Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-dccp-tfrc-voip-02.ps
Expires: January 2006

Sally Floyd
ICIR
Eddie Kohler
UCLA
19 July 2005

TCP Friendly Rate Control (TFRC) for Voice: VoIP Variant

Status of this Memo

Copyright Notice

Abstract

TCP-Friendly Rate Control (TFRC) is a congestion control mechanism for unicast flows operating in a best-effort Internet environment [RFC 3448]. This document proposes a VoIP variant to TFRC. 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. The VoIP variant of TFRC is designed for applications that send small packets, where the design goal is to achieve the same bandwidth in bps as a TCP flow using 1500-byte data packets. This variant is referred to in RFC 3448 as TFRC-PS, for applications that might vary their packet size in response to congestion. The VoIP variant of TFRC enforces a Min Interval of 10 ms between data packets, to prevent a single flow from sending small packets arbitrarily frequently.

Flows using the VoIP variant of TFRC 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 DropTail queues in units of bytes), the current VoIP variant of TFRC can receive considerably more than its

share of the bandwidth. (We note however that in all scenarios the VoIP variant of TFRC is better, in terms of congestion in the network, than the same application in the absence of congestion control).

TO BE DELETED BY THE RFC EDITOR UPON PUBLICATION:

Changes from draft-ietf-dccp-tfrc-voip-01.txt

* Added modified algorithm for calculating the loss event rate, for short intervals with multiple packet drops.
* Moved Faster Restart to a separate document.
* Added simulations with a configured byte drop rate.
* Added many more simulations, including DropTail with a queue in bytes.
* Added a discussion of unfairness for DropTail with a queue in bytes.

Changes from draft-ietf-dccp-tfrc-voip-00.txt

* Added more simulations.
* Added a Related Work section.

Table of Contents

1. 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 [RFC 2119].

2. VoIP Variant Introduction

This document specifies a VoIP variant for TCP-friendly rate control (TFRC) [RFC 3448]. 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.

The VoIP variant is intended for flows that need to send frequent small packets (limited by a minimum interval between packets of 10 ms). Conventional TFRC measures loss rates by estimating the loss event ratio as described in [RFC 3448], and uses this loss event rate to determine the sending rate in packets per round-trip time. This has consequences for the rate 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 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.

The VoIP variant of TFRC described here 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). Given this restriction, the VoIP variant effectively calculates the TFRC fair rate as if the bottleneck restriction was in bytes per second. Applications using the VoIP variant of TFRC could have a fixed packet size, or could vary their packet size in response to congestion.

The VoIP variant of TFRC 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 [RFC 3714] to the congestion control mechanisms of TFRC, it seems acceptable to standardize a variant of TFRC that allows VoIP 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 a small modification to TFRC, as described below.

However, 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, the VoIP variant of TFRC includes the limitation of the Min Interval between packets of 10 ms.

The VoIP variant of TFRC essentially assumes that the small-packet VoIP TFRC flow receives roughly the same packet drop rate as a large-packet TFRC or TCP flow. As we show, this assumption is not correct for all of the simulations in this document.

The VoIP variant of TFRC requires a modification in TFRC's calculation of the loss event rate, because a VoIP TFRC 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 VoIP TFRC 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, VoIP TFRC 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.

3. VoIP Variant Congestion Control

TFRC uses the TCP throughput equation given in Section 3.1 of [RFC 3448], which gives the allowed sending rate X in bytes per second as a function of the loss event rate, packet size, and round-trip time. [RFC 3448] specifies that the packet size s used in the throughput equation should be the packet size used by the application, or the estimated mean packet size if there are variations in the packet size depending on the data. This gives rough fairness with TCP flows using the same packet size.

The VoIP variant changes this behavior in the following ways.

- The nominal packet size: The nominal packet size s is set to 1460 bytes. Following [RFC 3714], 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.
- 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 packet size for the connection in bytes, excluding the transport and network headers.

The H parameter is set to the constant 40 bytes. Thus, if the VoIP TFRC 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 the VoIP variant 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 DCCP connection [DCCP] 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.

- 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, VoIP TFRC 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 VoIP TFRC, 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.

- A minimum interval between packets: The VoIP variant of TFRC 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 the VoIP variant. The motivation for this is discussed below.

4. VoIP Variant Discussion

4.1. The TCP Throughput Equation

The VoIP variant of TFRC uses the TCP throughput equation given in [RFC 3448]. As shown in Table 1 of [RFC 3714], for high packet drop rates, this throughput equation gives rough fairness with most aggressive possible current TCP: a SACK TCP flow using timestamps and ECN. Because it is not recommended for routers to use ECN-marking in highly-congested environments (e.g., with packet drop rates greater than 10%), 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.

