$jwperiod. \ \ font=small, label sep=jwperiod, justification=justified, single line check=false$

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS

Euro. Trans. Telecomms. **00**: 2–**??** (0000) Published online in Wiley InterScience

(www.interscience.wiley.com) DOI: 10.1002/ett.0000



Performance of TCP on low-bandwidth wireless links with delay spikes

Pasi Lassila

Pirkko Kuusela

Helsinki University of Technology, P.O.Box 3000, FIN-02015 TKK, Finland.

Email: Pasi.Lassila@tkk.fi, pkuusela@kolumbus.fi

SUMMARY

We model the goodput of a single TCP source on a wireless link experiencing sudden increases in RTT, i.e., delay spikes. Such spikes trigger spurious timeouts that reduce the TCP goodput. Renewal reward theory is used to derive a straightforward expression for TCP goodput that takes into account limited sending rates (limited window size), lost packets due to congestion and the delay spike properties such as the average spike duration and distribution of the spike intervals. The basic model is for i.i.d. spike intervals, and correlated spike intervals are modeled by using a modulating background Markov chain. Validation by ns2 simulations shows excellent agreement for lossless scenarios and good accuracy for moderate loss scenarios (for packet loss probabilities less than 5%). Numerical studies have also been performed to assess the impact of different spike interval distributions on TCP performance.

Keywords: TCP performance, wireless networks, spurious timeouts, delay spikes, renewal reward theory, semi-regenerative processes Copyright © 0000 AEIT

1. Introduction

TCP is the most widely used transport protocol for reliable data transmission over the Internet. In addition to offering a reliable packet delivery service, TCP also includes functionality for controlling the packet sending rate to avoid congesting the network. However, this functionality has been originally designed based on the characteristics of the fixed network. There the basic operational principle of TCP rate control is roughly that TCP increases its rate as long it is receiving acknowledgements correctly from the receiver, and as soon as a packet loss is detected, it is interpreted to indicate that there is congestion in the network and hence the sending rate should be reduced considerably. As a result of a loss the sending rate is either halved or it can even be reduced back to an initial small value, as happens in the case when the loss is detected via a timeout. To set the value of the timeout timer TCP relies on an adaptive estimation algorithm of the RTT (Round Trip Time). The underlying assumption in

the estimation is that changes in the RTT due to random fluctuations of the traffic in the Internet can be tracked but sudden unexpected considerable increases are interpreted as a sign of congestion and hence the timer will expire, triggering TCP's Go-Back-N retransmission scheme, and the sending rate is initialized according to the configured initial window size.

When considering the operation of TCP over a relatively low bandwidth wireless link, the delays in the observed RTTs of TCP can be highly variable. Moreover, the pattern of variability can be such that the measured RTTs contain very sharp spikes which can be even an order of magnitude larger than the typical measured RTT, see the measurement studies in GPRS networks [9],[14] and [11]. Furthermore, the magnitude of the observed RTTs are typically hundreds of milliseconds or even seconds, see [7] and [13]. The RTT estimation algorithm of TCP can not track such sudden considerable increases in the measured RTTs. These sudden increases in RTTs are called *delay*

Copyright © 0000 AEIT

Prepared using ettauth.cls [Version: 2007/01/05 v1.00]

spikes and potential reasons for their occurrence can be for example [10]: handovers typically result in delay spikes of several seconds occurring at a time scale of minutes in urban environment; link layer error recovery (reliable link layer protocols are usually used in modern cellular systems, such as EDGE/GPRS and UMTS) may also cause delay spikes, especially when the radio channel conditions change abruptly due to the mobile's movement, e.g., when entering a tunnel; scheduling of radio resources between circuit switched calls and data (as in GPRS) can cause delay spikes. These delay spikes trigger so called spurious timeouts and result in unnecessary retransmissions and congestion control actions on the part of TCP, as the packets are not lost, they are simply delayed. To enable TCP to handle RTT spikes, some experimental algorithms have been proposed, namely the Eifel algorithm [10],[17], or F-RTO algorithm [24], but even they can not completely remove the effect of delay spikes, and hence evaluating their impact is necessary.

In this paper, we consider a relatively low bandwidth wireless link, e.g., in a GPRS network, with the radio link operating in the reliable mode. A simple modeling framework is provided to study the impact of spurious timeout events on the long run (steady state) goodput of TCP, i.e., on the amount of successfully sent traffic per time unit. In addition to the detrimental impact of spurious timeouts on TCP goodput, we also include the impact of congestion related losses in our model, whereas in our earlier paper [16] a lossless low bandwidth GPRS wireless link was considered. On a lossless wireless link a realization of the evolution of TCP sending rate is determined by the following: An RTT spike creates a period of silence during which there is a (spurious) timeout. Then the source starts increasing its sending rate exponentially (slow start) until it reaches a window size corresponding to the maximum sending rate of the physical link given the RTT. The source keeps sending at the maximum rate until there is an RTT spike again and the process repeats. The impact of congestion losses occurring elsewhere along the route of the TCP flow (for example, in the fixed network) is that the sending rate evolution must also account for the characteristic sawtooth pattern and possible congestion related timeouts.

