

Issues in Model-Based Flow Control

Sridhar Ramesh and Injong Rhee
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534

Abstract

This paper examines potential fairness problems associated with a model based approach to TCP-friendly flow control for non-TCP traffic. In specific, such an approach involves using a TCP-friendly formula that estimates the throughput of a TCP session with the same end-to-end traffic characteristics as the non-TCP connection under consideration. The inputs to this formula include the round trip time, the timeout value, and the packet loss fraction of the connection. This paper shows that estimating the loss fraction per transmitted packet highly depends on the current transmission rate of the connection as well as the actual loss fraction of the path. Thus the estimated loss fraction can contain errors which result in inaccurate estimation of the corresponding TCP throughput. This inaccuracy can push the transmission rate of the non-TCP connection away from the fair share of the bottleneck bandwidth on the end-to-end path, so that under steady state, the connection ends up receiving either over-allocation or under-allocation of bandwidth.

1 Introduction

Congestion control is an integral part of any best-effort Internet data transport protocol. It is widely accepted that the congestion avoidance mechanisms employed in TCP [1] have been one of the key contributors to the success of the Internet. A conforming TCP flow is expected to respond to congestion indication (e.g., packet loss) by drastically reducing its transmission rate and by slowly increasing its rate during steady state. This congestion control mechanism encourages the fair sharing of a congested link among multiple competing TCP flows. A data flow is said to be *TCP-friendly* if at steady state, it uses no more bandwidth than a conforming TCP connection running under comparable conditions.

Recently we have seen several efforts to develop a stochastic model of TCP congestion control that gives a simple analytical formula for the throughput of a TCP sender as a function of packet loss and round trip time (RTT) [2, 7, 3]. These efforts are propelled by the interests in using the formula for TCP-friendly flow control of a non-TCP flow such as UDP traffic [6, 8] and reliable multicast [5]. Typically, these flow control schemes work as follows. A receiver monitors packet loss rates and round trip delays, and using a TCP friendly formula, it estimates the throughput of a TCP connection running under the same operating conditions. The estimated throughput is sent as feedback to the sender. If the feedback throughput is less than or equal to the current transmission rate of the non-TCP flow, then the sender sets its rate to the feedback throughput. Otherwise, it increases its rate. It is critical to the correct functioning of this algorithm that:

1. The packet loss rates and round trip delays estimated by the receiver approximate, with a good degree of accuracy, those seen by a conforming TCP connection running under comparable conditions — like, for instance, a TCP connection sharing the same end-to-end path.
2. The throughput formula used to estimate the TCP throughput as a function of the packet loss rate and round trip delay must provide an accurate result.

In this paper, we show that there are situations where either 1, or 2, or both of the above conditions may not be satisfied. Further, we show that this could not only result in erroneous estimation of feedback throughput, but also in either overallocation or underallocation of bandwidth to the non-TCP flow in steady-state.

In Section 2, we briefly outline the TCP-friendly flow control algorithm proposed in [5], and the formula used in TCP throughput estimation. In Section 3, we describe the sources of error in TCP throughput estimation, and show how these errors result in unfair allocation of bandwidth by the flow control algorithm. Section 4 contains numerical examples which substantiate these findings.

2 Model-based Flow Control

As discussed in Section 1, model-based flow control involves a formula to estimate TCP throughput based on packet loss rates. A stochastic model developed by Padhye et al. [3] makes the TCP throughput estimation based on the following assumptions:

- When a packet is lost, all subsequent packets in the same RTT round are lost.
- The probability that a packet is lost in an RTT round, given that no previous packet in the same round is lost is independent of packet loss in earlier rounds. This probability — call it l_{act} — is defined as the actual packet loss fraction.

TCP throughput is then given by the following equation¹:

$$\mathcal{B}_{Pad}(l_{act}) = \frac{s}{t_{RTT}\sqrt{\frac{2bl_{act}}{3}} + t_0 \min\left(1, 3\sqrt{\frac{3bl_{act}}{8}}\right) l_{act}(1 + 32l_{act}^2)} \quad (2.1)$$

where l_{act} is the loss fraction of the TCP packets, t_{RTT} is the round trip delay of an end-to-end path where TCP runs, and t_0 is the TCP timeout value. This is an improvement over an earlier TCP throughput estimation model proposed by Floyd [2] and Ott et al.[7]. Their model gives the following formula:

$$\mathcal{B}_{Flo}(l_{act}) = \frac{1.22s}{t_{RTT}\sqrt{l_{act}}} \quad (2.2)$$

In general, a TCP throughput formula may be represented by some function $\mathcal{B}(l_{act})$ of the packet loss fraction l_{act} .

