An Extension of the TCP Steady-State Throughput Equation for Parallel TCP Flows

Dragana Damjanovic(student), Werner Heiss(faculty), Michael Welzl(faculty)
Institute of Computer Science
University of Innsbruck
Innsbruck, Austria
dragana.damjanovic|werner.heiss|michael.welzl@uibk.ac.at

I. Introduction

Almost a decade after its publication in [1], the steady-state throughput equation of TCP by Padhye et al. remains the most widely used method for calculating the throughput that a TCP sender obtains under certain environment conditions. While there is now a wealth of other models available, many of which are better in some aspect, none of them seems to strike the same balance between precision and ease of use that makes the equation by Padhye et al. the useful tool that it is.

Recently, some effort was made to model a number of TCP flows which share the same bottleneck at the same time instead of focusing on just a single one (the list of such models is omitted here due to space constraints). Their practical use is however limited because it is either assumed that the flows under consideration are the only flows in the network, or the model is too complicated.

In an effort to enable practical calculation of the throughput of several TCP flows across a real end-to-end Internet path, we present an extension of the equation by Padhye et al. to multiple flows. We do this by following the basic approach in [1], but considering a number of senders at the same time instead of a single one; this way we obtained an approximation of the throughput of a single end-to-end TCP flow based on the round-trip time (RTT) and packet loss (more precisely "loss event") ratio. Our approach still needs to be further refined and extended (e.g., slow start and the impact of the receiver window are still missing).

II. The model

In order to derive an equation for the throughput of several parallel TCP flows we extend the model presented in [1]. We assume that the reader is familiar with this paper and therefore will only repeat preliminary assumptions where needed and shortly repeat necessary definitions.

Consider n parallel TCP flows $f_1, ..., f_n$ starting at time $t = 0$. As in [1] we model TCP's congestion avoidance phase in terms of "rounds", assuming furthermore that the flows are synchronized in terms of rounds (i.e. in a round all flows send their current window size W_f before the next round starts for all flows, see also figure 1).

[1]The work described in this paper is supported by the European Union through the FP6-2006-IST-045256 project EC-GIN.

[2]http://www.dps.uibk.ac.at/~dragana/index.htm

For any given time $t \geq 0$ we define N_t as the number of packets transmitted by all flows in the interval $[0, t]$. Let $B_t := N_t/t$ be the cumulative throughput of all flows on that interval. Then we can define the long term steady-state throughput

$$B := \lim_{t\to\infty} B_t = \lim_{t\to\infty} \frac{N_t}{t}.$$

Let W be the cumulative window size of all flows, b the number of packets acknowledged by a received ack and TD denote a "triple duplicate" acknowledgment, i.e. the receipt of four acks with the same sequence number. We only consider TD events as loss indication here, meaning that the flows stay in congestion avoidance phase. Nevertheless the equation we derive already works reasonably well, as validations show in section III.

Whenever a flow f experiences a TD loss indication, it reduces its corresponding window size W_f to half. For n parallel flows we define a TD-period (or TDP) to be a period between two consecutive TD loss indications in whatever flow. Notice that a TDP is defined for all flows simultaneously (figure 1). Let p be the loss probability as defined in [1]. We

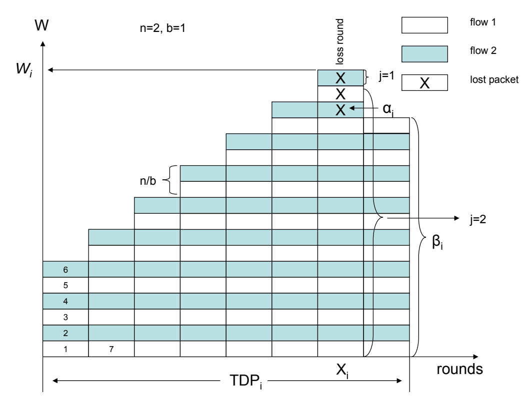

Fig. 1. A Triple Duplicate Period (TDP)

assume that all flows share the same path and have the same average round-trip time. The flows are mixed (i.e. in a round packet 1 belongs to f_1, packet 2 to f_2, and so on). As in [1], we also assume that packet losses are correlated within a round, meaning that if a packet is lost all consecutive packets in the same round are lost.

For a period TDP_i let Y_i be the number of packets sent in that period and A_i be the duration of that period. It can be shown ([1]) that

$$B = \frac{E[Y]}{E[A]} \ . \qquad (1)$$

In each round W is incremented by n/b, hence the number of packets sent per round is incremented by n every b rounds. α_i denotes the sequence number of the first packet lost in TDP_i and X_i the round in which this loss occurs. In analogy to [1] we obtain a total number of $Y_i = \alpha_i + W_i - 1$ packets sent in $X_i + 1$ rounds in TDP_i, hence

$$E[Y] = E[\alpha] + E[W] - 1 \ . \qquad (2)$$

The derivations of $E[\alpha]$, $E[A]$ are still the same as in the original paper and we have

$$E[Y] = \frac{1-p}{p} + E[W] \ , \qquad (3)$$

$$E[A] = (E[X] + 1)RTT \ , \qquad (4)$$

where RTT stands for the average value of the round trip time.