In modeling, the first observation is that these spurious timeouts are caused by random events occurring in the mobile's environment, which are also independent of TCP's packet sending characteristics. Therefore, we model these events as an outside disturbance with various levels of generality regarding the inter-occurrence time of the RTT spikes which trigger spurious timeouts. Initially,

the impact of the length of the RTT spike is taken into account by considering only the mean value of these spikes. In between RTT spikes, we incorporate the above aspects of the sending rate evolution by approximating the exact stochastic sample path behavior by sample path averages assuming that packet losses occur independently with a given probability. First the case of i.i.d. (independent and identically distributed) times between RTT spikes is considered in the framework of renewal reward processes. Then a generalization by using Markov renewal reward theory is given, where the distributions between RTT spikes can be modulated by a background discrete time Markov chain (thus producing correlated times between the RTT spikes). The models are validated through ns2 simulations. The results show excellent agreement for lossless scenarios. Congestion losses have a noticeable impact on performance already when packet loss probability is comparably low, say 2%, and in such circumstances the accuracy of our model is still good, but for higher loss probabilities, say higher than 5%, our models underestimate the performance somewhat. Additionally, we show how the model can be further generalized to include the distribution of the lengths of the RTT spikes (and not only its mean).

The paper is organized as follows: The renewal reward theory and models without and with congestion losses are described in Section 2 in the i.i.d. case. Then several extensions, e.g., non-i.i.d. spike intervals, are outlined in Section 3. In Section 4 results are compared to ns2 simulations and sensitivity on the RTT spike process is studied. Finally, conclusions are given in Section 5.

1.1. Related work

TCP modeling philosophies can be roughly categorized in two: TCP modeling from the point of view of a single TCP connection and TCP bandwidth sharing modeling with multiple flows. We first discuss the modeling of TCP bandwidth sharing, which mostly focuses on fixed networks. In fixed networks, fluid models have been applied extensively for modeling multiple TCP flows sharing the network resources under a static number of flows. In fluid models, the behavior of TCP's sending rate evolution is modeled by differential equations and the impact of losses depends on the packet level model, see, e.g., [20]. It is commonly assumed that during a loss epoch all flows are affected equally. When RED (or any other AQM method) is used, this is a valid assumption since RED will randomize the losses [20]. However, with Drop Tail routers the assumption of all flows being equally

Copyright © 0000 AEIT Prepared using ettauth.cls

affected may not hold, for example flows with higher instantaneous windows will suffer more than flows with small windows. Models including this kind of proportional loss rates are given in [6] and [18]. Typically, fluid models have been used to analyze the (control theoretic) stability of the system. Finally, so called processor sharing models have been applied to model the case where flows have a finite length, they arrive randomly and TCP implements the fair sharing of the resources, see, e.g., [23].

However, in this paper the modeling view point concerns a single TCP flow. In this situation, the idea is to abstract the influence of the network to the notion of the packet loss process, and the interaction with other TCP flows is not considered. A lot of research has been done in modeling the steady state throughput of a single persistent TCP flow in fixed networks with a given packet loss process. The simple 'square-root-p' formula was derived, e.g., in [19]. Using more complex assumptions on the nature of the packet loss process more refined formulas have also been derived, see, e.g., [22] and [3]. Notably, in [3] the loss process can be an arbitrary point process and the throughput of the congestion control phase can be shown to depend on the correlations such that any positive correlations between loss events actually improves the throughput (deterministic loss process being the worst case). Recent work on modeling the delay of non-persistent TCP flows (i.e., the file transfer delay) in fixed networks includes [4] and [5]. The authors in [4] model the rate evolution of TCP for short files at a very detailed level. The approach includes a packet level model of TCP to determine the packet loss probability, which is based on assuming that TCP flows share a single bottleneck link. The work in [5] considers the throughput of a single finite TCP flow in isolation experiencing random losses. By using the properties of so called Mellin transforms, results on the mean throughput and the distribution of the throughput have been obtained.

In TCP models for wireless channels, the per packet loss process is typically presented by a two-state Markov process, representing the wireless channel's alteration between good/bad states, resulting in correlated packet losses, see, e.g., [2],[15], [26]. However, these models do not include spurious timeouts (RTT delay spikes).

Work related to modeling spurious timeouts has been done in [8], where a model for TCP Reno experiencing spurious timeouts has been given which very closely follows the operation of the actual TCP Reno protocol. The model is based on an extension of the well known model in [22] and considers a large bandwidth delay product, with no sending rate limitations (we include the sending rate

limit and congestion losses). The model takes into account packet losses, but the effect of delay spikes are modeled only through the mean length and mean time between the spikes, whereas our model allows to distinguish between different distributions and non-i.i.d. processes. In [1] the measured RTT process has been modeled with a semi-Markov process and from the dynamics of the model the behavior of TCP has been deduced, including the occurrence of spurious timeouts and spurious duplicate ACKs (packets can also be reordered due to delay spikes caused by handoffs). The model does not consider the impact of congestion losses.

