Limitations of Equation-Based Congestion Control

Injong Rhee and Lisong Xu

***Abstract*—We study limitations of an equation-based congestion control protocol, called TCP-Friendly Rate Control (TFRC). It examines how the three main factors that determine TFRC throughput, namely, the TCP-friendly equation, loss event rate estimation, and delay estimation, can influence the long-term throughput imbalance between TFRC and TCP. Especially, we show that different sending rates of competing flows cause these flows to experience different loss event rates. There are several fundamental reasons why TFRC and TCP flows have different average sending rates, from the first place. Earlier work shows that the convexity of the TCP-friendly equation used in TFRC causes the sending rate difference. We report two additional reasons in this paper: 1) the convexity of $1/x$ where x is a loss event period and 2) different retransmission timeout period (RTO) estimations of TCP and TFRC. These factors can be the reasons for TCP and TFRC to experience initially different sending rates. But we find that the loss event rate difference due to the differing sending rates greatly amplifies the initial throughput difference; in some extreme cases, TFRC uses around 20 times more, or sometimes 10 times less, bandwidth than TCP. Despite these factors influencing the throughput difference, we also find that simple heuristics can greatly mitigate the problem.**

***Index Terms*—Congestion control, equation-based rate control.**

I. Introduction

EQUATION-BASED rate control is being adopted as an Internet standard for congestion control for multimedia streaming and multicast (see [17] and [25]). TCP-Friendly Rate Control (TFRC) [10] is one example of that. However, there are several pieces of anecdotal evidence suggesting significant discrepancy between the throughput[1] achieved by TFRC and that by TCP [3], [9], [22], [27]. A prevailing thought is that the throughput discrepancy does not pose much threat to the Internet. While that notion is debatable, this paper focuses on the reasons why such discrepancy occurs.

Earlier work [24] provides the first set of theoretical reasons on why TFRC sometimes may not give the same throughput as TCP, more precisely, why TFRC throughput can be less than the *target throughput* $f(1/\mathbf{E}[\theta])$ where $f(\cdot)$ is the TCP-friendly equation [18] used by TFRC, and $1/\mathbf{E}[\theta]$ is the average loss event rate expressed in loss event intervals θ ($\mathbf{E}[\theta]$ is the average loss event interval). The target throughput $f(1/\mathbf{E}[\theta])$ is an estimate of the throughput of competing TCP flows, and according to [24], sets an upper bound to TFRC throughput in most operating conditions. The authors call this behavior the *conservativeness* of TFRC and show it is mainly due to the convexity of $1/f(1/x)$. They offer conservativeness as alternative to TCP friendliness and define "when TFRC can be TCP-friendly in the long run and in some case, excessively so" [24].

In general, there are three main factors that determine the throughput of TFRC: the TCP-friendly equation, loss event rates, and network delays (including RTO estimation). In this paper, we examine how some of these factors influence the difference in the throughput of TCP and TFRC. **The main contributions of our work** are as follows. 1) We analytically and empirically show that when competing TCP and TFRC flows on the same bottleneck have different sending rates, their observed loss event rates can be significantly different; lower sending rate flows, irrespective of whether the flows are of TCP or TFRC, can have higher loss event rates. 2) We empirically show that the different loss event rates caused by these differing sending rates can greatly amplify the initial throughput difference. These results may seem not surprising if we can assume a perfect source of bits, with an output rate of y, that verifies a given loss-throughput formula $f(p)$ with equality $y = f(p)$. But unfortunately, there are reasons to believe that $y \neq f(p)$, that is, even if both TCP and TFRC sources are assumed to see the same loss event rate, the equality does not hold. If $y \neq f(p)$, then our work negates the implicit assumption made by the authors of TFRC that TFRC flows competing in the same end-to-end path as TCP flows will "see" the same p as TCP.

There are several factors for the initial sending rate difference of TFRC and TCP flows that triggers the loss event rate difference. The work by Vojnović and Boudec [24] offers one. This paper provides two additional reasons.

First, our analysis based on the convexity theory shows that $f(\mathbf{E}[1/\hat{\theta}])$ is a tighter bound to TFRC throughput where $\hat{\theta}$ is a weighted moving average of θ. Note that $\mathbf{E}[\hat{\theta}] = \mathbf{E}[\theta]$, but $\mathbf{E}[1/\hat{\theta}] \neq 1/\mathbf{E}[\theta]$ because $1/x$ is a convex function of x. Under a low loss rate condition (e.g., $p < 0.03$, RTT $= 0.1$ s, and RTO $= 0.4$ s), $f(\mathbf{E}[1/\hat{\theta}])$ is a lower bound to TFRC throughput (i.e., TFRC throughput lies between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$). Under a high loss rate condition, $f(\mathbf{E}[1/\hat{\theta}])$ is a tighter upper bound to TFRC throughput than $f(1/\mathbf{E}[\theta])$. In most operating conditions, $f(\mathbf{E}[1/\hat{\theta}])$ tracks TFRC throughput much more closely than $f(1/\mathbf{E}[\theta])$. Intuitively, this result indicates that as TFRC uses instantaneous values of $1/\hat{\theta}$ to make instantaneous rate adjustments, its long-term throughput tends to follow a function of $\mathbf{E}[1/\hat{\theta}]$ instead of a function of $1/\mathbf{E}[\hat{\theta}]$. Interestingly enough, the difference between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$ is positively proportional to $\mathbf{VAR}[\hat{\theta}]/\mathbf{E}[\hat{\theta}]^3$ where $\mathbf{VAR}[\hat{\theta}]$ is the variance in estimated loss event interval samples. This latter finding implies that increase in the variance and also in the loss

Manuscript received December 7, 2004; revised May 23, 2005, April 5, 2006, and June 21, 2006; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Zukerman. This work was supported in part by the National Science Foundation under Grants NSF-ANI 0074012 and NSF CAREER ANI-9875651. Part of the results was presented at ACM SIGCOMM'05.

I. Rhee is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534 USA (e-mail: rhee@csc.ncsu.edu).

L. Xu is with the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0115 USA (e-mail: xu@cse.unl.edu).

Digital Object Identifier 10.1109/TNET.2007.893883

[1]In this paper, throughput means the long-term average sending rates.

event rate can drive TFRC throughput further away from the target rate, which is conjectured in [24]. Our contribution for the latter part is to prove it by analysis.

Synthesizing these findings, we conclude that even if the gap between the two bounds, $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$, can be reduced by reducing the variance, since $f(\mathbf{E}[1/\hat{\theta}])$ is a tighter upper bound to TFRC throughput under high loss rates, it does not affect the gap between $f(\mathbf{E}[1/\hat{\theta}])$ and TFRC throughput. Also we can only reduce the variance in measurement, but not the intrinsic variance present in samples caused by the dynamic nature of Internet traffic. Thus, under high loss rates, the sending rate of TFRC can always be lower than the target TCP rate.

Second, TCP and TFRC employ different RTO estimation schemes. TFRC RFC [10] recommends that RTO be set to four times a moving average of RTT. A standard-conformant TCP [20], on the other hand, sets its RTO to a moving average of RTT plus four times the variance of RTT samples, and the RTO should be at least 1 s. Although many commercial TCP implementations adopt a different minimum RTO value, no matter how the minimum value is set, there are cases where TFRC and TCP may end up having different RTO values depending on the network delays; under short network delays, TFRC tends to set its RTO to a smaller value than TCP, and under long network delays, vice versa. The different RTO values cause TCP and TFRC to have different sending rates.

These factors can provide the initial sending rate gap between TCP and TFRC which may be a trigger for TFRC to have a different loss event rate than TCP. Several studies [3], [19], [24] also show that a slowly responsive flow such as TFRC and CBR (e.g., ping, acknowledgments) may get a higher loss event rate than TCP (because of its slower response to transient congestion) but not vice versa. Our analysis provides reasons for both of the cases. Our work further shows that the loss event rate difference has a feedback effect as it further widens the initial throughput difference. In some extreme cases, we observe that TFRC can use over twenty times more bandwidth than TCP and sometimes, ten times less bandwidth than TCP.

A heuristic (or policy) can be applied to artificially correct some ill-behavior of TFRC, or to give some "calculated" advantage to TFRC over TCP since TFRC serves a different class of applications than TCP. We view the RTO estimation technique recommended by TFRC RFC as such a policy. However, our work indicates that any policy decision that changes the sending rate of TFRC to deviate from that of TCP must be done with a great care because the sending rate difference can be greatly amplified by the loss event rate difference (caused by the sending rate difference). In fact, the RTO policy issue provides an excellent showcase to apply our work. We demonstrate that a simple policy change designed using the insights we developed from our study can fix much of the throughput imbalance we observed in practice.

Based on these findings, we close the loop by designing and evaluating several heuristics to mitigate the throughput difference. Surprisingly a very simple heuristic that manipulates the RTO values of TFRC to be always larger than TCP's by some constant factor works the best in which the throughput difference can be kept within 20% on average under all the operating conditions where we have tested.

The remainder of this paper is organized around the above-mentioned analytical results and their experimental verification. Section II describes the definitions and assumptions we make for analysis. Section III describes the network setup for the simulation study in the paper. Section IV discusses the loss event rate difference. Section V discusses the effect of the convexity of $1/x$. Section VI discusses the effect of RTO difference, and Section VII presents the heuristics.

II. Definitions and Assumptions

TFRC uses the following simplified TCP-friendly equation as described in RFC 3448 [10]. Equation (1) is equivalent to the original one [18, Eq. (30)], if $p \leq 2/3\sqrt{2/(3b)}$ (or $p \leq 0.54$ if $b = 1$):

$$f(p) = \frac{s}{t_{\mathrm{RTT}}\sqrt{\frac{2bp}{3}} + t_0 3\sqrt{\frac{3bp}{8}}p(1+32p^2)}. \quad (1)$$

As defined in [18], p is the probability that a packet is lost in a RTT round, given that no previous packet in the same round is lost, is independent of packet loss in earlier rounds. t_{RTT} is the round-trip delay, and t_0 is the retransmission timeout period. b is 2 if delayed acknowledgment is used and 1 otherwise. s is the packet size.

(A1) We assume that all packets in the same end-to-end network path are subject to the same loss probability p, and p is stationary ergodic.

Note that TFRC uses the loss event rate instead of the packet loss rate to calculate its sending rate. As described in [10], a loss event is defined to be one or more lost or ECN-marked packets within one RTT, and the loss event rate is the number of loss events as a fraction of the total number of transmitted packets. This is because (1) models the throughput of TCP/NewReno where all the packets lost in the same congestion window are treated as a single loss event and cause only one window reduction. As described in [11] and [15], this behavior of TCP/NewReno is due to the intuition that all the packets lost within the same window are likely caused by the same instance of congestion and a window reduction by the losses does not take effect until the packets in the reduced window arrive to the congested link—which is one RTT period after the first loss in the window. Penalizing TCP flows more than once for the same instance of congestion as done in the original TCP/Reno [1] has been shown to lower the utilization of network capacity [15]. TCP/NewReno remedies this behavior of TCP/Reno.

Following the notations in [24], we denote θ_n to be the nth loss event interval, which is the number of packets sent between the nth loss event and the $(n+1)$th loss event. Let $\hat{\theta}_n$ denote the weighted average of loss event intervals, which can be obtained as follows: $\hat{\theta}_n = \sum_{l=1}^{L} w_l \theta_{n-l}$. TFRC RFC suggests $L = 8$, and the values of w_l to have the same values for $1 \leq l \leq L/2$, and linearly decrease with l for the other values of l.