Looking at the evolution of the window size W we have to consider the number j_i of flows that experience loss in the penultimate round (loss round) of period TDP_i. For $i = 1, 2, \ldots$ we have

$$W_i = W_{i-1}\frac{n - \frac{j_i}{2}}{n} + (n - j_i) + \frac{nX_i}{b} \ . \qquad (5)$$

Let β_i be the cumulative number of packets sent in the last round, then we can write

$$Y_i = \sum_{k=0}^{X_i/b-1}\left(W_{i-1}\frac{n - \frac{j_i}{2}}{n} + (n - j_i) + nk\right)b + \beta_i =$$
$$X_i\left(W_{i-1}\frac{n - \frac{j_i}{2}}{n} + (n - j_i)\right) + \frac{{X_i}^2}{2b} - \frac{X_i}{2} + \beta_i \ . \qquad (6)$$

From equations (3) and (6) we obtain

$$\frac{1-p}{p} + E[W] = E[X]E[W] - \frac{E[X]E[j]}{4n} + \frac{nE[X]}{2} + E[\beta] \qquad (7)$$

(assuming that W, X and j are mutually independent), and from equation (5)

$$E[W]E[j] = 2n^2 - 2nE[j] + \frac{2n^2E[X]}{b} \ . \qquad (8)$$

To derive expressions for $E[j]$ and $E[\beta]$ we assume that loss occurs identically distributed over the size of the window of the penultimate round (figure 1), hence

$$E[\beta] = \frac{E[W]}{2} \qquad (9)$$

and, assuming $E[1/W] = 1/E[W]$ we have

$$E[j] = \frac{n^2}{2E[W]} - \frac{n}{2E[W]} + \frac{n(E[W] - n + 1)}{E[W]} \ . \qquad (10)$$

Substituting equations (9) and (10) in (8) and (7) we obtain

$$E[X] = \frac{b(2E[W]^2 + (1-n)E[W] - 2n^2 + 2n)}{4nE[W]} \qquad (11)$$

and a quartic equation for $E[W]$

$$0 = 12bpE[W]^4 + (4bp - 16np - 12bnp)E[W]^3 +$$
$$(6bnp + 32np - 5bn^2p - bp - 32n)E[W]^2 +$$
$$(4bn^3p - 4bnp)E[W] - (4bn^4p + 8bn^3p - 4bn^2p \ .$$

There is a symbolic solution of this equation that can be obtained by symbolic analysis tools or e.g. by Ferrari's method. The resulting expressions are left out here, since they are far from readable. By picking the right solution for E[W] (now only depending on n, p and b) and plugging in expression (11) into (4), we arrive at formulas for $E[A]$ and $E[Y]$ and hence have a formula (equation (1)) for the cumulative throughput B of n flows, depending only on n, p, b and RTT.

III. Validation

As our validations show, our equation performs well in simulations (using the ns-2 simulator) with a wide range of realistic loss conditions and in real-life measurements (using PlanetLab - `http://www.planet-lab.org`).

For our ns-2 simulations we used a simple network configuration called "dumbbell". We varied parameters, like the loss percentage and the number of flows, to validate our model in different network conditions.Having in mind that the real loss distribution is a combination of independent random and burst loss events, we checked both extreme ends of this spectrum by comparing our equation with a loss model with independent random losses and with a two-state Gilbert model. We also used a network configuration with background traffic. Because of lack of space here we show just results with 1, 2, 3 and 5 flows using ns-2 simulator and a random loss model. The rest of our measurements will be presented on the poster.

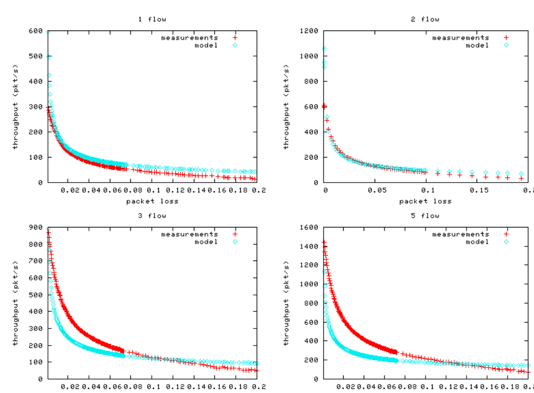

Fig. 2. 1, 3 and 5 flows ns-2 with random loss model

References

[1] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication,* pp. 303–314, 1998.