In comparison to [8] and [1], using a more abstract representation of how the TCP's goodput accumulates (and not attempting to model the protocol itself), we obtain a model that is straightforward to utilize and which allows quite general RTT spike processes. Finally, in comparison to our earlier paper [16], the present paper includes the impact of congestion losses (i.e., a lossless scenario is not assumed anymore). Also, we generalize the model to allow the RTT spikes to have a distribution (instead of representing the spike durations only through their mean).

2. Renewal model for TCP performance

In this section, the basic model for TCP goodput is derived under the assumption of i.i.d. distributed times between the RTT spikes. We first introduce the theoretical framework, and subsequently give the models without and with congestion losses, respectively.

2.1. Preliminaries and the renewal reward theorem

We model the TCP goodput (amount of successfully sent traffic per time unit) of one persistent source on a wireless link that represents the joint maximum rate of the time slots reserved for a given user in a GPRS system. The modeling abstraction here is that the impact of the wireless interferences is experienced by TCP as unexpectedly long RTT spikes. Typically, TCP models are discrete and consider the behavior per packet or per RTT round, as in [22]. Our abstraction is a continuous-time model describing the actual TCP sending rate that produces goodput. Similar continuous time approximations have been applied in fixed networks in [12] and [21]. However, common to [22], [12] and many other papers is the use of renewal theory.

In this scenario, the goodput of TCP is determined by an external stochastic process that generates the RTT spikes that lead to TCP timeouts, and possible congestion

Copyright © 0000 AEIT Prepared using ettauth.cls

related loss events. The intervals between the spikes are random and the duration of the spike is modeled using its mean duration, denoted by SD. The evolution of TCP's (successful) sending rate is described by a) a silence for time SD due to the RTT spike, and b) sending rate evolution according to TCP's rate control algorithms.

Remark 1: During the RTT spike the TCP protocol is likely to perform an exponential back-off (depending on the length of the spike compared to the RTT). If the RTT spike duration is short (< 10 seconds), it is convenient to model the time when no packets get through as the duration of the RTT spike. In this case the times when TCP attempts to send packets is close to the time when the link becomes capable of delivering packets again. If the RTT spike duration is long (15 sec or more) the TCP exponential back-off determines when sending becomes possible.

We model the long term average goodput of the TCP connection by applying the framework of renewal models. Let Z_n denote the nth RTT spike, and take $Z_0=0$. These are the renewal times. The length of the nth renewal period is denoted by $T_n=Z_n-Z_{n-1}$. Consider as the reward the number of packets sent during the nth renewal cycle, denoted by A_n . By denoting with $X_n(t)$ the sending rate at time t during the nth cycle, A_n equals

$$A_n = \int_0^{T_n} X_n(t) \, dt.$$

Assume that the T_n s are i.i.d. random variables with finite mean E(T) and probability density function f_T . According to the well-known renewal reward theorem (see, e.g., [25]), the time-averaged total reward, i.e., TCP goodput G_{TCP} , converges to cycle averaged rewards,

$$G_{\text{TCP}} = \frac{E(A)}{E(T)}.$$
 (1)

2.2. Basic model without congestion losses

To derive more explicit forms for the TCP goodput formula we first assume a lossless scenario. In this case, the evolution of TCP's (successful) sending rate is described entirely by a) a silence for time SD due to the RTT spike, b) an exponential increase in the sending rate (the slow start phase), and c) reaching an upper bound on the sending rate due to the low bandwidth of the wireless link. As defined above, the time between the nth and (n+1)th RTT spike represents the nth renewal period in our model and its length is denoted by T_n .

The evolution of the time derivative of the TCP's actual sending rate during nth renewal period, X_n , is modeled

Copyright © 0000 AEIT Prepared using ettauth.cls

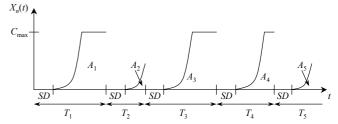


Figure 1: Notation for the renewal model.

according to the phases described earlier

$$\dot{X}_n(t) = \begin{cases} 0, & t < SD, \\ \frac{\ln 2}{R} X_n(t) dt, & SD \le t \le L + SD, \\ 0, & t > L + SD, \end{cases}$$
 (2)

where

 $X_n = \text{TCP}$ sending rate, with $X_n(SD) = 1$,

R = TCP's RTT,

SD = RTT spike duration (no goodput increase),

L =time to reach maximum sending rate C_{\max} .

For simplicity we have taken $X_n(SD)=1$. Also note that in (2) the term $\frac{\ln 2}{R}$ implies that the rate doubles for every RTT (slow start). The notation introduced above is illustrated in Figure 1. The solution of (2) is

$$X_n(t) = \begin{cases} 0, & t < SD, \\ 2^{\frac{t-SD}{R}}, & SD \le t \le L + SD, \\ C_{\text{max}}, & t > L + SD, \end{cases}$$
(3)

where $L = \frac{R}{\ln 2} \ln C_{\text{max}}$.