A typical model-based flow control scheme operates as follows. First, the non-TCP flow estimates the value of l_{act} , t_{RTT} , and t_0 . Let l_{est} be the estimate of l_{act} . Then, using a TCP throughput formula such as those in (2.1) and (2.2), the receiver of the non-TCP flow computes the value $\mathcal{B}(l_{est})$ which is fed back to its sender as the estimated steady state throughput of a TCP connection operating under comparable conditions. If $\mathcal{B}(l_{est}) \leq \mu$, the sender sets its transmission rate to $\mathcal{B}(l_{est})$. Otherwise, it increases its rate by some amount. The receiver continues to report a new value of

¹This is only an approximate formula. The exact formula can be found in [3]

$\mathcal{B}(l_{est})$, and the sender makes the adjustment of its rate accordingly. The idea is that if the feedback throughput value is accurate enough, the transmission rate eventually converges to the fair share of bandwidth (i.e., $\frac{B}{n+1}$).

To estimate l_{act} of a TCP flow under observation, Padhye et al. [3] count the number of TCP loss indications (triple duplicate acknowledgements, and timeouts) over a certain period, and divide the result by the total number of TCP packets transmitted over that duration. This is an approximation. Let l_{app} be the resulting value. l_{app} is used as input to their formula to estimate the throughput of a TCP connection under observation.

However, when a non-TCP flow is regulated using (2.1), it is not possible to compute l_{app} since the non-TCP flow does not have the same congestion window sizes as TCP. Instead, Handley et al. [5] estimate l_{act} on the non-TCP flow by dividing the number of *loss events* by the total number of packets transmitted. Loss events are registered 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 a loss event, followed by another RTT back-off, and so forth. Let the resulting value be l_{est} , as defined earlier.

Let μ be the bit rate of transmission of packets over the non-TCP connection. Let B be the total bandwidth available between the source and a given destination. If there are n active TCP sessions between the source and the destination, the throughput of each, in steady state, is given by $\frac{B-\mu}{n}$. If B is the total bandwidth shared among n TCP sessions and one non-TCP flow running on the same end to end path, the objective of TCP-friendly algorithms is to ensure that the rate μ of the non-TCP flow is maintained at the fair share given by $B_{fair} = \frac{B}{n+1}$, in steady-state. The known TCP-friendly algorithms achieve this by a feedback mechanism. Specifically, an estimate of the throughput on each of the n TCP sessions is fed back to the sender which then uses this to regulate its transmitting rate μ . Assuming that the TCP sessions share the residual bandwidth equally, the steady-state throughput of each TCP session is $\frac{B-\mu}{n}$. Ideally, this is the parameter which must be fed back to the sender. It can be seen that when $\mu > B_{fair}$, $\frac{B-\mu}{n} < B_{fair}$, and when $\mu < B_{fair}$, $\frac{B-\mu}{n} > B_{fair}$.

However, as discussed earlier, the TCP throughput estimate is calculated based on the loss fraction of packets on the non-TCP flow. This obviously means that errors in estimating the loss fraction could lead to errors in estimating TCP throughput. In the following sections, we show that this indeed happens, and could even result in unfair allocation of bandwidth. In particular, we show that:

1. When $\mu > \frac{1}{RTT}$ (i.e., one packet per RTT), $l_{est} < l_{act}$. Therefore, substituting l_{est} for l_{act} in any formula which estimates TCP throughput based on l_{act} results in over-estimation of the TCP throughput.
2. When $\mu < B_{fair}$, $l_{est} > l_{app}$. Thus, substituting l_{est} for l_{app} in a formula estimating TCP throughput based on l_{app} gives us an erroneous estimate. This results in over-estimation of the throughput.
3. When $\mu > B_{fair}$, we could have $l_{est} < l_{app}$ under certain conditions. Thus, substituting l_{est} for l_{app} in a formula estimating TCP throughput based on l_{app} gives us an erroneous estimate and under-estimation of TCP throughput.
4. The formula used to estimate TCP throughput, such as the one provided in [3], may itself have some inaccuracy.
5. Depending on the starting rate of μ , these errors introduced in TCP throughput estimation push μ further away from the fair share, rather than forcing it to converge to the fair share. Thus, at steady state, the non-TCP flow may end up receiving either over-allocation or under-allocation of bandwidth.