The TFRC throughput within interval n is then given by $f(1/\hat{\theta}_n)$. In this paper, we consider only the *basic rate control* of TFRC where TFRC sets its transmission rate to the rate produced by the formula (i.e., $f(1/\hat{\theta}_n)$) at time T_n at which the nth loss event is detected by the source. For the precise definition of the basic control, see [24]. Vojnović and Boudec [24] show for the basic control of TFRC that the long-run throughput $\overline{B}_{\mathrm{TFRC}}$

can be approximated as below if the covariance $\mathrm{COV}(\theta, \hat{\theta}) \approx 0$ (for convenience, we drop the Palm probability notation):

$$\overline{B}_{\mathrm{TFRC}} = \frac{\mathbf{E}[\theta]}{\mathbf{E}\left[\frac{\theta}{f(\frac{1}{\hat{\theta}})}\right]} \approx \frac{1}{\mathbf{E}\left[\frac{1}{f(\frac{1}{\hat{\theta}})}\right]}. \quad (2)$$

There is empirical evidence that $\mathrm{COV}(\theta, \hat{\theta}) \approx 0$ [28]. That is, the successive loss events are occurring almost independently.

We define p_{tfrc} and p_{tcp} to be the average loss event rates experienced by TFRC and TCP respectively. Below we discuss how we measure these values.

Equation (1) is developed based on p, but p cannot be measured directly. To estimate p, Padhye *et al.* [18] count the number of TCP loss indications (triple duplicate acknowledgment, and timeouts) over a long-term period, and divide the result by the total number of TCP packets transmitted over that period. This value is p_{tcp}. In [18], p_{tcp} is used in the place of p in (1) to show that the measured TCP throughput closely follows the TCP equation.

TFRC uses (1), but unfortunately, TFRC can compute neither p_{tcp} nor p. Instead a TFRC flow use p_{tfrc} in place of p which is computed by dividing the number of loss events observed from that flow by the total number of packets transmitted within the observation period [9]. TFRC registers a loss event as follows. The first packet loss is counted as a loss event. Following this, there is a back-off for the duration of an RTT during which no packet loss is counted. The next packet loss after this back-off is counted as another loss event, followed by another RTT back-off, and so forth. We note that $p_{\mathrm{tfrc}} = 1/(\mathbf{E}[\theta])$.

(A2) We assume that $\hat{\theta}_n$ is an unbiased estimator of θ. Thus, $\mathbf{E}[\hat{\theta}] = \mathbf{E}[\theta]$.

(A3) We assume that RTT does not change within a flow.

(A4) We also assume that all the network flows are using the same data packet size. Acknowledgment packets may have different sizes.

We use the *difference ratio* as the main metric for showing difference between two quantities. The difference ratio of quantities a and b is defined to be $\max\{(a-b)/b, (b-a)/a\}.A$ where $A = 1$ if $a \geq b$, and $A = -1$ otherwise. That is, when a is smaller than b, its difference ratio is negative and otherwise it is positive. We use this metric because it treats both negative and positive differences by the same proportion.

III. Simulation Setup

To verify the theoretical findings through experiments, we conduct *ns* simulation. Our setup uses a typical dumbbell topology where each network flow is connected to the bottleneck link through independent access links at both destination and source. The bandwidth and one-way delay of the bottleneck link are set to 15 Mbps and 50 ms unless noted otherwise. The link implements RED with the default adaptive setting and the buffer size is limited to two times the bandwidth delay product. Each TCP and TFRC source and sink are connected through different access links to the bottleneck link and the delays of the access links are randomly varied from 1 to 3 ms to remove any phase effect. We fix the number of TFRC flows to 5 and also have the matching number of TCP flows sharing bottleneck links with the TFRC flows. These flows are used to compare the performance of TCP and TFRC. To observe the behavior of TCP and TFRC under various network loads, we add background long-lived TCP flows to the forward direction. The number of background long-lived TCP flows are varied from 5 to 400. For each run, web traffic is added to the forward and backward directions of the bottleneck link and emulate about 20 to 100 web sessions and the web traffic occupies about 20 to 60% of the bottleneck bandwidth depending on the network load and bandwidth. The web traffic model of *ns* is close to SURGE [4]. To increase dynamics on the bottleneck link, 50 short-term TCP flows with random starting and ending times are added to both directions. Random burst UDP traffic with the Pareto distribution is also added to the forward direction occupying about 1 to 2% of the bottleneck bandwidth. To measure the delay and packet loss rate in the bottleneck link, ping traffic to the forward direction (with 100-ms interval) is added occupying much less than 1% of the bottleneck bandwidth. We run the simulation for 1000 s and took the measurement after the first 200 s.

IV. Impact of Loss Event Rates

Various reasons have been identified by previous studies for the loss event rate difference:

(R1) TFRC reacts to transient congestion more slowly than TCP. Several studies [3], [24] show that this slow responsiveness may cause TFRC to see more loss event rates than TCP.

(R2) Under very low-level multiplexing where only one or two flows coexist, TCP can have a higher loss event rate than TFRC [24].

(R3) Bonald *et al.* [5] show that in the drop tail router, a bursty traffic such as TCP may experience more loss than CBR. This behavior is less pronounced in a RED router (note that the packet loss rate used in [5] is the total number of lost packets over the total number of packets transmitted so it is different from p).

In this section, we show that the loss event rate difference also occurs when and because TCP and TFRC flows competing in the same bottleneck link transmit at different sending rates. In particular, we show that

(R4) If the sending rate of TFRC is "sufficiently" lower than that of TCP, then $p_{\mathrm{tfrc}} > p_{\mathrm{tcp}}$.

(R5) If the sending rate of TFRC is "sufficiently" higher than that of TCP, then $p_{\mathrm{tfrc}} < p_{\mathrm{tcp}}$.

In what follows, we qualify the term "sufficiently."

A. Difference in Average Loss Event Rates

We compare the number of packets sent over a TFRC connection over a measurement period T, to that sent over a TCP connection running in the same end-to-end path. Let T be equal to N round-trip times. Let W_i^{tcp} and W_i^{tfrc}, $i = 1, 2, \ldots, N$, be the numbers of packets sent in the ith round-trip time over the TCP and TFRC connections respectively. Therefore, the numbers of packets sent over the TCP and TFRC sessions, respectively, are given by $N^{\mathrm{tcp}} = \sum_{i=1}^{N} W_i^{\mathrm{tcp}}$ and $N^{\mathrm{tfrc}} = \sum_{i=1}^{N} W_i^{\mathrm{tfrc}}$.

The probability of having no packet losses in a window of size W_i^{tcp} is $(1-p)^{W_i^{\mathrm{tcp}}}$, where p is the probability that a packet is lost independently from losses in the previous RTT rounds (as

defined in Section II). The probability of having a loss event in a window of size W_i^{tcp} is therefore $1-(1-p)^{W_i^{\text{tcp}}}$.

By definition, p_{tcp} is the number of loss events divided by the number of transmitted packets

$$p_{\text{tcp}} = \frac{\sum_{i=1}^{N} 1-(1-p)^{W_i^{\text{tcp}}}}{\sum_{i=1}^{N} W_i^{\text{tcp}}}. \tag{3}$$

In the same way, we can define p_{tfrc} as follows:

$$p_{\text{tfrc}} = \frac{\sum_{i=1}^{N} 1-(1-p)^{W_i^{\text{tfrc}}}}{\sum_{i=1}^{N} W_i^{\text{tfrc}}}. \tag{4}$$

Strictly speaking, for finite N, p_{tcp}, and p_{tfrc} are only an approximation for their corresponding loss event rates since p is ergodic. For sufficiently large N, these quantities converge to the corresponding loss event rates with probability one.

Equation (4) considers TFRC like a window-based protocol although it is rate-based. This is reasonable because TFRC adjusts the sending rate at every RTT just like TCP, a window-based protocol, and $W_i^{\text{tfrc}} = R_i \times \text{RTT}_i$ where R_i is the average sending rate of TFRC during the ith RTT interval (RTT_i).

The following two lemmas are useful in proving our main theorem.

Lemma 4.1: For any q such that $0 < q < 1$, a function

$$F(m,q) = \frac{1+q+q^2+\cdots+q^{m-1}}{m}$$

is a monotonically decreasing function of m.

Proof:

$F(m+1,q) = (1+q+q^2+\cdots+q^{m-1}+q^m)/m+1$.

$F(m+1,q)$ can be rewritten as

$$\frac{1+q^m/m+q+q^m/m+q^2+q^m/m\cdots+q^{m-1}+q^m/m}{m+1}.$$

Since $0<q<1$, $q^i < q^{i-1}$ for $0 \le i \le m$, we have

$$F(m+1,q) < \frac{1+1/m+q+q/m+\cdots+q^{m-1}+q^{m-1}/m}{m+1} = \frac{1+q+\cdots\cdots+q^{m-1}}{m} = F(m,q).$$

■

Lemma 4.2: Let $G(m,q) = F(m,q)\times m = 1+q+q^2+\cdots+q^{m-1}$, and $M = \frac{\sum_{i=1}^{N} m_i}{N}$ for some nonnegative integer sequence m_i, then

$$\frac{\sum_{i=1}^{N} G(m_i,q)}{N} \le G(M,q).$$

Proof: Consider the sequence $\{q^{m_1}, q^{m_2}, \ldots\ldots, q^{m_N}\}$. Its arithmetic mean is given by

$$E_a(q) = \frac{\sum_{i=1}^{N} q^{m_i}}{N}.$$

Its geometric mean is $E_g(q) = q^M$, where $M = (\sum_{i=1}^{N} m_i)/rN$. $E_a(q) \ge E_g(q)$. Also, $0<q<1$, so $(1-q)>0$. Therefore

$$\frac{1-E_a(q))}{1-q} \le \frac{1-E_g(q))}{1-q}$$

but

$$\frac{1-E_a(q))}{1-q} = \frac{\sum_{i=1}^{N} G(m_i,q)}{N}$$
$$\frac{1-E_g(q))}{1-q} = G(M,q)$$

■

Theorem 4.1, below, proves R4 and R5. Let $E_W^{\text{tcp}} = (\sum_{i=1}^{N} W_i^{\text{tcp}})/N$, the mean window size of TCP during T, and $E_W^{\text{tfrc}} = (\sum_{i=1}^{N} W_i^{\text{tfrc}})/N$, the mean on the number of TFRC packets per RTT during T. Let $W_{\max}^{\text{tcp}} = \max\{W_i^{\text{tcp}}, \text{ for } \forall i, 1 \le i \le N\}$ and $W_{\max}^{\text{tfrc}} = \max\{W_i^{\text{tfrc}}, \text{ for } \forall i, 1 \le i \le N\}$, the maximum numbers of packets sent by the TCP and TFRC connections respectively during an RTT period in T. If $E_W^{\text{tcp}} > W_{\max}^{\text{tfrc}}$ (which implies that the sending rate of TCP is larger than that of TFRC), we prove that $p_{\text{tfrc}} > p_{\text{tcp}}$ (i.e., R4). If $E_W^{\text{tfrc}} > W_{\max}^{\text{tcp}}$ (which implies that the sending rate of TFRC is larger than that of TCP), we prove that $p_{\text{tcp}} > p_{\text{tfrc}}$ (i.e., R5).

Theorem 4.1: $p_{\text{tfrc}} > p_{\text{tcp}}$ if $E_W^{\text{tcp}} > W_{\max}^{\text{tfrc}}$, and $p_{\text{tcp}} > p_{\text{tfrc}}$ if $E_W^{\text{tfrc}} > W_{\max}^{\text{tcp}}$.

Proof: It suffices to prove the first case (R4). The second case (R5) can be proved in the same way because TFRC and TCP are treated essentially the same way in the proof.