4.2. Accounting for Header Size

[RFC 3714] 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, the VoIP variant of TFRC 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 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 the VoIP variant cannot send more than 100 packets per second.)

The VoIP variant 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, because using the exact header size in bytes would have little additional benefit. The VoIP variant's use of an assumed 40-byte header is sufficient to get a rough estimate of the throughput, and to give the application some incentive not to use unnecessarily-many small packets. Because we are only aiming at rough fairness, and at a rough incentive for applications, the use of a 40-byte header in the calculations of the header bandwidth seems sufficient.

4.3. The VoIP 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.

The VoIP variant for TFRC, then, includes a Min Interval of 10 ms. This provides additional restrictions on the use of unnecessarily many small packets.

One justification for the Min Interval is the practical one 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 judgement 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 unnecessarily many small packets.

The Min Interval causes the VoIP variant of TFRC 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 was sending at most 100 pps, it would be able to use the VoIP variant of TFRC. If the flow wished to increase its sending rate to more than 100 pps, but to keep the same packet size, it would no longer be able to achieve this with the VoIP variant to TFRC, 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, and 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. But even without a mechanism to prevent flows from subverting the Min Interval by opening multiple parallel connections, it seems useful to include the Min Interval in the VoIP variant of TFRC.

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 the VoIP variant of TFRC, 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, the VoIP variant of TFRC could be sending N small packets (where N small packets equal one large 1500-byte packet). The loss measurement used with the VoIP variant of TFRC has to be able to detect a connection that is sending multiple packets per round-trip time in the face of multiple packet losses or marks per round-trip time.

In the VoIP variant of TFRC, 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 as N. The average loss interval I_mean is still averaged over the most recent eight 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.

5. A Comparison with RFC 3714

6. The VoIP Variant with Applications that Modify the Packet Size

To be done.

7. Simulation Results

7.1. Simulations with Configured 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, VoIP TFRC 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	TCP SendRate	ECN TCP SendRate	TFRC SendRate
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
0.066	176.40	178.16	211.05
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

Table 1:
"Total Sending Rates (Kbps) for Configured Packet Drop Rates. The TFRC flow uses 1460-byte data packets, with a maximum data sending rate of 1000 Kbps."

* Note: These ECN scenarios are not realistic, as routers are not likely to mark packets when packet drop/mark rates are 20% or higher.

Table 1 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 packets, 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. 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, but this would not be relevant in practice, as routers are advised to drop rather than mark packets during high levels of congestion.

Packet DropRate	TCP SendRate	ECN TCP SendRate	VoIP TFRC SendRate	Stnd TFRC SendRate
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
0.066	174.07	184.37	17.69	6.46
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

Table 2:
"Total Sending Rates (Kbps) for Configured Packet Drop Rates. The TFRC flows use 14-byte data packets, with a maximum data sending rate of 5.6 Kbps."

* Note: These ECN scenarios are not realistic, as routers are not likely to mark packets when packet drop/mark rates are 20% or higher.

Table 2 shows the results of simulations where each VoIP TFRC 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.

The TFRC sending rate in Table 2 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 2 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 VoIP TFRC flow is relatively close to the desired goal of fairness in bps with TCP.

Table 2 shows that with VoIP TFRC, the 5.6 Kbps data stream doesn't reduce its sending rate until packet drop rates greater than 20%, as desired. With packet drop rates of 30-40%, the sending rate for the VoIP TFRC 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 VoIP TFRC flow is somewhat greater than that of the average large-packet TCP flow.

We note that the high sending rate for ECN TCP in environments with high marking rates is largely irrelevant, as routers are advised to drop rather than mark ECN-capable packets in times of high congestion. The sending rate of a TCP connection using timestamps is similar to the sending rate shown for a standard TCP connection without timestamps.

Byte DropRate	TCP SendRate	ECN TCP SendRate	VoIP TFRC SendRate	Stnd TFRC SendRate
0.00001	423.02	406.44	17.69	17.69
0.0001	117.41	114.34	17.69	17.69
0.001	0.41	3.38*	17.69	8.37
0.005	0.26	0.26*	18.39	1.91
0.01	0.31	0.26*	7.07	0.84
0.02	0.29	0.26*	1.61	0.43
0.04	0.12	0.26*	0.17	0.12
0.05	0.15	0.26*	0.08	0.06

Table 3:
"Total Sending Rates (Kbps) for Configured Byte Drop Rates.
The TFRC flows use 14-byte data packets, with a maximum
data sending rate of 5.6 Kbps."