3 Errors in TCP Throughput Estimation and their Impact on Resource Allocation

In this section, we look at the different sources that cause errors in estimating the equivalent TCP throughput and show that their potential consequence is unfair allocation of resources to the non-TCP flow.

First, we compare the packets sent over the non-TCP stream over the measurement period T , to that sent over a TCP session. Let T be equal to N round trip times.

Let W_i , $i = 1, 2, \dots, N$, be the number of packets sent in the i^{th} round over the TCP connection under observation. Therefore, the number of packets sent over the TCP session is given by $\sum_{i=1}^N W_i$. The probability of having no packet losses in a window of size W_i is $(1 - l_{act})^{W_i}$. The probability of having a loss event in a window of size W_i is therefore $1 - (1 - l_{act})^{W_i}$. The expected number of loss events in N round trip times is given by :

$$Loss_{TCP} = \sum_{i=1}^N 1 - (1 - l_{act})^{W_i}$$

By definition, l_{app} is the number of loss events divided by the number of transmitted packets. This is, strictly speaking, a random variable, but if measured over a sufficiently large interval, it may be approximated by its expected value. Therefore:

$$l_{app} \approx \frac{\sum_{i=1}^N 1 - (1 - l_{act})^{W_i}}{\sum_{i=1}^N W_i} \quad (3.3)$$

In the same duration, μT packets are transmitted over the non-TCP session (we assume μT is an integer). The expected number of loss events is given by:

$$Loss_{non-TCP} = \sum_{i=1}^N 1 - (1 - l_{act})^{\frac{T\mu}{N}}$$

l_{est} is given by the number of loss events on the non-TCP flow divided by the total number of packets transmitted. If measured over a sufficiently long interval, it is approximately equal to its expected value. Thus, we have:

$$l_{est} \approx \frac{\sum_{i=1}^N 1 - (1 - l_{act})^{\frac{T\mu}{N}}}{T\mu} = N \left(\frac{1 - (1 - l_{act})^{\frac{T\mu}{N}}}{T\mu} \right) \quad (3.4)$$

Since $\mu < \frac{B-\mu}{n}$, and TCP is in steady state (congestion avoidance mode), there are more packets transmitted on any TCP connection, than on the non-TCP connection. Hence, $\sum_{i=1}^N W_i > T\mu$.

3.1 Error in Loss Fraction Estimation

Let l_{act} be the actual loss fraction, and let l_{est} be the estimate measured on the non-TCP connection. Let l_{app} be the estimate measured on one of the TCP connections. Then, we have:

THEOREM 1

$$l_{est} > l_{app} \text{ if } \mu < \frac{B - \mu}{n}$$

PROOF : See Appendix A.

THEOREM 2

$$l_{est} < l_{app} \text{ if } \mu > \frac{B - \mu}{n}, \text{ and } \frac{T\mu}{N} > W_i, \text{ for } 1 \leq i \leq N$$

PROOF : See Appendix B.

THEOREM 3 *When $\mu > \frac{1}{RTT}$, $l_{est} < l_{act}$.*

PROOF : See Appendix C.

3.2 Impact of Errors in Loss Fraction on TCP Throughput Estimation

First, consider the case where TCP throughput is estimated as a function of l_{act} , such as (2.1). Irrespective of the actual model chosen, $\mathcal{B}(l)$ should necessarily be a monotonically decreasing function of l . Let $\beta(l) = -\frac{\partial \mathcal{B}}{\partial l} > 0$, for all l such that $0 < l < 1$. Thus, when $l_{est} < l_{act}$, the estimate $B_{feed} = \mathcal{B}(l_{est})$ is likely to be larger than the actual value $\mathcal{B}(l_{act}) = \frac{B - \mu}{n}$ which is the correct feedback parameter. In some cases, we may even have the following situation:

$$B_{fair} < \mu < B_{feed} = \mathcal{B}(l_{est})$$

Consider the case where the non-TCP transmission rate μ is given by $\frac{B}{n+1} + n\Delta B$, for some $\Delta B > 0$. Now, suppose we are given a function $\mathcal{B}(l)$ which correctly estimates the throughput on a TCP connection given the value of the loss fraction of

packets, l_{act} . For the value of μ considered, $\mathcal{B}(l_{act}) = \frac{B-\mu}{n}$, since we have assumed that $\mathcal{B}(l_{act})$ correctly estimates TCP throughput. Hence,

$$\mathcal{B}(l_{act}) = \frac{B}{n+1} - \Delta B$$

Thus, when the value of μ is greater than the *fair share*, $\frac{B}{n+1}$, the feedback parameter is less than the fair share. This facilitates reduction of the transmission rate on the non-TCP connection to preserve bandwidth fairness.