Our reward A_n , the number of packets sent during the nth renewal cycle, is then given by

$$A_{n} = \int_{0}^{T_{n}} X_{n}(t)dt$$

$$= \begin{cases} 0, & T_{n} < SD, \\ \frac{R}{\ln 2} (2^{\frac{T_{n} - SD}{R}} - 1), & SD \leq T_{n} \leq L + SD, \\ \frac{R}{\ln 2} (2^{\frac{L}{R}} - 1) + \\ C_{\max}(T_{n} - L - SD), & T_{n} > L + SD. \end{cases}$$
(4)

To compute E(A) needed in (1), we integrate (4) over the distribution of the delay spikes, f_T ,

$$\begin{split} E(A) &= E\left(\int_0^T X(t)dt\right) \\ &= \frac{R}{\ln 2} \left(\int_{SD}^{L+SD} (2^{\frac{y-SD}{R}} - 1) f_T(y) dy \right. \\ &\left. + (C_{\max} - 1) \int_{L+SD}^{\infty} f_T(y) dy \right) \\ &\left. + C_{\max} \int_{L+SD}^{\infty} (y - L - SD) f_T(y) dy. \end{split}$$

and then the TCP goodput is given by (1), i.e., $G_{\rm TCP}=E(A)/E(T)$. The parameters in the formula are the round trip time R, the bandwidth limitation $C_{\rm max}$, the RTT spike duration SD and the distribution of the spikes.

Example with exponential distribution: The expression for the expected reward simplifies considerably if we assume that the intervals between RTT spikes follow the exponential distribution with parameter 1/E(T). In that case the goodput becomes

$$\begin{split} G_{\text{TCP}} &= \frac{E(A)}{E(T)} \\ &= \frac{e^{-SD/E(T)} \left(\frac{R}{E(T)} - \ln 2C_{\text{max}}^{1-R/(E(T)\ln 2)}\right)}{R/E(T) - \ln 2}. \end{split}$$

Observe that here the goodput actually depends on the time ratios $\mathrm{SD}/E(T)$ and R/E(T), in addition to the bandwidth limitation C_{max} .

2.3. Model including congestion losses

In this section the model is generalized such that the impact of congestion related losses, timeouts and slow starts are approximately included in the model. Even though the wireless link is assumed to be lossless, congestion related packet losses can occur elsewhere along the path of the TCP connection, e.g., in the fixed part of the network.

Essentially, congestion losses make the sending rate of the TCP connection fluctuate according to the familiar sawtooth pattern with randomly varying intervals between the rate drops. Our model does not try to exactly model this stochastic behavior. Instead, our approach is to replace the exact stochastic behavior with a model where the time between losses is a constant representing the mean time between losses. We use the term *congestion cycle* to refer to the rate evolution during the time between loss events.

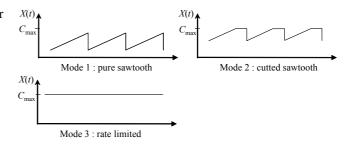


Figure 2: The three different operation modes of TCP under congestion avoidance.

To be more precise, it is assumed that packet losses occur with a probability p > 0 per packet. Each congestion cycle begins with the retransmission of the packet that was lost at the end of the previous congestion cycle, and after that 1/p-1 packets are sent until a new loss occurs (number of packets sent before packet loss is geometrically distributed). Thus, altogether 1/p packets are sent in each cycle. We assume that the typical operation of TCP under congestion losses obeys the so called congestion avoidance rules (linear increase, multiplicative decrease by factor 1/2, no timeouts or slow starts). Depending on p, the round trip time R and the rate limit C_{\max} , we distinguish between three different operation modes for TCP in congestion avoidance: (1) pure sawtooth, (2) cutted sawtooth, and (3) rate limited. Pure sawtooth refers to a situation, where the system parameters are such that TCP rate fluctuates according to a perfect saw tooth pattern with slope equal to (1/R). Cutted sawtooth is a situation, where the rate evolution has a linear increase component with the same slope as in pure sawtooth, but rate increase is limited to the physical rate limit $C_{\rm max}$. Finally, rate limited refers naturally to the case where the sending rate evolution is only limited by the physical limit $C_{\rm max}$. These different modes are illustrated in Figure 2.