We have

$$p_{\text{tcp}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{W_i^{\text{tcp}}} (1-p)^{j-1}p}{\sum_{i=1}^{N} W_i^{\text{tcp}}} = \frac{\sum_{i=1}^{N} G(W_i^{\text{tcp}}, 1-p)\times p}{\sum_{i=1}^{N} W_i^{\text{tcp}}}.$$

Therefore, from Lemma 4.2, it is seen that

$$p_{\text{tcp}} \le \frac{G(E_W^{\text{tcp}}, 1-p)\times p}{E_W^{\text{tcp}}}. \tag{5}$$

From Lemma 4.1, we have

$$\frac{N\times G(E_W^{\text{tcp}}, 1-p)\times p}{N\times E_W^{\text{tcp}}} < \frac{N\times G(W_{\max}^{\text{tfrc}}, 1-p)\times p}{N\times W_{\max}^{\text{tfrc}}}. \tag{6}$$

Also from Lemma 4.1

$$\frac{G(W_{\max}^{\text{tfrc}}, 1-p)}{W_{\max}^{\text{tfrc}}} \le \frac{G(W_i^{\text{tfrc}}, 1-p)}{W_i^{\text{tfrc}}}, \quad \forall i, 1 \le i \le N$$
$$G(W_{\max}^{\text{tfrc}}, 1-p)\times W_i^{\text{tfrc}} \le G(W_i^{\text{tfrc}}, 1-p)\times W_{\max}^{\text{tfrc}}.$$

Taking the summation in both sides, we get

$$\sum_{i=1}^{N} G(W_{\max}^{\text{tfrc}}, 1-p)\times W_i^{\text{tfrc}} \le \sum_{i=i}^{N} G(W_i^{\text{tfrc}}, 1-p)\times W_{\max}^{\text{tfrc}}.$$

Simplifying the above, we get

$$\frac{G(W_{\max}^{\text{tfrc}}, 1-p)}{W_{\max}^{\text{tfrc}}} \leq \frac{\sum_{i=i}^{N} G(W_i^{\text{tfrc}}, 1-p)}{\sum_{i=1}^{N} W_i^{\text{tfrc}}}.$$

Then from the above inequality, we can have the following inequality:

$$\frac{N \times G(W_{\max}^{\text{tfrc}}, 1-p) \times p}{N \times W_{\max}^{\text{tfrc}}} \leq \frac{\sum_{i=i}^{N} G(W_i^{\text{tfrc}}, 1-p) \times p}{\sum_{i=1}^{N} W_i^{\text{tfrc}}}. \quad (7)$$

The RHS of (7) is p_{tfrc}. Combining (5)–(7), we can prove $p_{\text{tcp}} < p_{\text{tfrc}}$. ■

Suppose that a CBR flow is competing with TCP in the same end-to-end path. Let μ_{cbr} be the sending rate of this flow during time T and let μ_{tcp} be the average sending rate of TCP during T. We can prove the following case: if $\mu_{\text{tcp}} > \mu_{\text{cbr}}$, then the loss event rate observed by the flow, p_{cbr}, is always higher than p_{tcp}.

Corollary 4.1: $p_{\text{cbr}} > p_{\text{tcp}}$ if $\mu_{\text{cbr}} < \mu_{\text{tcp}}$.

Proof: $E_W^{\text{tcp}} = (\sum_{i=1}^{N} W_i^{\text{tcp}})/N > (\mu_{\text{cbr}} T)/N = W_{\max}^{\text{cbr}}$ where $W_{\max}^{\text{cbr}}$ is the number of packets in any RTT during T sent by a CBR flow, because $\mu_{\text{cbr}} < \mu_{\text{tcp}}$. By Theorem 4.1, the corollary is true. ■

We can also prove that when two such CBR flows A and B are competing and they have different sending rates, then the following is trivially true from Theorem 4.1.

Corollary 4.2: If $\mu_{\text{cbr}}^A > \mu_{\text{cbr}}^B$, then $p_{\text{cbr}}^A < p_{\text{cbr}}^B$.

B. Simulation: Loss Event Rate Difference

In this section, we provide the experimental evidence that sending rate difference between TCP and TFRC can cause the loss event rate difference between them. Since many factors can influence loss event rates, it is difficult to isolate one factor from the others. Thus, to facilitate our discussion, we assume that R1 through R5 discussed at the beginning of this section are the only reasons that could force TCP and TFRC to have different loss event rates.

To remove the effect of R2, we keep our simulation and experimental environments more dynamic by introducing a high level of statistical multiplexing through different types of background traffic and also a large number of competing flows. To remove the effect of R3, we use a RED router for our bottleneck link and check whether all the flows see the same packet loss rates. But controlling and quantifying the effects of R1, R4, and R5 separately from each other are not trivial as these effects may collectively cause the loss event rate difference. For instance, to remove the effect of R1, we have to keep the responsiveness of TCP and TFRC the same (so that any loss event rate difference is caused by the other factors). But that is not possible because TCP is inherently more responsive than TFRC—for instance, TFRC does not reduce its rate by half as TCP does for a loss event. Likewise, removing the effects of R4 and R5 is not trivial because we cannot force TFRC to send at the same rate as TCP since the effect of R1 alone may force TFRC and TCP to have different sending rates.

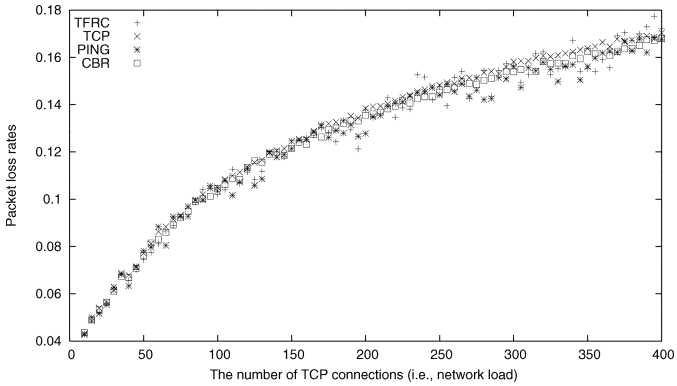

Fig. 1. Packet loss rates of CBR, TFRC, Ping, and TCP traffic. All types of flows are experiencing similar packet loss rates under RED queue. Each run consists of a different number of TCP flows. The x axis represents different runs.

We solve the "separation problem" by using a set of CBR flows, each with a different sending rate. Since our analysis in Section IV-A assumes nothing about the way that the sending rate is controlled, the analysis is still applicable to CBR. Furthermore, CBR flows have the maximum effect of R1 because they do not respond to congestion at all. By mixing TFRC and TCP flows with CBR flows, we can also prove additional properties about loss event rates. We discuss these properties below.

To our simulation setup discussed in Section III, we add additional 14 CBR flows, each with a different sending rate within 1 Mbps to 16 Kbps. The arrival of packets in a CBR flow is randomized without affecting the average sending rate to avoid the phase effect [12]. We have five TFRC flows running for each run of the experiment. As discussed in Section III, the network load is controlled by adjusting the number of long-lived TCP flows. We set the bottleneck bandwidth to 20 Mbps (with aggregate CBR traffic taking up about 2 Mbps), and keep the same background traffic as discussed in Section III. For each run, we measure the loss event and sending rates of TCP, TFRC, and CBR (the loss event rate of CBR is measured in the same way as in TFRC).

We first measure the packet loss rates of different types of flows in each run to make sure that we do not have the effect of R3. The packet loss rate of a flow is obtained from the total number of lost packets divided by the number of packets transmitted in the same way as in [5]. We plot the average values in Fig. 1 for each type of flows in each run which simulates a different network load environment (created by varying the number of long-lived TCP flows). From the figure, we observe that all the flows in the same run experience the same packet loss rates (note that this loss rate is different from p). This happens, as shown in [5], because the bottleneck queue performs the random early drop (RED) policy and is consistent with the finding in [5]. Even if they all have the same packet loss rates, we observe their loss event rates to be quite different. Fig. 2 shows the loss event rates of each flow and Fig. 3 shows the sending rates of TCP, TFRC, and CBR in each run. The sending rate of CBR fluctuates a little because of the randomized arrival process. Evidently, the lower the sending rate of a CBR flow is, the larger its loss event rate is, which confirms Corollary 4.2 .

In all experiments, the CBR flows with smaller sending rates than TCP always have a higher loss event rate than TCP—con-

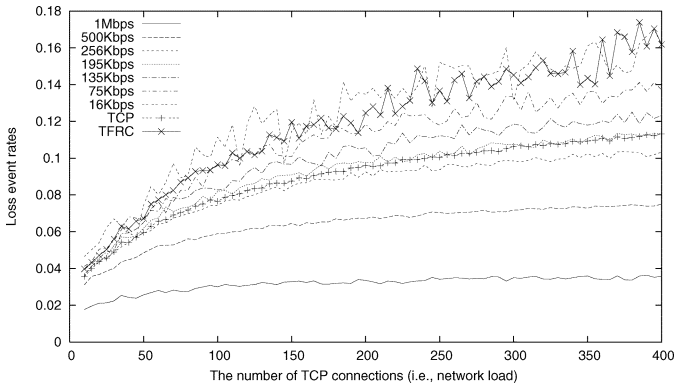

Fig. 2. Loss event rates of CBR, TFRC, and TCP flows. CBR flows with different sending rates experience different average loss event rates. In this figure, we observe the effects of R1, R4, and R5 altogether to cause the loss event rate differences.

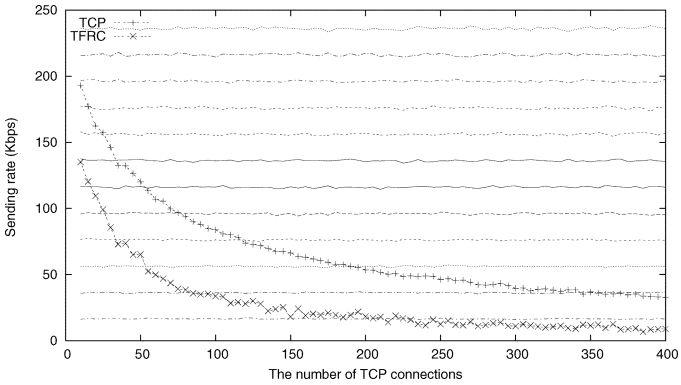

Fig. 3. Sending rates of TCP, TFRC, and CBR in the same run as in Fig. 2. Those flat lines indicate measured CBR rates.

sistent with Corollary 4.1. The flows with a significantly higher sending rate than TCP (in particular, 1-Mbps and 500-Kbps CBR flows) have a lower loss event rate than TCP. Note that Theorem 4.1 requires, in order for TFRC (and CBR) to see a lower loss event rate than TCP, the maximum TCP congestion window size during an observation period T to be always less than the average number of packets per RTT by a TFRC connection. Because of this restriction, unless CBR flows have a significantly higher sending rate than TCP, this behavior is not observed. On the other hand, the 195-Kbps CBR flow gets approximately the same loss event rate as TCP whose sending rate varies from 200 to 30 Kbps. This phenomenon occurs most likely because the effect of R1 cancels out the effect of R5 (if there is any), and these forces somehow maintain a balance around 195 Kbps. We do not know why this particular sending rate creates such a balance. Below we analyze the experiment data further to relate the CBR result to the loss event rate difference between TCP and TFRC.

