better than a queue limited in packets for matching the worst-case queue size to the worst-case queueing delay in seconds.

Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

Informative References

[EA03] W. Eddy and M. Allman. A Comparison of RED's Byte and Packet Modes, Computer Networks, 42(2), June 2003.

[P04] T. Phelan, TFRC with Self-Limiting Sources, October 2004, <http://www.phelan-4.com/dccp/>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC879] Postel, J., "TCP maximum segment size and related topics", RFC 879, November 1983.

[RFC1144] Jacobson, V., "Compressing TCP/IP headers for low-speed serial links", RFC 1144, February 1990.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

[RFC3714] Floyd, S. and J. Kempf, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.

[RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4342] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.

[RFC3448bis] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", Work in Progress, March 2007.

[S05] Peter Sholander, private communications, August 2005. Citation for acknowledgement purposes only.

[V00] P. Vasallo. Variable Packet Size Equation-Based Congestion Control. ICSI Technical Report TR-00-008, April 2000, <http://www.icsi.berkeley.edu/cgi-bin/pubs/publication.pl?ID=001183>

[VOIPSIMS] Web page <http://www.icir.org/tfrc/voipsims.html>.

[WBL04] J. Widmer, C. Boutremans, and Jean-Yves Le Boudec. Congestion Control for Flows with Variable Packet Size, ACM CCR, 34(2), 2004.

Authors' Addresses

Sally Floyd
ICSI Center for Internet Research
1947 Center Street, Suite 600
Berkeley, CA 94704
USA

EMail: floyd@icir.org

Eddie Kohler
4531C Boelter Hall
UCLA Computer Science Department
Los Angeles, CA 90095
USA

EMail: kohler@cs.ucla.edu

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.