The idea is then to compute the time it takes to send 1/p packets assuming the above three possible operation modes. This is denoted by $\tau_{\rm CA}$, with $\tau_{\rm CA}$ depending on the system parameters $C_{\rm max}$, p and R. However, just capturing the congestion avoidance behavior of TCP is not enough. To include the impact of congestion related timeouts and slow starts in the model, we also compute $\tau_{\rm SS}$, which denotes the time it takes to send 1/p packets assuming the same operation mode as used in $\tau_{\rm CA}$ with an idle period of length TO (modeling the mean length of a timeout period), an exponential increase of rate up to a threshold value (which equals the starting value of the corresponding $\tau_{\rm CA}$ cycle), and the same rate evolution mode as in the

corresponding cycles of length τ_{CA} . Associated with each cycle is a probability, β , that the cycle is a timeout-slow start cycle or a congestion avoidance cycle, with probability $(1-\beta)$. The probability of a timeout cycle is taken from [22], where a formula is given for the probability of a timeout given the window size w just prior to the loss event and the packet loss probability p,

$$\beta(w,p) = \min\left(1, \frac{(1-q^3)(1+q^3(1-q^{w-3}))}{1-q^w}\right), (5)$$

where q=1-p. Figure 3 illustrates the above by showing the assumed rate evolution of TCP during a congestion cycle (with length $\tau_{\rm CA}$) followed by a slow start cycle (with length $\tau_{\rm SS}$) for the three TCP operation modes.

To keep the presentation at this point succinct we leave the exact derivation of τ_{CA} and τ_{SS} until later as it involves lengthy, but straightforward calculations (also heuristics for fixing w in (5) are discussed later). The details can be found in Appendix. It suffices at this point to recall that depending on the parameters p, R and C_{max} , we fix the TCP mode and compute τ_{CA} . Associated with this we compute the length of the timeout-slow start cycle, τ_{SS} , and its probability β . Observe that, in the lossless model in the previous section, we assumed a slowstart after each RTT spike. In contrast to this, now we have slowstarts related to congestion loss timeouts with probability β , but there is no slowstart after each RTT spike.

2.3.1. Computation of the reward By conditioning on T, the random variable for the time between delay spikes, we

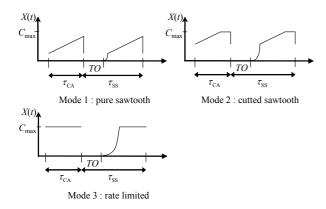


Figure 3: Illustration of $\tau_{\rm CA}$ and $\tau_{\rm SS}$ for the three TCP modes.

Copyright © 0000 AEIT Prepared using ettauth.cls can express E(A) needed in (1) as,

$$E(A) = E(E(A \mid T))$$

$$= \int_{SD}^{\infty} E(A \mid t - SD) f_T(t) dt.$$
 (6)

The problem is now to determine $E(A \mid T')$, where T' = T - SD. The rate evolution between delay spikes is represented in our model by two types of cycles with different lengths, $\tau_{\rm CA}$ and $\tau_{\rm SS}$, each with probabilities $(1-\beta)$ and β , respectively. Note that $\tau_{\rm CA} < \tau_{\rm SS}$ for each mode, as both cycles send 1/p packets. We assume that when computing $E(A \mid T')$, the contribution from the last cycle before T' is reached can, without loss of accuracy, be approximately taken into account by assuming that the number of sent packets during the remainder equals $(1/p) \cdot \tilde{t}/\tau$, where \tilde{t} is the remaining time until T' and τ equals $\tau_{\rm CA}$ or $\tau_{\rm SS}$ depending on whether the last cycle is of type $\tau_{\rm CA}$ or $\tau_{\rm SS}$. This implies that it is assumed that during the last unfinished cycle, packets are sent at a constant rate.

Let n denote the number of *complete* τ_{CA} cycles and m the number of *complete* τ_{SS} cycles in a sequence which is stopped right after T' is exceeded. The probability of such a sequence obeys the binomial distribution,

$$Bin(n, m, \beta) = \binom{n+m}{m} \beta^m (1-\beta)^n.$$

To compute E(A | T') we need to consider whether exceeding is caused by a short (τ_{CA}) or long cycle (τ_{SS}) . If it is caused by a short cycle (with probability $(1 - \beta)$), the set of admissible pairs of integer numbers (n, m), denoted by \mathcal{S}_1 , is defined as

$$S_1 = \{ (n, m) : \\ n\tau_{\text{CA}} + m\tau_{\text{SS}} < T' \land (n+1)\tau_{\text{CA}} + m\tau_{\text{SS}} \ge T' \},$$

and the number of sent packets equals $(1/p) \cdot (n+m+(T'-n\tau_{\text{CA}}-m\tau_{\text{SS}})/\tau_{\text{CA}})$. If it is a long cycle (with probability β) that results in exceeding T', the set of admissible pairs of integer numbers (n,m), denoted by \mathcal{S}_2 , is defined as,

$$S_2 = \{(n, m) : n\tau_{CA} + m\tau_{SS} < T' \land n\tau_{CA} + (m+1)\tau_{SS} \ge T'\},$$

and the number of sent packets equals $(1/p)\cdot(n+m+(T'-n\tau_{\rm CA}-m\tau_{\rm SS})/\tau_{\rm SS})$. Thus, $E(A\,|\,T')$ can be

expressed as

$$E(A \mid T') = \frac{1}{p} \cdot \left(\sum_{(n,m) \in \mathcal{S}_1} g_1(n,m,\beta) + \sum_{(n,m) \in \mathcal{S}_2} g_2(n,m,\beta) \right), \quad (7)$$

where the functions g_1 and g_2 are given by

$$\begin{split} g_1(n,m,\beta) &= \\ &\mathrm{Bin}(n,m,\beta)(1-\beta) \left(n + m + \frac{T' - n\tau_{\mathrm{CA}} - m\tau_{\mathrm{SS}}}{\tau_{\mathrm{CA}}} \right), \\ g_2(n,m,\beta) &= \\ &\mathrm{Bin}(n,m,\beta)\beta \left(n + m + \frac{T' - n\tau_{\mathrm{CA}} - m\tau_{\mathrm{SS}}}{\tau_{\mathrm{SS}}} \right). \end{split}$$