In Fig. 2, TFRC shows a consistently higher loss event rate, and the difference is increasing as the network load increases. By examining *only* Fig. 2, it is hard to discern whether the lower sending rates of TFRC than TCP (as shown in Fig. 3) contribute to causing the lower loss event rates for TFRC because the effects of R1 and R4 are mixed. To gather more evidence, we conduct the following two measurements. (M1) We measure the loss event rates of CBR flows whose average sending rates are

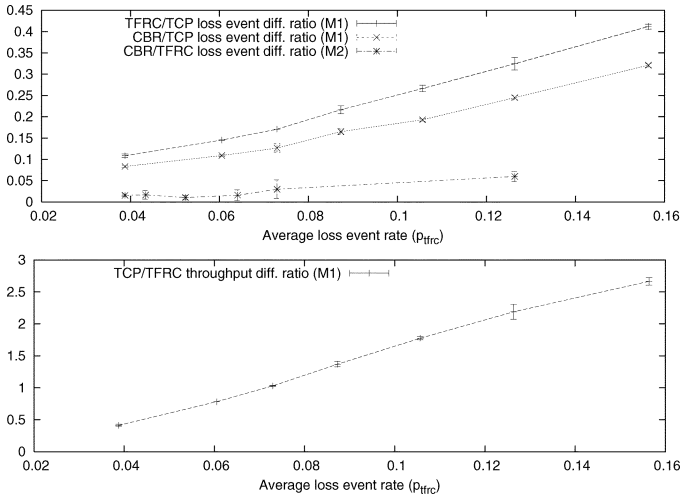

Fig. 4. This figure compares the effect of differing sending rates on the loss event rate difference, and the effect of different responsiveness to congestion on the loss event rate difference. We can see the effects of R1 and R4. It also shows the correlation between the loss event rate difference and the loss event rate, and that between the loss event rate difference and the throughput difference.

approximately the same as TCP (in Fig. 3, those CBR flows intersect with TCP) and also measure the loss event rates of TCP and TFRC (here note that while CBR and TCP have approximately the same sending rate, TFRC does not have the same rate). (M2) We measure the loss event rates of CBR flows whose average sending rates are approximately the same as TFRC (in Fig. 3, those CBR flows intersect with TFRC). The top of Fig. 4 shows the result in which each data point is obtained from an average from 30 runs; error bars correspond to one standard deviation. For instance, in Fig. 3, under the run with 35 long-lived background TCP flows, the sending rates of TCP and the CBR flow (with 165-Kbps rate) are approximately the same. We run this same environment (with different random seeds) for 30 times. These runs have an average p_{tfrc} of 0.038 and the data point for 0.038 in the top of Fig. 4 is obtained from the average values from these 30 runs.

1) Effect of Responsiveness: Let us analyze the effect of R1 from the results of M1 and M2. The top of Fig. 4 shows the loss event difference ratios of CBR and TCP from M1, of CBR and TFRC from M2. In the figure, we find that the CBR flows in M1 experience a higher loss event rate than TCP, and the CBR flows in M2 get a higher loss event rate than TFRC. Note that CBR is less responsive to congestion than TCP and TFRC. These loss event rate differences are likely caused by R1 because there is little effect of differing sending rates (due to the ways M1 and M2 are set up) and the other factors (R2 and R3) are effectively eliminated in the experiment. We note that the loss event rate difference ratio of CBR and TCP is much larger than that of CBR and TFRC. This implies that the degree of responsiveness significantly affects the amount of loss event rate difference because TFRC is less responsive than TCP. That is, the more responsive a flow is, the higher loss rate it gets, which confirms the results from [3] and [24].

2) Effect of Different Sending Rates: If there is no effect of R4 and R5, then irrespective of sending rates, the loss event rates of CBR, TCP, and TFRC must be in the following order: CBR $>$ TFRC $>$ TCP. However, we find in Fig. 4 that the loss

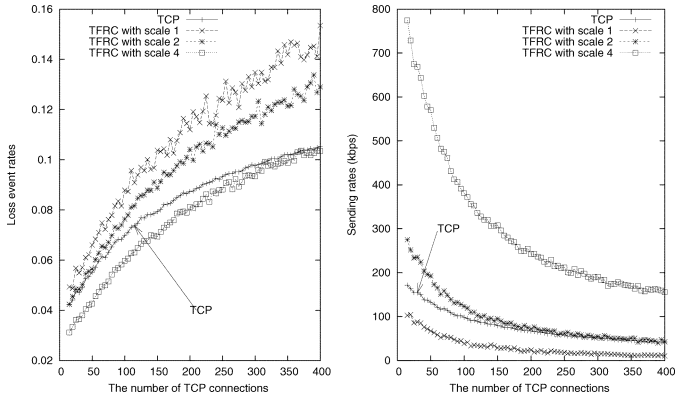

Fig. 5. Loss event rate and sending rate of TFRC with a different scale factor and those of TCP.

event difference ratio of TFRC and TCP is significantly larger than that of CBR and TCP when measured under M1, so we get TFRC $>$ CBR $>$ TCP instead. This implies that there exist other forces in place than R1. In M1, TFRC has a significantly lower sending rate than TCP and CBR (see bottom of Fig. 4 where TFRC has from 2 to 4 times less throughput than TCP). This "inversion" in the loss event difference ratios can be explained by Theorem 4.1, which states the lower sending rate of TFRC can force TFRC to have a higher loss event rate than TCP.

Another interesting observation we make from Fig. 4 is that the loss event rate difference ratio of TFRC and TCP and that of CBR and TCP in M1 are positively correlated to the loss event rates and also to the throughput difference ratio of TCP and TFRC (see bottom of Fig. 4). In other words, the effects of R1 and R4 also tend to increase as the loss rate increases, which is shown by the increasing loss event rate difference ratios of CBR and TFRC, and of TFRC and TCP in M1. We believe that these correlations are why TFRC has a significantly bigger drop in throughput under high loss rates. We need further study to provide a theoretical reason for these correlations.

We have not yet shown the case for R5 where TFRC has a higher sending rate but a lower loss event rate than TCP although the experiments with CBR flows indicate a strong likelihood for R5. In Section VI, we observe this case indeed arises from the practice, but we find that the phenomenon is highly correlated with the RTO estimation of TFRC. Since it requires further explanation about the behavior of TFRC, we defer that discussion to a later section. Instead, we conduct the following simple experiments. To the earlier experimental setup with CBR, we add another type of TFRC flows in which the output of (1) is multiplied by a constant factor at each time a feedback throughput is given to the TFRC sender for rate adjustment; it effectively (and also artificially) increases the sending rate of TFRC. According to R5, these flows must have a lower loss event rate than TCP which is indeed shown in Fig. 5. In the figure, the TFRC flow with scale factor 4 has a distinctively lower loss event rate than TCP as its sending rate gets almost four times larger than TCP. Also we observe that the loss event rates of the TFRC flows tend to increase faster than TCP as the network load increases. It is because the effect of R1 also increases along with loss event rates, as shown in the bottom of Fig. 4. It can be viewed that the "scaled" versions of TFRC will be less responsive to congestion than the original TFRC since it sends at a higher rate and thus would have a higher loss event rate than TCP. Yet, in our measurement, they have a lower loss event rate than TCP (especially, the flow with scale factor 4 under a less than 8% loss event rate). This strongly suggests that the sending rate difference is one of the main causes that TFRC can have a lower loss event rate than TCP in the experiment.

In the following sections, we study why TCP and TFRC can have different sending rates, from the first place, although running in the same environment, and provide empirical evidence suggesting that the loss event difference amplifies the initial sending rate differences.

V. Impact of TCP-Friendly Equation

In Section IV, we showed some theoretical and empirical evidence that the difference in average sending rates causes TFRC and TCP to experience different loss event rates. A natural question is why TCP and TFRC would have different sending rates from the first place. [24] provides one answer for the question; it proves that the convexity of $1/f(1/x)$ makes TFRC conservative in most operating conditions so that $f(1/\mathbf{E}[\theta])$ forms an upper bound to TFRC throughput. However, the conservativeness of TFRC with respect to the target throughput defines only an upper bound on TFRC throughput. An important open problem lies in a lower bound: if conservative, how conservative can TFRC be? A lower bound in combination with an upper bound provides information on how well TFRC tracks the target throughput, and whether the throughput difference is a fundamental property of TFRC.

A. Upper and Lower Bounds on the Long-Term TFRC Throughput

In obtaining the upper and lower bounds of the long-term sending rate of TFRC, $\overline{B}_{\mathrm{TFRC}}$, we heavily use Jensen's inequality. Let

$$f_u(\hat{\theta}) = \frac{1}{f(\frac{1}{\hat{\theta}})} \qquad f_l(\frac{1}{\hat{\theta}}) = \frac{1}{f(\frac{1}{\hat{\theta}})}.$$

Their second derivatives are respectively given by the following:

$$\frac{d^2(f_u(\hat{\theta}))}{d(\hat{\theta})^2} = \frac{3}{4}\sqrt{\frac{2b}{3}}t_{\mathrm{RTT}}\hat{\theta}^{-2.5} + \frac{45}{8}\sqrt{\frac{3b}{2}}t_0\hat{\theta}^{-3.5} + 756\sqrt{\frac{3b}{2}}t_0\hat{\theta}^{-5.5} \quad (8)$$

$$\frac{d^2\left(f_l\left(\frac{1}{\hat{\theta}}\right)\right)}{d\left(\frac{1}{\hat{\theta}}\right)^2} = -\frac{1}{4}\sqrt{\frac{2b}{3}}t_{\mathrm{RTT}}\hat{\theta}^{1.5} + \frac{9}{8}\sqrt{\frac{3b}{2}}t_0\hat{\theta}^{0.5} + 420\sqrt{\frac{3b}{2}}t_0\hat{\theta}^{-1.5}. \quad (9)$$

Equation (8) is greater than zero for any positive $\hat{\theta}$ and therefore, $f_u(\hat{\theta})$ is a convex function of $\hat{\theta}$. Thus, by Jensen's inequality, we have

$$\mathbf{E}[f_u(\hat{\theta})] \geq f_u(\mathbf{E}[\hat{\theta}]). \quad (10)$$

Equation (9) is a monotonically decreasing function of $\hat{\theta}$. Note that the first term on the right-hand side of (9) is negative. For a large value of $\hat{\theta}$, f_l is negative and thus, a concave function of $\hat{\theta}$ and for a small value of $\hat{\theta}$, it is positive and

thus a convex function of $\hat{\theta}$. For instance, for $t_{\mathrm{RTT}} = 100$ ms, $t_0 = 400$ ms, and $b = 1$, it becomes negative when $\hat{\theta}$ is over 35 (i.e., a loss event rate of 0.03). Therefore, by Jensen's inequality, we conclude

$$\begin{cases} \mathbf{E}\left[f_I\left(\frac{1}{\hat{\theta}}\right)\right] \geq f_I\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right) & \text{for small } \hat{\theta} \\ \mathbf{E}\left[f_I\left(\frac{1}{\hat{\theta}}\right)\right] \leq f_I\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right) & \text{for large } \hat{\theta}. \end{cases} \tag{11}$$

Combining (10) and (11), we have

$$\begin{cases} \frac{1}{f\left(\frac{1}{\mathbf{E}[\hat{\theta}]}\right)} \leq \frac{1}{f\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right)} \leq \mathbf{E}\left[\frac{1}{f\left(\frac{1}{\hat{\theta}}\right)}\right] & \text{for small } \hat{\theta} \\ \frac{1}{f\left(\frac{1}{\mathbf{E}[\hat{\theta}]}\right)} \leq \mathbf{E}\left[\frac{1}{f\left(\frac{1}{\hat{\theta}}\right)}\right] \leq \frac{1}{f\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right)} & \text{for large } \hat{\theta} \end{cases} \tag{12}$$

where $1/(f(1/\mathbf{E}[\hat{\theta}])) \leq 1/(f(\mathbf{E}[1/\hat{\theta}]))$ is because f is a decreasing function of $1/\hat{\theta}$, and $\mathbf{E}[1/\hat{\theta}] \geq 1/(\mathbf{E}[\hat{\theta}])$ by Jensen's inequality.