However, when the feedback parameter is calculated based on l_{est} , the estimate of l_{act} rather than the actual value l_{act} , there is no such guarantee. As shown earlier,

$$l_{est} = \frac{\sum_{i=1}^N \sum_{j=1}^{T\mu/N} (1 - l_{act})^{j-1} l_{act}}{T\mu}$$

where T is the period of observation and N is the number of round trip times in T . Therefore, $l_{est} = l_{act} - \Delta l < l_{act}$. The feedback parameter $\mathcal{B}(l_{est}) = \mathcal{B}(l_{act} - \Delta l) \approx \frac{B}{n+1} - \Delta B + \beta(l_{act})\Delta l$. If $\Delta B < \frac{\beta(l_{act})\Delta l}{n+1}$, then we have:

$$\mathcal{B}(l_{est}) \approx \frac{B}{n+1} - \Delta B + \beta(l_{act})\Delta l > \frac{B}{n+1} + n\Delta B = \mu$$

Next, we consider the case where TCP throughput is calculated using a formula based on l_{app} . Assuming we are provided a formula $\mathcal{B}(\cdot)$ that accurately estimates TCP throughput, we should have:

$$\mathcal{B}(l_{app}) = \frac{B - \mu}{n}$$

Let $\mu = \frac{B}{n+1} + n\Delta B$. But since we use l_{est} instead of l_{app} , the feedback parameter is given by:

$$\mathcal{B}(l_{est}) \approx \frac{B}{n+1} - \Delta B + \beta(l_{app})(l_{app} - l_{est})$$

Thus, the following problems could be encountered:

1. When $\mu < \frac{B-\mu}{n}$, $l_{est} > l_{app}$. There could be a situation where $\mathcal{B}(l_{est}) \leq \mu < B_{fair}$.
2. When $\mu > \frac{B-\mu}{n}$, and $l_{est} < l_{app}$, there could be a situation where $\mathcal{B}(l_{est}) \geq \mu > B_{fair}$.

3.3 Errors in TCP Throughput Model

In the earlier section, we considered the error due to inaccurate estimation of the loss fraction, assuming that the throughput model to be perfect. However, it is seen that the throughput estimation based on (2.1) also introduces some error (see [3]). From the tables provided in [3], it was observed that the relative error approached 100% in certain extreme cases. In this section, we show that this error can add on to the error in loss fraction estimation to result in erroneous feedback to the non-TCP sender.

Suppose the model estimates TCP's throughput to be $\mathcal{B}_{mod}(l_{act}) = B_{act} + B_\epsilon$, where B_{act} is the actual TCP throughput. Consider a link between two nodes with a minimum round trip time of T_1 seconds. Let $N_1 = \lceil B_{act} \times T_1 \rceil$. Let $l_1 = l_{act} - \Delta l = l_{act} \times \frac{1 - (1 - l_{act})^{N_1}}{N_1}$. It is easily seen that $\Delta l > 0$ if $N_1 > 1$. Let the bandwidth of the link under consideration be $B = nB_{act} + \frac{\mathcal{B}_{mod}(l_{act}) + \mathcal{B}_{mod}(l_{act} - \Delta l)}{2}$. Now, suppose we have a flow of rate $\mu = \frac{\mathcal{B}_{mod}(l_{act}) + \mathcal{B}_{mod}(l_{act} - \Delta l)}{2}$, sharing this link with n , $n \geq 1$, TCP connections. The throughput of each TCP session is $\frac{B - \mu}{n} = B_{act}$. Using (2.1) however, we get different results. If l_{act} is estimated correctly, the TCP bandwidth is estimated as:

$$\mathcal{B}_{mod}(l_{act}) = B_{act} + B_\epsilon$$

Since $\Delta l > 0$, and $\mathcal{B}_{mod}(l)$ is a decreasing function of l , $\mathcal{B}_{mod}(l_{act} - \Delta l) > \mathcal{B}_{mod}(l_{act})$. So,

$$\mu - \mathcal{B}_{mod}(l_{act}) = \frac{\mathcal{B}_{mod}(l_{act} - \Delta l) - \mathcal{B}_{mod}(l_{act})}{2} > 0$$

In other words, $\mu > B_{act} + B_\epsilon$.