Given (7), E(A) is computed as in (6) and TCP goodput is given by (1).

2.3.2. Approximation using Wald's equation It is also possible to derive an accurate approximation for $E(A \mid T')$ by applying Wald's equation (see [25]). The benefit of the approximation is that it is more explicit than the above exact formula. Wald's equation states that, given a sequence of i.i.d. distributed random variables Y_1, Y_2, \ldots and a stopping time N for the Y_n , then

$$E\left(\sum_{n=1}^{N} Y_n\right) = E(N)E(Y).$$

In our case, Y_n is defined as

$$Y_n = \left\{ \begin{array}{ll} \tau_{\rm SS}, & \text{with probability } \beta, \\ \tau_{\rm CA}, & \text{with probability } (1-\beta), \end{array} \right.$$

and

$$E(Y) = \beta \tau_{SS} + (1 - \beta) \tau_{CA}$$
.

The stopping time N can be defined as the time when T' is exceeded upon completion of the Nth cycle. Using Wald's equation, $E(A \mid T')$ can be bounded by first noting that

$$(1/p) E(N-1) < E(A | T') < (1/p) E(N),$$

from which it follows

$$(1/p)\,\frac{E\left(\sum_{n=1}^{N-1}Y_n\right)}{E(Y)} < E(A\,|\,T') < (1/p)\,\frac{E\left(\sum_{n=1}^{N}Y_n\right)}{E(Y)}.$$

Copyright © 0000 AEIT Prepared using ettauth.cls Because $T' - \tau_{SS} \leq E\left(\sum_{n=1}^{N-1} Y_n\right)$ and additionally $T' + \tau_{SS} \geq E\left(\sum_{n=1}^{N} Y_n\right)$, the above can be further bounded

$$(1/p) \cdot \frac{T' - \tau_{SS}}{E(Y)} < E(A \mid T') < (1/p) \cdot \frac{T' + \tau_{SS}}{E(Y)}.$$
 (8)

Numerically we have observed that a very good approximation of $E(A \mid T')$ is given by taking the average of the upper and lower bounds in (8),

$$E(A \mid T') \approx (1/p) \cdot \frac{T'}{E(Y)}.$$
 (9)

Note that the above approximation T'/E(Y) for the mean number of cycles, can be also interpreted as following from assuming that we have Poisson arrivals with rate 1/E(Y). Thus, by (6) and (1), TCP goodput is given by

$$G_{\text{TCP}} = \frac{1}{E(T)} \int_{SD}^{\infty} E(A \mid t - SD) f_T(t) dt$$

$$\approx \frac{1}{p E(T) E(Y)} \int_{SD}^{\infty} (t - SD) f_T(t) dt. \quad (10)$$

Example with exponential distribution: If we assume that the RTT spikes obey an exponential distribution with mean 1/E(T), $\int_{SD}^{\infty} (t-SD) f_T(t) dt = E(T) \cdot e^{-SD/E(T)}$ and (10) simplifies to

$$G_{\rm TCP} \approx \frac{e^{-SD/E(T)}}{p \cdot E(Y)}.$$

Note that above E(Y) depends on R, p and C_{max} .

Accuracy of the approximation: Next we illustrate the accuracy of (9) compared to the exact solution (7) when computing the conditional reward $E(A \mid T')$. To this end let us here denote by a the value of $E(A \mid T')$ as given by (7) and by \hat{a} the approximate value of $E(A \mid T')$ as given by (9). We are interested in the relative error of the approximation (9), given by $(a - \hat{a})/a$, as a function of T. The relevant dependencies in the error are that for small T the error is bigger (especially if $T < \tau_{SS}$) and for large T the error is close to zero. For a given value of Tthe bigger the difference in the value of τ_{SS} and τ_{CA} the bigger the error. Also, the value of β affects the accuracy. Note that the error does not decrease monotonously for increasing T. These issues are illustrated in Figure 4, where we plot the relative error as a function of T with $\tau_{SS} = 1$ for $\beta = 0.1$ (left figure) and $\beta = 0.5$ (right figure) with $\tau_{\rm CA}=0.1$ or $\tau_{\rm CA}=0.5$. Due to the accuracy and simplicity of (9), in the numerical results of Section 4, we have used the approximation (9) in the computations.