Finally, the upper and lower bounds of the average TFRC throughput can be obtained from (2) and (12) as follows:

$$\begin{cases} \overline{B}_{\mathrm{TFRC}} \leq f\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right) \leq f\left(\frac{1}{\mathbf{E}[\hat{\theta}]}\right) = f\left(\frac{1}{\mathbf{E}[\theta]}\right) & \text{for small } \hat{\theta} \\ f\left(\mathbf{E}\left[\frac{1}{\hat{\theta}}\right]\right) \leq \overline{B}_{\mathrm{TFRC}} \leq f\left(\frac{1}{\mathbf{E}[\hat{\theta}]}\right) = f\left(\frac{1}{\mathbf{E}[\theta]}\right) & \text{for large } \hat{\theta}. \end{cases} \tag{13}$$

From the above, we can see that the inequality between $\overline{B}_{\mathrm{TFRC}}$ and $f(\mathbf{E}[1/\hat{\theta}])$ is controlled by the shape of the function $1/x$ while that between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$ is controlled by the shape of f. In the bounds, the relation between $\overline{B}_{\mathrm{TFRC}}$ and $f(1/\mathbf{E}[\hat{\theta}])$ is consistent with the results from [24]. Note that [24] shows some special cases where TFRC may not be conservative (i.e., $\overline{B}_{\mathrm{TFRC}} > f(1/\mathbf{E}[\theta])$). But they occur when the condition of $\mathrm{COV}(\theta, \hat{\theta}) \approx 0$ does not hold. Since we assume that this condition is true, our analysis does not have those cases. Note that we make this assumption only to derive the upper and lower bounds of TFRC throughput analytically and it is not used for the other results in this paper.

For small $\hat{\theta}$ (i.e., high loss event rates), $f(\mathbf{E}[1/\hat{\theta}])$ is an upper bound for $\overline{B}_{\mathrm{TFRC}}$, and moreover, it is a tighter upper bound than $f(1/\mathbf{E}[\hat{\theta}])$. For large $\hat{\theta}$ (i.e., low loss event rates), $f(\mathbf{E}[1/\hat{\theta}])$ is a lower bound for $\overline{B}_{\mathrm{TFRC}}$, and moreover, our simulation shows that $\overline{B}_{\mathrm{TFRC}}$ is closer to $f(\mathbf{E}[1/\hat{\theta}])$ than to $f(1/\mathbf{E}[\hat{\theta}])$. The results are found in Section V-C.

Overall, $f(\mathbf{E}[1/\hat{\theta}])$ is a more accurate approximation to the average TFRC throughput $\overline{B}_{\mathrm{TFRC}}$ than $f(1/\mathbf{E}[\theta])$. Since the $f(1/\mathbf{E}[\theta])$ is the desired throughput of the TFRC protocol, it is necessary to minimize the difference between those two bounds.

B. Gap Between Bounds

In this subsection, we study the factors that affect the gap between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\hat{\theta}])$.

First, we calculate the difference between $\mathbf{E}[1/\hat{\theta}]$ and $1/\mathbf{E}[\hat{\theta}]$. Let $g(x) = 1/x$, then the Taylor-series expansion of $g(x)$ with respect to point x_0 is given by

$$g(x) = g(x_0) + g'(x_0)(x - x_0) + \frac{1}{2}g''(x^*)(x - x_0)^2 \tag{14}$$

where x^* is a point between x and x_0. Let $x_0 = \mathbf{E}[x]$, and then take expectation on both sides

$$\begin{aligned} \mathbf{E}[g(x)] &= \mathbf{E}[g(\mathbf{E}[x]) + g'(\mathbf{E}[x])(x - \mathbf{E}[x]) \\ &\quad + \frac{1}{2}g''(x^*)(x - \mathbf{E}[x])^2] \\ &= g(\mathbf{E}[x]) + g'(\mathbf{E}[x])(\mathbf{E}[x] - \mathbf{E}[x]) \\ &\quad + \frac{1}{2}\mathbf{E}[g''(x^*)(x - \mathbf{E}[x])^2] \\ &= g(\mathbf{E}[x]) + \frac{1}{2}\mathbf{E}[g''(x^*)(x - \mathbf{E}[x])^2] \\ &\approx g(\mathbf{E}[x]) + \frac{1}{2}\mathbf{E}[g''(\mathbf{E}[x])(x - \mathbf{E}[x])^2] \\ &= g(\mathbf{E}[x]) + \frac{1}{2}g''(\mathbf{E}[x])\mathbf{E}[(x - \mathbf{E}[x])^2] \\ &= g(\mathbf{E}[x]) + \frac{1}{2}g''(\mathbf{E}[x])\mathbf{VAR}[x] \end{aligned} \tag{15}$$

where we approximate $g''(x^*)$ with $g''(\mathbf{E}[x])$, as x^* is a point between x and $\mathbf{E}[x]$.

Substituting $g(x)$ with $1/\hat{\theta}$, we get

$$\mathbf{E}\left[\frac{1}{\hat{\theta}}\right] - \frac{1}{\mathbf{E}[\hat{\theta}]} \approx \frac{\mathbf{VAR}[\hat{\theta}]}{\mathbf{E}[\hat{\theta}]^3}. \tag{16}$$

The difference between $\mathbf{E}[1/\hat{\theta}]$ and $1/\mathbf{E}[\hat{\theta}]$ increases as the variance of $\hat{\theta}$ increases and also as the loss event interval decreases (i.e., the loss event rate increases). Since f is a monotonically increasing function of $\hat{\theta}$, the larger the difference between $\mathbf{E}[1/\hat{\theta}]$ and $1/\mathbf{E}[\hat{\theta}]$, the larger the difference between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$. It implies that as long as some variance in $\hat{\theta}$ exists (which is likely because of the inherent variance in the Internet traffic), the difference between TFRC sending rate $\overline{B}_{\mathrm{TFRC}}$ and the target throughput $1/\mathbf{E}[\hat{\theta}]$ always exists.

C. Simulation Results: Impact of Equation

The objectives of the simulation are twofold. First, our analysis involves some assumptions and approximations. The main assumption of $\mathrm{COV}(\theta, \hat{\theta}) \approx 0$ is observed by several researchers through the Internet measurements [28]. We do not belabor that subject. The other simplifications and approximations regarding the rate control of TFRC (basic versus comprehensive) or in computing the gap between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\hat{\theta}])$ might have some impact on the correctness of the analysis. We show in this section that despite these simplifications and approximations, our analysis is very consistent with the experimental results. Second, our analysis shows only relative bounds and inequality, but does not quantify these bounds. Thus, it is hard to find out whether the sending rate gap caused by the factors identified in Section V has a major impact on the actual throughput difference between TCP and TFRC. Relating the results from Sections V-A and V-B back to those from Section IV, we also verify our conjecture that the loss event rate difference (caused by sending rate difference) can amplify the sending rate difference defined by the bounds.

For all *ns* experiments, we have TFRC set its RTO value in the same way as the standard-conformant TCP in order to eliminate the effect of different RTO estimation techniques on the performance of TFRC. In Section VI, we study how different network

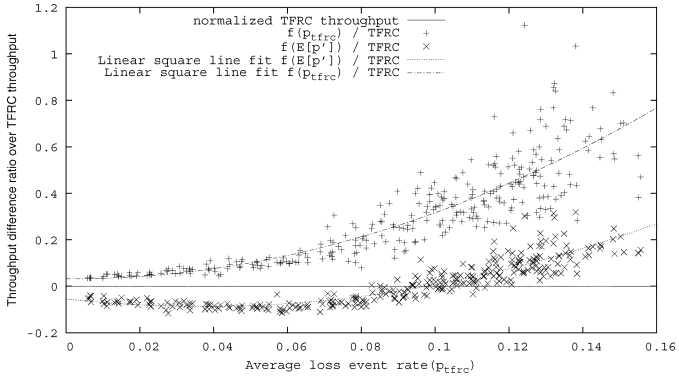

Fig. 6. Per-flow throughput ratio plotted over average loss event rate p. $p' = 1/\theta$ and $p_{\mathrm{tfrc}} = 1/\mathbf{E}[\theta]$. $L = 8$. All the ratios are computed with respect to normalized TFRC throughput.

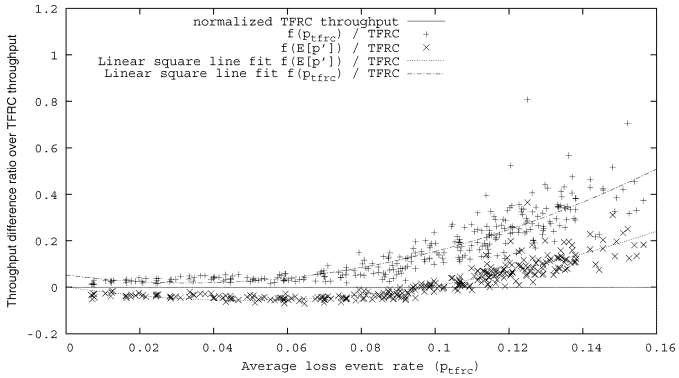

Fig. 7. Same graph as Fig. 6, but with $L = 16$. Note that the difference between $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$ is smaller as L increases, but the difference between TFRC throughput and $f(\mathbf{E}[1/\hat{\theta}])$ does not change much.

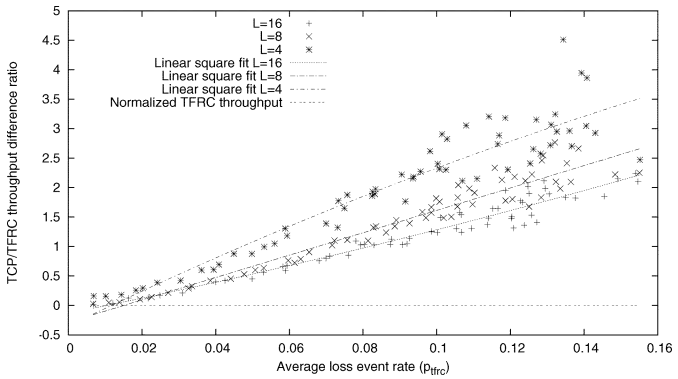

Fig. 8. Average throughput difference ratio of TCP and TFRC. All the ratios are computed with respect to normalized TFRC throughput.

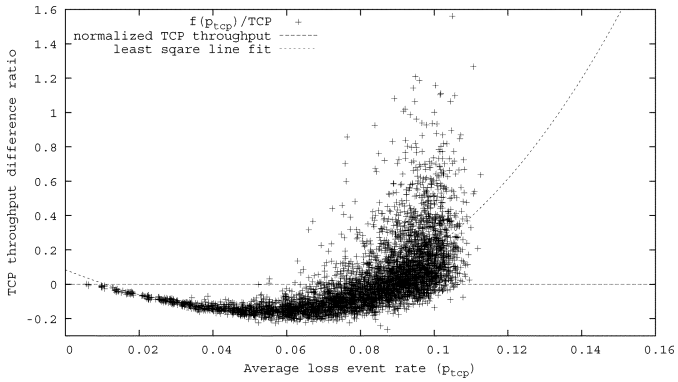

Fig. 9. Difference ratio of TCP-friendly equation $f(p_{\mathrm{tcp}})$ and TCP throughput.

delay estimation techniques influence the throughput difference between TCP and TFRC. This section focuses on the effect of TFRC rate control and the equation.

In the simulation, we measure $1/\mathbf{E}[\hat{\theta}]$ by taking p_{tfrc}; as discussed in Section II, $p_{\mathrm{tfrc}} = 1/\mathbf{E}[\hat{\theta}]$ $(= 1/\mathbf{E}[\theta])$. We measure $\mathbf{E}[1/\hat{\theta}]$ by summing the sample of $1/\hat{\theta}$ at each instance of loss events and in the end of each run, dividing the total by the total number of loss events.

Figs. 6 and 7 show the difference ratio of $f(1/\mathbf{E}[\hat{\theta}])$ and TFRC throughput $\overline{B}_{\mathrm{TFRC}}$, and that of $f(\mathbf{E}[1/\hat{\theta}])$ and $\overline{B}_{\mathrm{TFRC}}$. In the simulation runs that produced Fig. 6, we set the loss event interval averaging window size L to 8 and in Figs. 7–16. The line on 0 indicates the normalized TFRC throughput. The other curves are created by least square line fitting. From the figures, we can see that under low loss rates, $f(\mathbf{E}[1/\hat{\theta}])$ is a lower bound to TFRC throughput but very closely tracks TFRC throughput, while under high loss rates, it becomes a tighter upper bound to TFRC throughput than $f(1/\mathbf{E}[\theta])$. As the loss rate increases, the gap between $f(1/\mathbf{E}[\hat{\theta}])$ and $f(\mathbf{E}[1/\theta])$ increases, confirming that the gap is inversely proportional to $\mathbf{E}[\theta]$. Note that under $L = 16$ (Fig. 7), the gap has significantly decreased from that under $L = 8$, implying that the gap is positively proportional to the variance in $\hat{\theta}$. Another important point to note is that the gap between TFRC throughput and $f(\mathbf{E}[1/\theta])$ also increases along with the loss event rate. We conjecture that this is likely due to the convexity of $1/(f(1/x))$.