But l_{act} is not correctly estimated. The estimate of l_{act} is given by:

$$l_{est} = \frac{1 - (1 - l_{act})^{\mu \times RTT}}{\mu \times RTT}$$

where RTT is the mean round-trip time. Obviously, $RTT \geq T_1$. Besides, $\mu > B_{act} + B_\epsilon$. Hence, from Lemma 1, it can be seen that $l_{est} < l_1 < l_{act}$. We can write $l_{est} = l_{act} - Dl$ where $Dl > \Delta l$.

The estimate of the TCP bandwidth, using (2.1), is given by:

$$\mathcal{B}_{mod}(l_{est}) = \mathcal{B}_{mod}(l_{act} - Dl)$$

As mentioned earlier, $\mathcal{B}_{mod}(l)$ is a decreasing function. Therefore,

$$\mathcal{B}_{mod}(l_{est}) = \mathcal{B}_{mod}(l_{act} - Dl) > \mathcal{B}_{mod}(l_{act} - \Delta l)$$

Also,

$$\mathcal{B}_{mod}(l_{est}) - \mu > \mathcal{B}_{mod}(l_{act} - \Delta l) - \mu = \frac{\mathcal{B}_{mod}(l_{act} - \Delta l) + \mathcal{B}_{mod}(l_{act})}{2} > 0$$

The fair share is given by:

$$B_{fair} = B_{act} + \frac{\mathcal{B}_{mod}(l_{act} - \Delta l) - \mathcal{B}_{mod}(l_{act})}{2n + 2} < B_{act} + B_{\epsilon} + \frac{\mathcal{B}_{mod}(l_{act} - \Delta l) - \mathcal{B}_{mod}(l_{act})}{2} = \mu$$

Thus, we have a situation where $\mathcal{B}_{mod}(l_{est}) > \mu > \mathcal{B}_{mod}(l_{act}) > B_{fair}$, i.e., one where the error in estimating l_{act} , coupled with the error introduced by the formula (2.1) results in erroneous throughput estimation. In the next section, we show how this may even result in unfair allocation of resources.

3.4 Non-Convergence to Fair-Share

In the earlier sections, we have shown that the errors in estimating the loss fraction and errors in the TCP throughput formula could result in the following situations:

$$(i) \ B_{feed} > \mu > B_{fair} \quad \text{OR} \quad (ii) \ B_{feed} < \mu < B_{fair}$$

where μ is the rate of the non-TCP flow, B_{fair} is the fair share, and $B_{feed} = \mathcal{B}(l_{est})$ is the feedback parameter.

In this section, we show that when $B_{feed} > (\text{resp } <) \mu > (\text{resp } <) B_{fair}$, the non-TCP rate never converges to the fair share in steady state. In addition, the long-term average throughput of the non-TCP flow does not converge to the fair share either.

We prove this for the case where $B_{feed} > \mu > B_{fair}$, and the proof for the case $B_{feed} < \mu < B_{fair}$ is similar. We first observe that $B_{feed} = \mathcal{B}(l_{est})$, where l_{est} is itself a function of μ . Therefore, we can write $B_{feed} = B^{\Omega}(\mu)$. Assuming $\mathcal{B}(l_{est})$ is a continuous function, and l_{est} is a continuous function of μ , $B^{\Omega}(\mu)$ is also continuous. We know that $B^{\Omega}(\mu) > \mu$. Hence, we have either:

$$B^{\Omega}(\nu) = \nu \quad \text{for some } \nu > \mu \tag{3.5}$$

Or:

$$B^\Omega(\nu) > \nu \text{ for all } \nu > \mu \tag{3.6}$$

If (3.5) is true, then the feedback parameter when the initial rate of the non-TCP flow is ν is given by $B^\Omega(\nu) = \nu$. Therefore, the rate as determined by the non-TCP flow control protocol converges to $\nu > \mu > B_{fair}$. The long-term average of the flow is also ν .