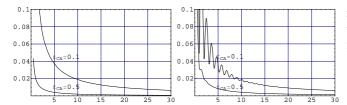


Figure 4: The relative error of (9) compared with (7) as a function of T for $\beta=0.1$ (left) and $\beta=0.5$ (right) with $\tau_{\rm SS}=1$ and $\tau_{\rm CA}=0.1$ or $\tau_{\rm CA}=0.5$.

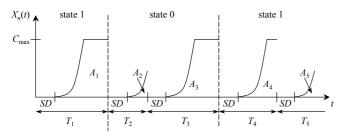


Figure 5: Illustration of the Markov renewal model in the lossless case with slow start.

3. Extensions to the model

3.1. Non-i.i.d RTT spike intervals

In the previous model the intervals between RTT spikes, T_n s, were required to be i.i.d. Next we show how we can introduce correlation between T_n s by considering the time-averages of semi-regenerative processes. In such processes the cycles are no longer i.i.d, but periods are "modulated" by an embedded Markov chain B_n , whose invariant distribution is denoted by π . We assume that the time scale of the modulating process is slower than the RTT spike process.

To illustrate the model in a simple setting we assume that the modulating discrete time Markov process has two states 0 and 1, with transition probabilities p_{00} , p_{01} , p_{10} and p_{11} , and hence the equilibrium probabilities are simply $\pi_0 = p_{10}/(p_{01}+p_{10})$ and $\pi_1 = p_{01}/(p_{01}+p_{10})$. Furthermore, the possible state transition times in the modulating process are only the times when an RTT spike occurs. Thus, the state transition times of the embedded process are complete regeneration times of the joint process. This is illustrated for the lossless situation with slow start and rate limitation in Figure 5.

In this case the sequence $(B_n,(A_n,T_n))$ is a Markov marked chain and (B_n,A_n,T_n) is called a Markov renewal sequence. Then if $E_\pi(A)<\infty$ and $E_\pi(T)<\infty$, the

Copyright © 0000 AEIT Prepared using ettauth.cls renewal reward theorem can be applied by averaging the reward also over the steady state distribution of the modulating Markov process [25],

$$G_{\text{TCP}} = \frac{E_{\pi}(A)}{E_{\pi}(T)}.$$

3.1.1. Model without congestion losses If the RTT spike intervals have a probability density function $f_0(t)$ at state 0 and $f_1(t)$ at state 1, the goodput formula becomes

$$G_{\text{TCP}} = \frac{E_{\pi} \left(\int_{0}^{T} X(u) du \right)}{E_{\pi}(T)}$$

$$= \frac{\pi_{0} \int_{0}^{\infty} f_{0}(t) \int_{0}^{t} X(u) du \, dt}{\pi_{0} \int_{0}^{\infty} f_{0}(t) t dt + \pi_{1} \int_{0}^{\infty} f_{1}(t) t dt}$$

$$+ \frac{\pi_{1} \int_{0}^{\infty} f_{1}(t) \int_{0}^{t} X(u) du \, dt}{\pi_{0} \int_{0}^{\infty} f_{0}(t) t dt + \pi_{1} \int_{0}^{\infty} f_{1}(t) t dt}.$$

Example with exponential distribution: If we assume exponential probability densities $f_0(t) = 1/\mu_0 e^{-t/\mu_0}$ and $f_1(t) = 1/\mu_1 e^{-t/\mu_1}$, the goodput formula simplifies to

$$G_{\text{TCP}} = \frac{\pi_0 e^{-SD/\mu_0} \mu_0 \frac{R - \mu_0 \ln 2C_{\text{max}}^{1 - \frac{R}{\mu_0 \ln 2}}}{R - \mu_0 \ln 2}}{\pi_0 \mu_0 + \pi_1 \mu_1} + \frac{\pi_1 e^{-SD/\mu_1} \mu_1 \frac{R - \mu_1 \ln 2C_{\text{max}}^{1 - \frac{R}{\mu_1 \ln 2}}}{R - \mu_1 \ln 2}}{\pi_0 \mu_0 + \pi_1 \mu_1}.$$
 (11)

3.1.2. Model with congestion losses Using the above notation we briefly state how (10) translates into the Markov modulated renewal setting. For general RTT spike distributions we get

$$G_{\text{TCP}} = \frac{\pi_0 \int_{\text{SD}}^{\infty} E(A \mid t - SD) f_0(t) dt}{\pi_0 \int_0^{\infty} f_0(t) t dt + \pi_1 \int_0^{\infty} f_1(t) t dt} + \frac{\pi_1 \int_{\text{SD}}^{\infty} E(A \mid t - SD) f_1(t) dt}{\pi_0 \int_0^{\infty} f_0(t) t dt + \pi_1 \int_0^{\infty} f_1(t) t dt} \approx \frac{\pi_0 \int_{\text{SD}}^{\infty} (t - \text{SD}) f_0(t) dt}{p E(Y) \left(\pi_0 \int_0^{\infty} f_0(t) t dt + \pi_1 \int_0^{\infty} f_1(t) t dt\right)} + \frac{\pi_1 \int_{\text{SD}}^{\infty} (t - \text{SD}) f_1(t) dt}{p E(Y) \left(\pi_0 \int_0^{\infty} f_0(t) t dt + \pi_1 \int_0^{\infty} f_1(t) t dt\right)}.$$
(12)