Fig. 8 shows the throughput difference ratio of TCP and TFRC as we vary L. As L increases, we observe that the ratio reduces. But the differences are small. One of the reasons to the phenomenon is, as explained in Section V-B, that although the reduced variance reduces the difference between $f(1/\mathbf{E}[\hat{\theta}])$ and $f(\mathbf{E}[1/\theta])$, it does not affect the difference between $f(\mathbf{E}[1/\theta])$ and TFRC throughput. But the difference ratio of TCP and TFRC shown in Fig. 8 is much larger than that of $f(\mathbf{E}[1/\theta])$ and TFRC throughput shown in Figs. 6 and 7. This indicates that there must be other, but more influential, factors (including the loss event rate difference) affecting this gap between TCP and TFRC.

In search for the other reasons for the gap between TCP and TFRC, we first look at how faithful the TCP equation in (1) is in predicting actual TCP throughput. Fig. 9 shows the throughput difference ratio of $f(p_{\mathrm{tcp}})$ and TCP obtained from the simulation runs in Fig. 6. The results are interesting since the equation itself shows significant discrepancy from TCP throughput. As the loss event rate increases, the difference ratio increases—similar to the patterns in Figs. 6 and 7. However, since under high loss event rates, TCP shows smaller throughput than the equation and so does TFRC, this discrepancy must, in fact, reduce the throughput difference between TCP and TFRC. However, Fig. 8 shows it does not; in fact, the throughput difference ratio increases as the loss event rate increases. It is also interesting to note that, the loss event rate of TCP does not increase beyond 0.11 while that of TFRC increases up to 0.16. This implies that TCP and TFRC flows are experiencing different loss event rates even under the same conditions. Thus, Figs. 6 and 9 are not di-

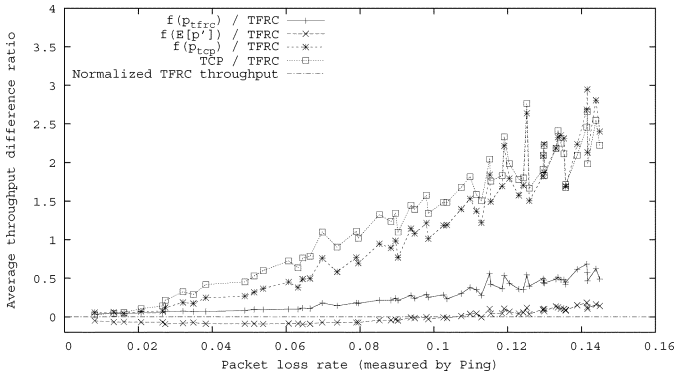

Fig. 10. Average throughput difference ratio with respect to TFRC throughput.

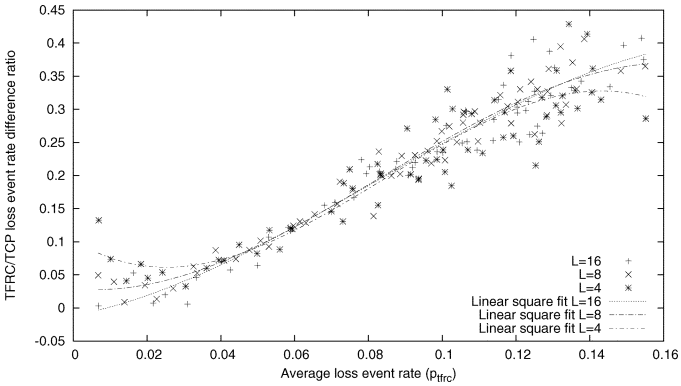

Fig. 11. Average difference ratio of p_{tfrc} and p_{tcp} measured in the same runs as in Fig. 8. The figure shows that TFRC tends to have higher loss rates than TCP and the difference increases as the loss event rate increases.

rectly comparable as the two types of flows see different loss event rates.

For direct comparison, we compute in each run of simulation the average throughput difference ratios for TCP, $f(\mathbf{E}[1/\hat{\theta}])$, $f(1/\mathbf{E}[\theta])$ and $f(p_{\mathrm{tcp}})$, each with respect to the average TFRC throughput. Fig. 10 plots the results over the packet loss rate observed by a ping traffic in each simulation run (because loss event rates cannot be used to compare $f(p_{\mathrm{tcp}})$ and TFRC throughput). All the data points over the same packet loss rate value are from the same run. From this figure, we observe that the TCP equation is, in fact, reasonably accurate compared to the actual difference between TFRC and TCP. This can be seen from that the difference ratio of $f(p_{\mathrm{tfrc}})$ and the TFRC throughput is much smaller than that of $f(p_{\mathrm{tcp}})$ and the TFRC throughput. Although there exists some significant gap caused by the convexity of the equation and the compounding effect of piecewise loss interval calculations (i.e., $\mathbf{E}[1/\hat{\theta}]$ versus $1/\mathbf{E}[\theta]$), there exists a bigger gap between $f(p_{\mathrm{tfrc}})$ and $f(p_{\mathrm{tcp}})$, which is directly attributable to disparate loss event rates observed by TFRC and TCP. Fig. 11 plots p_{tfrc} and p_{tcp} over the loss event rate observed by TFRC in the experiment for Fig. 8. Clearly, the variance does not affect the average loss events. More important, the ratio increases as the loss event rate increases. These results confirm our conjecture that the relatively small sending rate difference delineated by the bounds among $\overline{B}_{\mathrm{TFRC}}$, $f(\mathbf{E}[1/\hat{\theta}])$ and $f(1/\mathbf{E}[\theta])$ $(= f(p_{\mathrm{tfrc}}))$ gets amplified by the loss event rate difference.

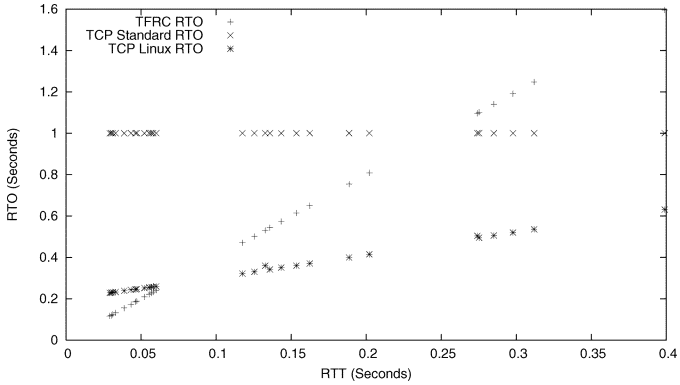

Fig. 12. RTO values of TCP (standard), Linux TCP, and TFRC connections to 28 different PlanetLab sites.

VI. Impact of RTO Estimation

TFRC RFC [10] recommends a different RTO estimation technique from TCP's. TFRC sets RTO to be four times a weighted moving average R of RTT samples. TCP [20] sets its RTO to be $\max\{1\text{ s}, R + 4D\}$ where D is a weighted moving average of the variance in RTT samples. This difference could potentially cause TFRC's RTO value to be much smaller than TCP's under a short delay network, and much larger under a long delay network. For instance, under a 10-ms RTT path, TFRC can have 40-ms RTO while a standard conformant TCP has 1-s RTO. Under a 500-ms RTT path, TFRC can have 2-s RTO while TCP having about 1 s. We acknowledge that many commercial implementations of TCP often use a smaller minimum RTO value than 1 s. However, no matter how this minimum value is set, since TFRC sets its RTO differently from TCP, its RTO value can potentially significantly differ from TCP. Fig. 12 shows the RTO values obtained from TFRC and TCP sessions from our site to 28 different sites in the PlanetLab [21]. We use the Linux TCP stack for the measurement (the values are taken directly from the kernel). Since the Linux TCP stack uses a much smaller minimum RTO value (200 ms) than 1 s, its RTO value tends to follow the actual RTT values when the delays become larger than 200 ms. TFRC RTO is much larger than Linux's RTO over long delay sites. We also plot the values for TCP standard (1 s in our measurement); the TFRC values are much smaller over low delay sites and larger over long delay sites.

Fig. 13 compares $f(p)$ with different RTT and RTO values. For this, we rewrite the function in (1) to be of a form $F(p, t_{\mathrm{RTT}}, t_0)$.[2] We examine a case for a 10-ms RTT network by comparing $F(p, 0.01, 0.04)$ and $F(p, 0.0.1, 1)$, and examine a case for a 500-ms RTT network by comparing $F(p, 0.5, 2)$ and $F(p, 0.5, 1)$. Note that in both cases, TCP sets its RTO close to 1 s. Fig. 13 shows their corresponding difference ratios. Under low loss rates, differing RTO values do not affect sending rates very much. But as the loss rate increases, the difference ratio increases. For a 10-ms network, $F(p, 0.01, 0.04)$ can give a feedback of around 18 times larger throughput estimate than a regular TCP $F(p, 0.0.1, 1)$ under 15% loss. In a 500-ms network, $F(p, 0.5, 2)$ can give a feedback of around a two times smaller throughput estimate than a regular TCP $F(p, 0.5, 1)$.

[2]We are overloading F; it is also for a different purpose in Section IV.

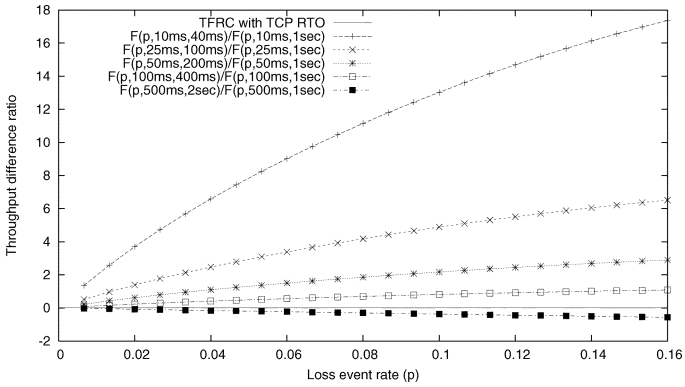

Fig. 13. Graph compares the value of the equation $f(p)$ when they have different RTO values. $f(p)$ is rewritten to be $F(p, t_{\mathrm{RTT}}, t_0)$ where t_{RTT} is the round-trip time and t_0 is the RTO. For instance, $F(p, 0.01, 0.04)$ is the TFRC equation with RTT 10 ms and RTO 40 ms. The functions with RTO 1 s are those following the TCP standard and the others follow the TFRC RFC.

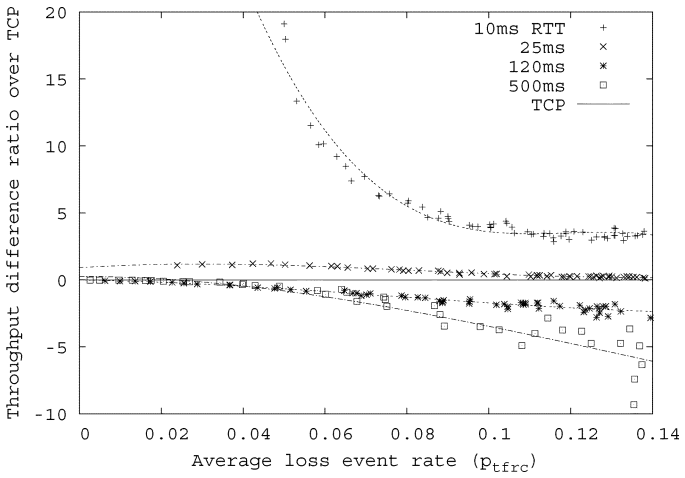

Fig. 14. Throughput difference ratio of TFRC and TCP under various delay networks. In these experiments, TFRC sets its RTO in the same way as recommended by the TFRC RFC. The curve lines are from least line fitting.