If (3.6) is true, when the initial rate ν_0 is greater than μ , it can be seen that the feedback parameter $B^\Omega(\nu_0)$ is greater than ν_0 . Writing $\nu_1 = B^\Omega(\nu_0)$, and in general, $\nu_{i+1} = B^\Omega(\nu_i)$ for $i > 0$, we have:

$$B^\Omega(\nu_i) > \nu_i \geq \nu_0$$

In practice, $\nu_{i+1} = \min(B^\Omega(\nu_i), B_{max})$, but since $\nu_i \leq B_{max}$, we can still write:

$$\nu_{i+1} \geq \nu_i \geq \nu_0$$

This means $\nu_i > \mu > B_{fair}$, for all $i \geq 0$. The long-term average of the flow is given by:

$$\gamma_\infty = \lim_{N \rightarrow \infty} \frac{\sum_{i=0}^{N-1} \nu_i}{N} > \lim_{N \rightarrow \infty} \frac{\sum_{i=0}^{N-1} \mu}{N} = \mu > B_{fair}$$

4 Numerical Examples

We consider three numerical examples of cases where the flow control algorithm using formula based feedback results in unfair allocation of resources.

In the first example, the formula is assumed to accurately compute the TCP throughput when the loss rate is correctly estimated, and the error in TCP throughput estimation is due to erroneous estimation of the loss fraction. There are two TCP sessions sharing the bottleneck link which has a bandwidth B of 500 KB/s. The fair share is therefore 166.7 KB/s. We assume that the non-TCP flow has an initial rate μ of 190 KB/s. An ns simulation was accordingly set-up, with two TCP connections sharing a 4 Mb/s link with a constant bit-rate source sending packets at 190 KB/s.

The observed loss fraction in the simulation, l_{act} , is 0.0156. However, the non-TCP loss fraction l_{est} , obtained by dividing the number of non-TCP loss events by the total number of non-TCP packets transmitted, is 0.0125. Through simulation, it was determined that the TCP loss fraction, l_{app} , is equal to 0.0125 when the non-TCP rate $\mu = 120$ KB/s. Therefore, any correct throughput formula estimates the TCP throughput based on l_{app} as $\frac{B-\mu}{n} = \frac{500-120}{2} = 190$ KB/s. In other words, the feedback parameter B_{feed} is equal to μ , and thus causes the non-TCP flow to be permanently set at $\mu = 190$ KB/s, though μ is significantly larger than the fair share. Since $\mu = 190$ KB/s, this leaves less than 155 KB/s for either of the TCP connections. Thus, the bandwidth allocation to the non-TCP flow is about 23% greater than that for each TCP connection.

In the second example, we illustrate the effects of estimating the TCP throughput based on the formula given in (2.1). There could be an error introduced by the formula, compounded by an error due to inaccurate estimation of the loss fraction. Among the measurements provided in [3], we consider the case where 100 serially initiated TCP connections were established for 100 second intervals between two hosts. An average throughput of 17.13 KB/s was observed per connection, with a loss fraction of 0.0078, mean RTT of 0.2501 seconds and time out period of 2.5127. The throughput estimation, as given by (2.1) was 33.4 KB/s. Now, suppose the bottleneck is a 496 KB/s (≈ 4 Mbps) link shared with 27 TCP sessions. The fair share is 17.75 KB/s, but when $\mu = 33.5$ KB/s, the TCP throughput is 17.13 KB/s. Assuming the TO and RTT parameters are the same as in the measurement described above, the approximate loss fraction l_{app} is .0078, and the formula in (2.1) estimates the TCP throughput as 33.4 KB/s. Assuming $l_{act} \approx l_{app}$, and applying the transformation for l_{est} given in (3.4), the estimate of the loss fraction on the non-TCP connection is .0076. Substituting this value in equation (2.1), $B_{feed} = \mathcal{B}(l_{est})$ is 34.01 KB/s which is higher than μ although μ itself is higher than the fair share. Thus, the resource allocation to μ is at least about 89% higher than the fair share.

The third example involves receivers at the end of a modem line wherein a strong correlation between the RTT and TCP window size may be observed. [3] also includes

experimental results with a receiver connected through a 28.8 Kb/s modem, and a significant discrepancy between measured TCP throughput and that obtained by the TCP formula is observed. For instance, in one of the experiments, the TCP connection transmitted 154 packets over a 100s interval for a throughput of 1.54 KB/s. The measured RTT is 4.726 seconds. The measured l_{app} is 0.064935. From Lemma 2, it can be shown that this corresponds to $l_{act} \geq 0.08415$. Now, consider a non-TCP flow with rate $\mu = 0.4$ KB/s sharing the same end-to-end path whose available bandwidth is 1.94 KB/s. Both the TCP and non-TCP flows have the same l_{act} , but since the non-TCP flow rate is less, its loss fraction estimate is given by $l_{est} \geq 0.8103$. Using this loss fraction estimate, the TCP throughput estimated by the formula is ≤ 0.346 KB/s. The fair share of bandwidth, on the other hand, is 0.97 KB/s. Thus, we have $B_{fair} > \mu > B_{fed}$. As seen earlier, this results in non-convergence of the non-TCP flow rate to the fair share. In this case, the non-TCP flow rate does not converge to any value greater than 0.4 KB/s. The TCP throughput is therefore at least 285% higher than the steady-state non-TCP rate.