* Note: These ECN scenarios are not realistic, as routers are not likely to mark packets when packet drop/mark rates are 20% or higher.

Byte DropRate	TCP Pkt DropRate	TFRC Pkt DropRate	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	1.0	0.43	2.32
0.05	1.0	0.94	1.05

Table 4:
"Converting Byte Drop Rates to Packet Drop Rates,
for 1500-byte TCP packets and 56-byte TFRC packets."

In contrast, Table 3 shows the TCP and TFRC send rates for various byte drop rates. For these simulations, for each *byte*, the packet containing that byte is dropped with probability p. Table 4 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 last column of Table 3 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 3 shows. with moderate byte drop rates (between 0.1% and 4%), even the standard TFRC flow with 14-byte data packets has higher throughput than the large-packet TCP flow. In these regimes, the packet drop rate for TCP is greater than 50%, and the TCP sending rate no longer varies as 1/sqrt(p), as it does with smaller packet drop rates. With these equal byte drop rates for the TCP and TFRC flows, the VoIP variant of TFRC makes the unfairness in favor of TFRC even worse, with the VoIP TFRC flow receiving significantly higher throughput than the TCP flow for byte drop rates from 0.1% to 2%.

Packet DropRate	TCP SendRate	ECN TCP SendRate	VoIP TFRC SendRate	Stnd TFRC SendRate
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
0.066	179.04	184.68	168.51	31.16
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

Table 5:
"Total Sending Rates (in Kbps) for Configured Packet Drop Rates. The TFRC flows use 200-byte data packets, with a maximum data sending rate of 160 Kbps."

* Note: These ECN scenarios are not realistic, as routers are not likely to mark packets when packet drop/mark rates are 20% or higher.

Using configured packet drop rates, Table 5 compares the average per-flow sending rates when the TFRC flow has a maximum data sending rate of 160 Kbps, with the application generating 200-byte data packets at 100 packets per second. As expected with equal packet drop rates, the performance of Standard TFRC is quite poor, while the performance of VoIP TFRC is essentially as desired for packet drop rates up to 30%. Again as expected, with packet drop rates of 40-50% the VoIP TFRC sending rate is somewhat higher than desired.

In general, Tables 2 and 5 show acceptable performance for VoIP TFRC in environments with stable packet drop rates, where the decision to drop a packet is independent of the packet size. However, in realistic 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 7.2 and 7.3 show simulations with packets dropped at Drop Tail or AQM queues, rather that from a probabilistic process.

VoIP mode for TFRC 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 for the simulations in this document will be made available in "http://www.icir.org/tfrc/voipsims.html".

7.2. Packet Dropping Behavior at Routers with DropTail 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, without the VoIP variant, 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 VoIP 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?

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC 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 6:
"Simulation Results with Drop-Tail Queues in Packets."

Table 6 shows the results of the second half of 100-second simulations, with five TCP connections and five VoIP TFRC connections competing with web traffic in a topology with a 3 Mbps shared link. The VoIP TFRC application generates 200-byte data packets every 10 ms, for a maximum data rate of 160 Kbps. The five TCP connections have roundtrip 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 DropTail queue is in units of packets, which each packet in the queue requires a single buffer, regardless of the packet size.

Table 6 shows the average TCP and TFRC sending rates, each averaged over the five flows. As expected, the VoIP TCP flows see similar packet drop rates as the TCP flows, though the VoIP TFRC flows receives higher throughput than the TCP flows with packet drop rates of 25% or higher.

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.06	239.81	0.00	185.19
25	0.09	189.02	0.01	184.95
50	0.14	99.46	0.01	185.07
100	0.20	16.42	0.02	183.77
200	0.26	4.46	0.03	181.98
400	0.29	4.61	0.05	151.88
800	0.49	1.01	0.08	113.10
1600	0.65	0.67	0.12	65.17

Table 7:
"Simulation Results with Drop-Tail Queues in Bytes."

However, the fairness results can change significantly if the DropTail 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. This, 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.

As Table 7 shows, with a DropTail queue in bytes, the VoIP TFRC flow sees a much smaller drop rate than the TCP flow, and as a consequence receives a much larger sending rate. For example, when the five TCP flows and five VoIP TFRC 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 VoIP TFRC 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.