However, the actual sending rates of TFRC also depend on the other factors identified in the earlier sections. We study how RTO values can influence the actual TFRC throughput by simulation, below.

Fig. 14 shows the throughput difference ratios of TFRC and TCP from simulation under various delay networks from 10 to 500 ms. In this simulation, we use the same setup as discussed in Section III, but vary the network delays. In addition, we follow the recommendation of the TFRC and TCP RFCs in setting their corresponding RTO values. In general, Fig. 14 shows a completely different pattern from Fig. 13. Clearly visible from the figure is that in the 10- and 500-ms networks, the absolute throughput difference ratios are much higher than those predicted in Fig. 13, especially under a low loss event rate region for the 10-ms network, and under a high loss event rate region for the 500-ms networks. In contrast, simulation runs over the other delay networks tend to show less throughput difference ratios than the predicted. In addition, the throughput difference ratios from all runs tend to reduce (i.e., the TCP throughput gets larger than the TFRC throughput) as the loss event rate increases.

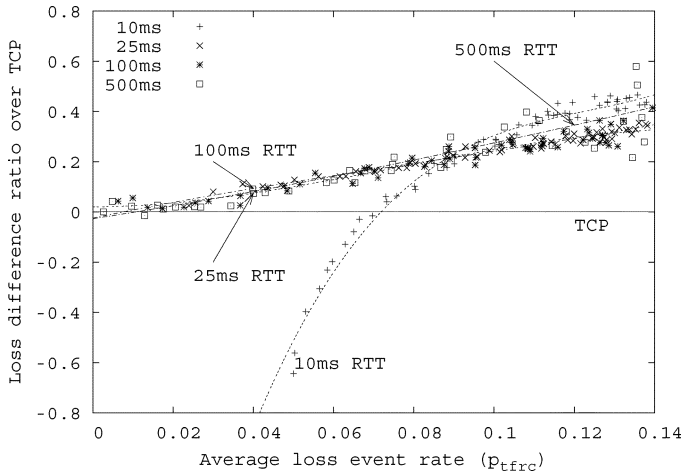

Fig. 15. Loss event rate difference ratio of TFRC and TCP in the same experiments as in Fig. 14.

A plausible explanation for the above phenomena can be made using R1, R4, and R5. In a low-delay network such as the 10-ms network, the feedback sending rate of TFRC [calculated using (1) which takes smaller RTO values than TCP] makes TFRC send at a higher sending rate than TCP. When the sending rate difference ratio of TFRC and TCP becomes higher to satisfy the condition for R5 in Theorem 4.1, R5 triggers TFRC to see a lower loss event rate than TCP. This further widens the sending rate difference ratio. For instance, around 5% loss event rate, the feedback sending rate difference ratio is around 8.5 (in Fig. 13). This has translated into more than 20 in the actual throughput difference ratio (in Fig. 14) while the loss event rate difference ratio of TFRC and TCP reaches -0.6. The other delay networks does not make the sending rate difference "sufficiently" high to trigger R5, which explains why they do not show as high throughput difference ratios as the 10-ms network. The throughput difference ratio does not grow indefinitely because the effect of R1 forces a slow responsive flow, such as TFRC, to experience more loss events than TCP. Furthermore, the effect of R1 increases as the loss event rate increases (as shown in Fig. 4). This is the reason why, as the loss event rate increases, the loss event rate difference ratio of TFRC and TCP increases in Fig. 15 and accordingly, the throughput difference ratio of TFRC and TCP reduces in Fig. 14. The 500-ms network shows a compounding effect of R1 and R4 under high loss event rates. In such a high delay network, the RTO values of TFRC are larger than those of TCP, and they make the feedback sending rate of TFRC to be smaller than the actual TCP sending rate, especially under medium to high loss event rates [where RTO has an increasingly higher impact according to (1)]. Accordingly, the TFRC source sends at a lower sending rate than TCP. When the sending rate is sufficiently lower than TCP's, R4 triggers TFRC to see a lower loss event rate than TCP, which happens under high loss event rate regions in Fig. 15. As the loss event rate increases, the effect of R4 is compounded with that of R1 to increase the loss event rate difference ratio (up to 0.5) and moves the throughput difference ratio of TFRC and TCP (up to -9.8) further into the negative region. These results confirm that the loss event rate difference (caused by some combinations of R1, R4, and R5)

can greatly amplify the initial sending rate difference caused by factors including different RTO values.

As we pointed out in the introduction, the issue with RTO estimation is not fundamental, but an artifact of policy. Therefore, it may be easily fixed by adopting a different policy. Our work indicates that any policy decision that changes the sending rate of TFRC to deviate from that of TCP must be done with a great care because sending rate difference can be greatly amplified by loss event rate difference.

VII. Heuristics to Mitigate Throughput Difference

In an effort to close the loop, we explore some heuristics in this section to correct the throughput difference between TFRC and TCP. We feel that the subject requires much more study, and one section of a paper may not be enough to cover the topic. Thus, our goal is very modest; we intend to demonstrate that fixing the problems may not be far out of reach and some very simple heuristics can be effective at least within the domain of the conditions we have tested.

Based on our simulation and analysis results shown in the previous sections, we follow several guidelines described below in designing heuristics methods.

1) TFRC should estimate RTO by using a method similar to (if not exactly the same as) the one recommended by the TCP RFC, due to the following reasons.
   - Using $\mathrm{RTO}_{\mathrm{tcp}}$ can greatly reduce the throughput difference, where $\mathrm{RTO}_{\mathrm{tcp}}$ denotes the value of RTO calculated based on the recommendation from TCP RFC. As we can see that the throughput difference triggered by improper RTO estimation as shown in Fig. 14 is significantly larger than that caused by loss event rate estimation and TCP-friendly equation alone as shown in Fig. 10.
   - It is hard to design a heuristics method with $\mathrm{RTO}_{\mathrm{tfrc}}$, where $\mathrm{RTO}_{\mathrm{tfrc}}$ denote the value of RTO calculated based on the recommendation from TFRC RFC. This is because Fig. 14 shows that TFRC with $\mathrm{RTO}_{\mathrm{tfrc}}$ may achieve higher or lower throughput than TCP depending on both RTT and loss event rates.
2) TFRC with $\mathrm{RTO}_{\mathrm{tcp}}$ always achieves less throughput than TCP as shown in Fig. 10, and the throughput difference increases as the loss event rate increases. Therefore, TFRC should be more aggressive and grab more bandwidth under high loss rates, while maintaining approximately the same throughput under low loss rates.
3) When TFRC uses $\mathrm{RTO}_{\mathrm{tcp}}$, the throughput difference decreases as RTT increases (the simulation result is not shown in the paper, but this observation can be confirmed by Fig. 18 that will be discussed later). Therefore, TFRC should grab more bandwidth with short RTTs, while maintaining approximately the same throughput with long RTTs.

We explore two simple heuristics. The first one is to multiply a constant factor c to the output of $f(p)$ as defined in (1) (i.e., the feedback sending rate), and set timeout period t_0 to $\mathrm{RTO}_{\mathrm{tcp}}$. This approach follows the first design guideline by setting t_0 to $\mathrm{RTO}_{\mathrm{tcp}}$. When TFRC uses $\mathrm{RTO}_{\mathrm{tcp}}$, it experiences a significant throughput drop over high loss event rates (as shown in Fig. 8). Thus, by simply scaling the feedback throughput, we may correct the throughput drop under high loss conditions, and this is consistent with our second design guideline. However, this method suffers from the effect that it also increases the throughput under low loss conditions, and hence may introduce some throughput difference under low loss conditions. Therefore, we need to be careful in choosing the scale factor since too large sending rate difference, especially, over low loss conditions might trigger loss event difference as discussed in Section IV.

The second approach is to use a new formula $f'(p)$ as defined in (17) to calculate the feedback sending rate, and set timeout period t_0 to $\mathrm{RTO}_{\mathrm{tcp}}$. Note that $f'(p)$ with $t_0 = \mathrm{RTO}_{\mathrm{tcp}}$ is equivalent to $f(p)$ with $t_0 = 1/c\mathrm{RTO}_{\mathrm{tcp}}$, thus we will refer to this approach as RTO scaling. This approach satisfies our first design guideline by using $\mathrm{RTO}_{\mathrm{tcp}}$

$$f'(p) = \frac{s}{t_{\mathrm{RTT}}\sqrt{\frac{2bp}{3}} + \frac{1}{c}t_0 3\sqrt{\frac{3bp}{8}}p(1+32p^2)}. \tag{17}$$

To investigate the impact of scaling factor c in (17) on TFRC throughput, we calculate the throughput ratio of (17) and (1) with $t_0 = \mathrm{RTO}_{\mathrm{tcp}}$ in both equations

$$\begin{aligned}\frac{f'(p)}{f(p)} &= \frac{t_{\mathrm{RTT}}\sqrt{\frac{2bp}{3}} + \mathrm{RTO}_{\mathrm{tcp}} 3\sqrt{\frac{3bp}{8}}p(1+32p^2)}{t_{\mathrm{RTT}}\sqrt{\frac{2bp}{3}} + \frac{1}{c}\mathrm{RTO}_{\mathrm{tcp}} 3\sqrt{\frac{3bp}{8}}p(1+32p^2)} \\ &= \frac{t_{\mathrm{RTT}} + \mathrm{RTO}_{\mathrm{tcp}}\frac{9}{4}p(1+32p^2)}{t_{\mathrm{RTT}} + \frac{1}{c}\mathrm{RTO}_{\mathrm{tcp}}\frac{9}{4}p(1+32p^2)}.\end{aligned} \tag{18}$$

We observe that when p is large, the ratio is greater than 1 if $c > 1$. That is, the scale factor can increase TFRC throughput under high loss rates. On the other hand, when p is small and close to zero, the ratio is approximately $t_{\mathrm{RTT}}/t_{\mathrm{RTT}} = 1$. That is, the scale factor slightly changes TFRC throughout under low loss rates. Therefore, the second approach follows our second design guideline. This can be explained intuitively. RTO affects the feedback throughput increasingly more as the loss event rate increases as shown in Fig. 13, because under high loss conditions, TCP is more likely to have timeouts.

We also observe that the throughput ratio decreases as t_{RTT} increases and both $\mathrm{RTO}_{\mathrm{tcp}}$ and p remains same. That is, the scaling factor makes TFRC grab more bandwidth only with short RTTs, and this satisfies our third design guideline.