5 Conclusions

Model-based TCP-friendly rate control of a non-TCP flow involves estimating the throughput of a TCP connection sharing the same end-to-end path, based on the estimated TCP packet loss fraction. In this paper, we have shown that the estimated TCP throughput may involve errors due to the following reasons:

- The estimate of the packet loss fraction on the non-TCP flow may involve errors.
- The formula used to estimate the TCP throughput may introduce some errors.

We have also shown that these errors could result in significant overallocation or underallocation of bandwidth to the non-TCP flow. Through the numerical examples in this paper, it may be seen that the overallocation or underallocation is of the order of 65% to 285%. However, we note that some of these numerical examples involved comparison between the expected throughput and the measured throughput based

on data traces. It was assumed that these traces are sufficiently long so that the time average throughput converged to the expected value. The validity of this assumption is to be verified.

Another factor of interest is the transient behaviour of a TCP connection. In the short term, the variation of the TCP throughput around the expected value could be of the same order as the expected throughput itself. For such time scales, a non-TCP flow may be considered reasonably TCP-friendly despite a 65% to 285% overallocation or underallocation of bandwidth. It remains to be seen if these time scales are of the same order as the life time of end-to-end network connections.

The stochastic TCP model which was used to derive the throughput formula assumes that packet loss in one TCP window is uncorrelated with packet loss in other windows. This may be reasonable in case of drop-tail routers, but a suitable adaptation for gateways using the Random Early Detection (RED [9]) algorithm remains to be seen. In addition, the model assumes no correlation between the TCP window size and the RTT which makes it unreliable when the receivers use a modem line.

References

- [1] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of ACM SIGCOMM'88*, August 1988.
- [2] S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic," *Computer Communications Review*, vol. 21, no. 5, October 1991.
- [3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modelling TCP Throughput: A simple model and its empirical validation," *Proceedings of SIGCOMM '98*, 1998.
- [4] J. Padhye and J. Kurose and D. Towsley and R. Koodli, TCP-Friendly Rate Adjustment Protocol for Continuous Media Flows over Best Effort Networks, *Proceedings of NOSSDAV'99*.

- [5] M. Handley, and S. Floyd, “Strawman Specification for TCP Friendly (Reliable) Multicast Congestion Control (TFMCC), *Reliable Multicast Research Group*, December 1998.
- [6] J. Mahdavi, and S. Floyd, “TCP-Friendly Unicast Rate-Based Flow Control,” *Technical Note on End2end Mailing List*, January 1997.
- [7] T. Ott, J. Kemperman, and M. Mathis, “Window Size Behavior in TCP/IP with Constant Loss Probability,” *DIMACS Workshop on Performance of Realtime Applications on the Internet*, Plainfield NJ, November 1996.
- [8] D. Sisalem, F. Emanuel, and H. Schulzrinne, “The Direct Adjustment Algorithm: ATCP-Friendly Adaptation Scheme,” *Preprint*, Columbia University, 1998.
- [9] S. Floyd and V. Jacobson, “Random Early Detection gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, August 1993, pp. 397-413.

A Proof of Theorem 1

LEMMA 1 For any q such that $0 < q < 1$, $F_m(q) = \frac{1+q+q^2+\dots+q^{m-1}}{m}$ is a monotonically decreasing function m .

PROOF : $F_{m+1}(q) = \frac{1+q+q^2+\dots+q^{m-1}+q^m}{m+1}$. $q < 1$.

Hence, $q^m < q^{m-1} < q^{m-2} < \dots < q^2 < q < 1$.

$F_{m+1}(q)$ can be rewritten as $\frac{1+q^m/m+q+q^m/m+q^2+q^m/m+\dots+q^{m-1}+q^{m-1}/m}{m+1}$.