7.3. 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 DropTail, the RED queue can be either in units of packets or of bytes. This can affect the packet dropping behavior when the RED mechanism is unable to control the average queue size, and the queue overflows.

Second, and orthogonally, RED can 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, with small to moderate levels of congestion, large-packet and small-packet flows with the same throughput in bps receive roughly the same *number* of packet drops. The simulations reported in the appendix show that for RED in packet mode, the packet drop rates for the VoIP TFRC flows are similar to those for the TCP flows, with a resulting acceptable throughput for the VoIP TFRC 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 VoIP TFRC 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, where the dropping function is varied as the packet drop rate changes. 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 these high-drop-rate regimes. One consequence is that when byte mode is combined 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 VoIP TFRC behavior over some of this range of scenarios. In this simulations, as in Section 7.2 above, the application for the VoIP TFRC flow uses 200-byte data packets, generating 100 packets per second.

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC 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 8:
"Simulation Results with RED Queues, Packet Mode.
RED in packet mode, queue in packets, standard RED."

For the simulations in Table 8, with a congested router with a RED queue in packet mode, the results are similar to those in Table 6 above. The VoIP TCP 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 the queue in bytes, and with or without Adaptive RED. These results seems generally acceptable.

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.04	286.90	0.01	184.92
25	0.07	192.67	0.02	184.06
50	0.11	92.98	0.04	181.69
100	0.17	57.02	0.07	154.88
200	0.17	9.55	0.11	120.83
400	0.29	7.68	0.16	77.38
800	0.36	1.97	0.23	23.79
1600	0.51	0.87	0.30	11.00

Table 9:
"Simulation Results with RED Queues, Byte Mode.
RED in byte mode, queue in bytes, standard RED."

Table 9 shows that with a standard RED queue in byte mode, there is a somewhat greater different between the packet drop rates between the TCP and VoIP TFRC flows, particularly for lower packet drop rates. For the simulation in Table 9, 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 VoIP TFRC flows. However, because the VoIP TFRC 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. While the sending rate for VoIP TFRC in the scenarios in Table 9 with higher packet drop rates are greater than desired, the results are significantly better than that of VoIP flows with no congestion control at all.

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.04	297.74	0.02	185.06
25	0.07	209.42	0.03	184.06
50	0.10	85.34	0.04	182.30
100	0.16	28.18	0.05	181.17
200	0.19	4.18	0.06	177.70
400	0.31	1.87	0.08	154.40
800	0.29	0.77	0.06	170.01
1600	0.59	0.48	0.02	173.91

Table 10:
"Simulation Results with RED Queues, Byte Mode.
RED in byte mode, queue in bytes, Adaptive RED."

In contrast, for the simulations in Table 10, 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, each *byte* is dropped with the same probability, and because of the use of Adaptive RED, this byte-dropping mode extends even for the high-packet-drop-rate regime.

As Table 10 shows, for a scenario with Adaptive RED in byte mode, the packet drop rate for the VoIP TFRC traffic is *much* lower than that for the TCP traffic, and as a consequence, the sending rate for the VoIP TFRC traffic in a highly congested environment is *much* higher than that of the TCP traffic. In fact, in these scenarios the TFRC VoIP congestion control mechanisms are largely ineffective for the VoIP traffic.

We note that the unfairness in these simulations, in favor of VoIP TFRC, is even greater than the unfairness shown in Table 7 for a DropTail 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 even know the extent of the deployment of standard RED, or or any of the proposed AQM mechanisms.

8. Discussion

The goal of the VoIP variant of TFRC has been for the TCP flows and the VoIP TFRC flows to have rough fairness 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, flows using the VoIP variant of TFRC could receive more bandwidth than competing TCP flows.

The VoIP variant of TFRC limits the sending rate in packet per second. The simulations by Tom Phelan explore how a limitation in sending rate complicates the issue of exploring the fairness between TCP and VoIP TFRC flows.

For applications with a maximum sending rate of 96 Kbps or less, VoIP TFRC only restricts the sending rate when the packet drop rate is fairly high. In this regime, the performance of TFRC is very much determined by the accuracy of the TCP response function in representing the actual sending rate of a TCP connection. In this regime of high packet drop rates, the performance of the TCP connection is very much affected by the TCP algorithm (e.g., SACK or not), by the minimum RTO, by the use or not of Limited Transmit, by the use of timestamps and/or of ECN, and the like. It is good to insure that simulations or experiments exploring fairness include the exploration of fairness with the most aggressive TCP mechanisms conformance with the current standards. Our simulations use SACK TCP with Limited Transmit and with a minimum RTO of 200 ms. Adding the use of timestamps has not made a big difference. We haven't used ECN, because our judgement is that the use of ECN is not advisable in high-packet-dropping regimes.