Using the same simulation setup discussed in Section III, we evaluate various scaling factors for the first and second approaches. Fig. 16 shows the throughput difference ratios of TFRC and TCP for various scale factors in the first approach. We observe that with scale factor 1.5, TFRC has throughput much closer to TCP than with the other scaling factors. The results from the second approach, shown in Fig. 17, promise much better performance. We compare the results from the second approach to the 1.5 scaling factor result of Fig. 16. As the first approach applies the same scale factor to the feedback throughput irrespective of the loss event rate, it does not correct the pattern of the original TFRC whose sending rate drops under high loss rates (although it reduces the difference). On the other hand, the second approach has an effect of virtually increasing the scale factor to the feedback sending rate as the loss event rate increases. Fig. 17 shows that as the RTO scale factor increases, the fitted lines become more flat. The RTO

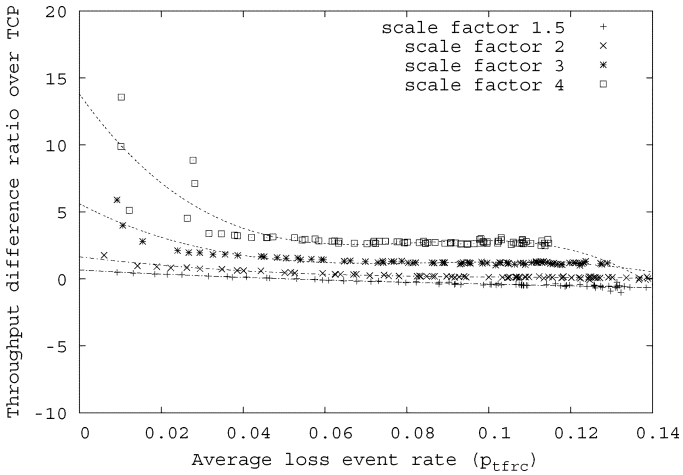

Fig. 16. Throughput difference ratio of TFRC and TCP for a different scaling factor. We apply a scale factor to the output of (1) to increase the throughput of TFRC. Scale factor 1.5 gives the best performance.

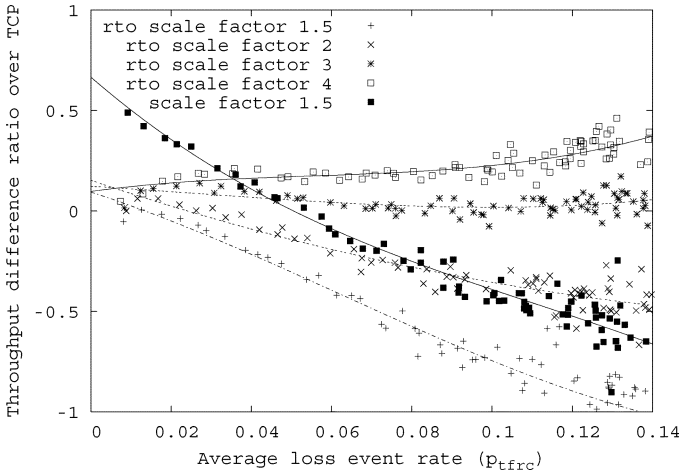

Fig. 17. Throughput difference ratio of TFRC and TCP for a different RTO scaling factor. We fix the value of RTO in TFRC to $\frac{1}{c}\mathrm{RTO}_{\mathrm{tcp}}$ where c is the RTO scale factor.

scale factor of 3 gives the best performance among the factors we tried. We further examine the effect of different network delays on the throughput difference when we apply various RTO scale factors. Fig. 18 shows the average values of *absolute* throughput difference ratios of TFRC and TCP. In the figure, one data point indicates the average value of the absolute throughput difference ratios of TFRC and TCP we obtained from all runs of different network loads with a fixed (physical) network delay (actual delays vary depending on the network load). An error bar marks one standard deviation away from an average value. The result indicates that RTO scale factor 3 gives the best performance even over various delay networks. We also see that the throughput ratios decrease as RTT increases.

VIII. Related Work

TFRC is based on pioneering work by Padhye *et al.* [18] that models the throughput of TCP using a loss event rate p, RTT and RTO. There are relatively a small number of studies that examine the behavior of TFRC [3], [6], [9], [22], [24], [27]. Most of them are based on simulation except [24]. The original TFRC

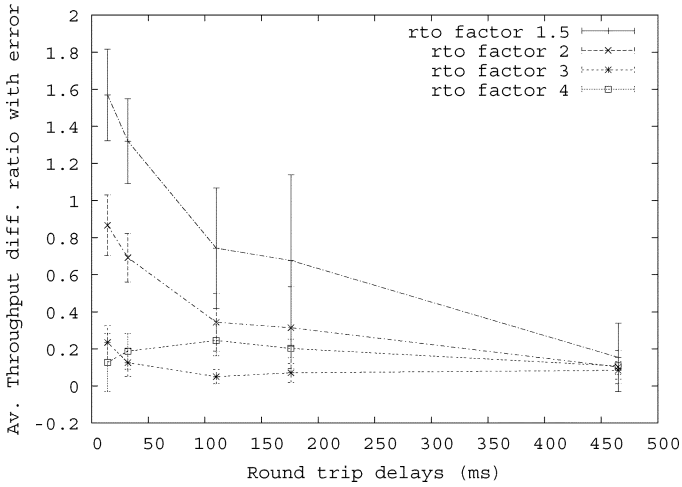

Fig. 18. Average of absolute difference ratios of TFRC and TCP over a different network delay. The error bars represent one standard deviation.

work [9] has extensively examined the behavior of TFRC in various Internet environments, mostly from an empirical perspective. Rhee *et al.* [22] show by simulation that the throughput of TFRC can be much lower than TCP under high loss environments and also when the feedback delay is very long. Bansal *et al.* [3] and Yang *et al.* [27] showed, by simulation, that the slow responsiveness of TFRC to transient congestion can make TFRC experience a higher loss rate, thus causing it to have a lower throughput than TCP. The loss rate difference between a constant rate flow (such as acknowledgments or ping) and TCP was also observed by Paxson [19]. The work [24] by Vojnović and Le Boudec is the only one we find in the literature that studies theoretical reasons why TFRC may not give the same throughput as TCP. Since we discuss the work extensively in the introduction, we do not describe it further in this section. Recently, Chen and Nahrstedt [6] showed by simulation the cases where TFRC may use less bandwidth than TCP in mobile ad hoc networks. Their work confirms the conservativeness of TFRC shown in [24], in MANET environments.

IX. Conclusion

In this paper, we examined how the three main factors that determine the TFRC throughput, namely the TFRC throughput equation, loss event rate estimation and RTO estimation, can influence the long-term throughput imbalance between TFRC and TCP. We give theoretical reasons why such imbalance occurs. The main findings are that: 1) any two competing flows sharing the same bottleneck link will see different loss event rates if they have significantly different sending rates, and 2) the loss event rate difference can greatly amplify the initial sending rate difference. Early work explains reasons only for the conservativeness of TFRC, but does not account for the reason why TFRC can have higher throughput than TCP. Our findings analytically explain reasons for that. We also provide a couple of more reasons why the sending rate of TFRC can be different from TCP initially to provide a trigger for different loss event rates to occur. These are namely the convexity of $1/x$ and different RTO estimation. While the other factors are fundamental, the issue with RTO estimation is an artifact of policy, and has much of relevance in practice.

Our work on TFRC can provide hints to engineers about possible network situations where TFRC might show ill-behaviors and guide them in performing "stress tests." In addition, as the authors of [24] point out, studying the causes of the throughput discrepancy helps engineers in designing new protocols that have similar goals as TFRC. For instance, new emerging networks, such as mobile ad hoc networks and high-speed long distance networks, whose characteristics are substantially different from the traditional Internet, present environments where TCP may not work so well [8], [16]. For such networks, a new congestion control technique is needed. We believe that our work can be useful for developing such a protocol as many new TCP-variant protocols are being proposed (e.g., [8]).

In this paper, we assume fixed RTTs and fixed packet sizes for all flows, which are not realistic assumptions. Developing analysis where these assumptions are relaxed is of future interest. We also assume that all packets in the same end-to-end network path are subject to the same loss probability. Thus, our results are valid for any AQM scheme that supports this assumption. Most AQM schemes including RED [13], PD-RED [23], REM [2], BLUE [7], and GREEN [26] have this property.

ACKNOWLEDGMENT

The authors would like to thank M. Vojnović for his generous help and comments, and M. Zukerman and the anonymous reviewers for their valuable feedback.

REFERENCES

[1] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," RFC 2581, 1999.
[2] S. Athuraliya, V. Li, S. Low, and Q. Yin, "REM: Active queue management," *IEEE Netw.*, vol. 15, no. 3, pp. 48–53, May 2001.
[3] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," in *Proc. ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001.
[4] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Measurement and Modeling of Computer Systems*, 1998, pp. 151–160.
[5] T. Bonald, M. May, and J. C. Bolot, "Analytic evaluation of RED performance," in *Proc. INFOCOM*, 2000, pp. 1415–1424.
[6] K. Chen and K. Nahrstedt, "Limitations of equatiProc.based congestion control in mobile ad hoc networks," in *Proc. Int. Workshop on Wireless Ad Hoc Networking (WWAN 2004) in Conjunction with ICDCS-2004*, Mar. 2004.
[7] W. Feng, K. Shin, D. Kandlur, and D. Saha, "BLUE active queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 513–528, Aug. 2002.
[8] S. Floyd, "High speed TCP for large congestion windows," RFC 3649, 2003.
[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000, pp. 43–56.
[10] S. Floyd, M. Handley, J. Padhye, and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol specification , 2003.
[11] S. Floyd and T. Henderson, "The New Reno modification to TCP's fast recovery algorithm," RFC 2582, 1999.
[12] S. Floyd and V. Jacobson, "Traffic phase effects in packet-switched gateways," *Internetw: Res. Exp.*, vol. 3, no. 3, pp. 115–156, Sep. 1992.
[13] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
[14] M. Goyal, R. Guerin, and R. Rajan, "Predicting TCP throughput from non-invasive network sampling," in *Proc. IEEE INFOCOM*, Jun. 2002.
[15] J. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," in *Proc. ACM SIGCOMM*, Aug. 1996.
[16] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proc. IEEE/ACM MOBICOM '99*, Aug. 1999, pp. 219–230.
[17] E. Kohler, M. Handley, S. Floyd, and J. Padhye, Datagram Congestion Control Protocol (DCCP) [Online]. Available: draft-ietf-dccp-spec-05.txt
[18] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM '98*, 1998, pp. 303–314.
[19] V. Paxson, "End-to-end internet packet dynamics," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 277–292, Jun. 1999.
[20] V. Paxson and M. Allman, "Computing TCP's retransmission timer," RFC 2988, 2000.
[21] PlanetLab [Online]. Available: http://www.planet-lab.org/
[22] I. Rhee, V. Ozdemir, and Y. Yung, TEAR: TCP emulation at receivers—Flow control for multimedia streaming Dept. Comput. Sci., North Carolina State Univ., Chapel Hill, NC, Tech. Rep., 2000.
[23] J. Sun, K. Ko, G. Chen, S. Chan, and M. Zukerman, "PD-RED: To improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, pp. 406–408, Aug. 2003.
[24] M. Vojnović and J. Le Boudec, "On the long run behavior of equation-based rate control," in *Proc. ACM SIGCOMM 2002*, 2002, pp. 103–116.
[25] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in *Proc. ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001.
[26] B. Wydrowski and M. Zukerman, "GREEN: An active queue management algorithm for a self managed internet," in *Proc. ICC*, May 2002, pp. 2368–2372.
[27] R. Yang, M. Kim, and S. Lam, "Transient behaviors of TCP-friendly congestion control protocols," in *Proc. INFOCOM*, Mar. 2001.
[28] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of internet path properties," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

Injong Rhee received the Ph.D. degree from the University of North Carolina, Chapel Hill.

He is currently an Associate Professor of computer science at North Carolina State University (NCSU), Raleigh. In 2000, he founded Togabi Technologies, Inc., a company that develops and markets mobile wireless multimedia applications for next-generation wireless networks, and he was CTO and CEO of the company until 2002 when he came back to NCSU. His research interests are computer networks, congestion control, multimedia networking, distributed systems, and operation systems.

Dr. Rhee received the NSF Early Faculty Career Development Award in 1999.

Lisong Xu received the B.E. and M.E. degrees from the University of Science and Technology, Beijing, China, and the Ph.D. degree from North Carolina State University, Raleigh, in 1994, 1997, and 2002, respectively, all in computer science.

From 2002 to 2004, he was a Postdoctoral Research Fellow at North Carolina State University. He is currently an Assistant Professor in computer science and engineering at the University of Nebraska-Lincoln. His research interests include computer networks and distributed systems.