Thus, we have:

$$F_{m+1}(q) < \frac{1 + 1/m + q + q/m + \dots + q^{m-1} + q^{m-1}/m}{m+1} = \frac{1 + q + \dots + q^{m-1}}{m} = F_m(q)$$

LEMMA 2 If $G_m(q) = F_m(q) \times m = 1 + q + q^2 + \dots + q^{m-1}$, and $M = \frac{\sum_{i=1}^N m_i}{N}$, then

$$\frac{\sum_{i=1}^N G_{m_i}(q)}{N} \leq G_M(q)$$

PROOF : Consider the sequence $\{q^{m_1}, q^{m_2}, \dots, q^{m_N}\}$. Its arithmetic mean is given by $AM(q) = \frac{\sum_{i=1}^N q^{m_i}}{N}$. Its geometric mean is $GM(q) = q^M$, where $M = \frac{\sum_{i=1}^N m_i}{N}$. $AM(q) \geq GM(q)$. Also, $0 < q < 1$, so $(1 - q) > 0$. Therefore, $\frac{1-AM(q)}{1-q} \leq \frac{1-GM(q)}{1-q}$. But $\frac{1-AM(q)}{1-q} = \frac{\sum_{i=1}^N G_{m_i}(q)}{N}$, and $\frac{1-GM(q)}{1-q} = G_M(q)$.

We have:

$$l_{app} = \frac{\sum_{i=1}^N \sum_{j=1}^{W_i} (1 - l_{act})^{j-1} l_{act}}{\sum_{i=1}^N W_i} = \frac{\sum_{i=1}^N G_{W_i}(1 - l_{act}) \times l_{act}}{\sum_{i=1}^N W_i}$$

Therefore, from Lemma 2, it is seen that:

$$l_{app} \leq \frac{G_{EW}(1 - l_{act}) \times l_{act}}{EW}, \quad \text{where } EW = \frac{\sum_{i=1}^N W_i}{N} \quad (1.7)$$

But $N \times EW = \sum_{i=1}^N W_i > \mu T$. Hence, from Lemma 1, we have:

$$\frac{N \times G_{EW}(1 - l_{act}) \times l_{act}}{N \times EW} < \frac{N \times G\left(\frac{T\mu}{N}\right)(1 - l_{act}) \times l_{act}}{T\mu} \quad (1.8)$$

Recognize that the RHS in (1.8) is l_{est} . Thus, combining (1.8) and (1.7) gives us the proof of Theorem 1.

B Proof of Theorem 2

From Lemma 1, it follows that $\frac{G_m(q)}{m} < \frac{G_k(q)}{k}$, for $k < m$. Therefore, $k \times G_m(q) < m \times G_k(q)$. Observe that l_{est} may be rewritten as :

$$l_{est} = \frac{\sum_{i=1}^N l_{act} \times G_{(T\mu/N)}(1 - l_{act})}{T\mu} = l_{act} \times N \times \frac{G_{(T\mu/N)}(1 - l_{act})}{T\mu}$$

Therefore, $l_{est} \times \frac{T\mu}{N} = l_{act} \times G_{\left(\frac{T\mu}{N}\right)}(1 - l_{act})$. But $\frac{T\mu}{N} > W_i$, for $1 \leq i \leq N$.

Hence, for $1 \leq i \leq N$, we have:

$$G_{\left(\frac{T\mu}{N}\right)}(1 - l_{act}) \times W_i < G_{W_i}(1 - l_{act}) \times \frac{T\mu}{N}$$

Taking the summation over $1 \leq i \leq N$, we get:

$$\sum_{i=1}^N G_{\left(\frac{T\mu}{N}\right)}(1 - l_{act}) \times W_i \times l_{act} < \sum_{i=1}^N G_{W_i}(1 - l_{act}) \times \frac{T\mu}{N} \times l_{act} \quad (2.9)$$

Simplifying both sides, we get:

$$l_{est} \times \frac{T\mu}{N} \times \sum_{i=1}^N W_i < l_{app} \times \frac{T\mu}{N} \times \sum_{i=1}^N W_i$$

Or, $l_{est} < l_{app}$.

C Proof of Theorem 3

We have:

$$l_{est} = \frac{\sum_{i=1}^N l_{act} \times G_{(T\mu/N)}(1 - l_{act})}{T\mu} = l_{act} \times \frac{G_{(T\mu/N)}(1 - l_{act})}{T\mu/N} = l_{act} \times F_{(T\mu/N)}(1 - l_{act})$$

But $\frac{T}{N} = RTT$, and $\frac{T\mu}{N} > 1$. Therefore, from Lemma 1, it follows that

$$F_{(T\mu/N)}(1 - l_{act}) < F_1(1 - l_{act}) = 1$$

Hence, $l_{est} < l_{act}$.