9. Implementation Issues

TBA

10. Security Considerations

TBA

11. IANA Considerations

There are no IANA considerations in this document.

12. Thanks

We thank Tom Phelan for discussions of the VoIP variant of TFRC and for his paper exploring the fairness between TCP and VoIP TFRC flows. We thank Joerg Widmer for feedback on earlier versions of this draft. We also thank the DCCP Working Group for feedback and discussions.

Normative References

[RFC 2119] S. Bradner. Key Words For Use in RFCs to Indicate Requirement Levels. RFC 2119.

[RFC 2434] T. Narten and H. Alvestrand. Guidelines for Writing an IANA Considerations Section in RFCs. RFC 2434.

[RFC 2581] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581.

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

Informative References

[CCID 3 PROFILE] S. Floyd, E. Kohler, and J. Padhye. Profile for DCCP Congestion Control ID 3: TFRC Congestion Control. draft-ietf-dccp-ccid3-06.txt, work in progress, October 2004.

[DCCP] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol, draft-ietf-dccp-spec-08.txt, work in progress, October 2004.

[JFAS05] A. Jain, S. Floyd, M. Allman, and P. Sarolahti. Quick-Start for TCP and IP. Internet-draft draft-amit-quick-start-04.txt, work in progress, February 2004.

[MAF04] A. Medina, M. Allman, and A. Floyd, Measuring the Evolution of Transport Protocols in the Internet, May 2004, URL "http://www.icir.org/tbit/".

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

[RFC 2861] M. Handley, J. Padhye, and S. Floyd. TCP Congestion Window Validation. RFC 2861, June 2000.

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

[V00] P. Vasallo. Variable Packet Size Equation-Based Congestion Control. ICSI Technical Report TR-00-008, April 2000. URL "http://www.icsi.berkeley.edu/techreports/2000.abstracts/ tr-00-008.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 <floyd@icir.org>
ICSI Center for Internet Research
1947 Center Street, Suite 600
Berkeley, CA 94704
USA

Eddie Kohler <kohler@cs.ucla.edu>
4531C Boelter Hall
UCLA Computer Science Department
Los Angeles, CA 90095
USA

13. Related Work on VoIP 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 VoIP Variant proposed in our document differs from [WBL04] in restricting its attention to flows that send at most 100 packets per second. The VoIP 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). A first step will be to investigate the range of behaviors actually present in today’s networks.

14. Simulation Results with RED Queue Management

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.04	299.95	0.04	184.15
25	0.06	234.00	0.06	176.43
50	0.08	172.42	0.08	147.82
100	0.11	110.83	0.12	89.78
200	0.13	41.90	0.15	62.37
400	0.23	11.38	0.20	25.05
800	0.27	2.88	0.29	12.26
1600	0.34	0.62	0.35	6.78

Table 11:
"Simulation Results with RED Queues, Packet Mode.
RED in packet mode, queue in packets, Adaptive RED."

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.04	319.87	0.03	183.32
25	0.05	246.38	0.05	175.27
50	0.08	148.18	0.07	152.02
100	0.12	96.38	0.11	92.87
200	0.18	43.30	0.17	55.87
400	0.24	12.58	0.21	20.25
800	0.30	4.22	0.27	20.08
1600	0.35	1.34	0.35	7.21

Table 12:
"Simulation Results with RED Queues, Packet Mode.
RED in packet mode, queue in bytes, Adaptive RED."

Web Sessions	TCP DropRate	TCP SendRate	VoIP_TFRC DropRate	VoIP_TFRC SendRate
10	0.04	326.74	0.03	184.79
25	0.05	260.50	0.04	180.28
50	0.08	173.62	0.07	154.04
100	0.11	81.94	0.10	103.87
200	0.16	48.10	0.14	60.50
400	0.25	8.64	0.19	32.55
800	0.35	7.68	0.26	14.62
1600	0.38	0.77	0.36	7.13

Table 13:
"Simulation Results with RED Queues, Packet Mode.
RED in packet mode, queue in bytes, standard RED."

15. A Discussion of Packet Size and Packet Dropping

This section gives a general summary of the relative advantages of packet-dropping behavior 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.
---------------------------------------------------------
```

Full Copyright Statement

Intellectual Property