Example with exponential distribution: Again, if we assume exponential probability densities $f_0(t)$ and $f_1(t)$, as above, we get

$$G_{\text{TCP}} \approx \frac{\pi_0 \mu_0 e^{-\text{SD}/\mu_0} + \pi_1 \mu_1 e^{-\text{SD}/\mu_1}}{(\pi_0 \mu_0 + \pi_1 \mu_1) \, p \, E(Y)}.$$
 (13)

3.2. Including the distribution of delay spike lengths

Thus far we have assumed that the length of the RTT spikes is constant, denoted by SD. As mentioned before, SD can be interpreted as modeling the mean length of the RTT spikes. The model can, however, be easily modified to allow a distribution for the RTT spikes. To this end, consider the case where the length of the RTT spikes are random variables denoted by SD and have a common distribution with density $f_{SD}(\cdot)$. Thus, associated with the nth renewal period Z_n we have SD_n . We assume that the value of SD_n is independent of Z_n . If $T_{n+1} = Z_{n+1} - Z_n < SD_n$, i.e., the next renewal epoch occurs before the nth spike is over, the reward from such realizations equals zero. With this, (6) can be expressed as

$$E(A) = \int_0^\infty f_{SD}(\tau) \int_{\tau}^\infty E(A \mid t - \tau) f_T(t) dt d\tau.$$

4. Numerical results

4.1. Validation of the models

To validate our models we have used the ns2 simulator equipped with the RTT spike generator contributed by Andrei Gurtov*. The simulation scenario that is used to estimate the goodput consists of transmitting TCP data over a single link having a constant service rate, but which can be turned on/off for random periods of time (to generate the RTT spikes). The TCP source type used in the simulations is TCP SACK. To estimate the goodput, we simply send a large file $(3 \cdot 10^6 \text{ packets})$ over the link and measure the time it took for TCP to send the given number of packets. To enforce the sending rate limit on TCP we have set the maximum window to be equal to the bandwidth delay product of the simulated system.

The accuracy of our model is evaluated for a small and larger bandwidth delay product scenario. To this end, we fix the packet size at 576 bytes and link one-way propagation delay equals 100 ms and we experiment with a low bandwidth link with rate 40 kbps and a higher bandwidth link with rate 100 kbps. These link speeds are comparable to those achieved with present day EDGE/GPRS systems. In the models, we also take here R to include the packet transmission time (in addition to the propagation delay), i.e., R=2·link propagation delay + packet transmission time ≈ 315 ms. This represents the average value of the RTT when RTT spikes are excluded,

and it is comparable with measurement results on RTTs in GPRS systems, see, e.g., [9, 14, 11, 7, 13]. Additionally, we vary the length of the RTT spikes (long/medium/short spikes) and also study the impact of congestion losses.

First the case of long RTT spikes is considered; in the simulations the spike durations were uniformly distributed in the range [5,10] s, i.e., SD=7.5 s in our models. The goodput is evaluated as a function of the mean time between RTT spikes, E(T). In the simulations for each E(T), the spikes are uniformly distributed in the range $[10,2\cdot E(T)-10]$ s. This range of delays is reasonable when the spikes are triggered, e.g., by hand overs and dynamic resource allocation at the link layer, see [10]. The results are in Figure 6 for three different loss scenarios: 0% loss (square), 2% loss (diamond), and 5% loss (triangle). The simulation results are always shown in dashed lines and the analytical results from the models in solid lines.

We see that in the low bandwidth lossless case the model and the simulations match exactly. In the low bandwidth case when packet losses are present, the model typically overestimates the goodput, but the agreement is very good. When the bandwidth is increased to 100 kbps, the lossless model overestimates the goodput, which is natural as we allow exponential increase of the bandwidth up to the bandwidth limitation (whereas in reality TCP switches to a linear increase after slowstart threshold is reached). In the model with packet losses the congestion avoidance behavior is incorporated in the model. There we overestimate the goodput for low packet loss and underestimate it for higher packet loss. The accuracy up to 2% packet losses is very good.

In the above, the spike process produces rather long spikes (with a mean $SD=7.5\,\mathrm{s}$), whereas the typical RTT equals roughly 300 ms. Then it can be anticipated that the spikes are indeed dramatic enough such that TCP's RTT estimator mechanism can not learn the properties of the spike process. Hence, our assumption that every spike results in a timeout and slow start is mostly valid. However, if the duration of the spikes is shortened, this may not necessarily be the case anymore. To check this, same simulations as earlier (with low/high bandwidth) were performed where the spike durations were generated from a uniform distribution in the range [1,5] (with a mean $SD=3\,\mathrm{s}$) and in the range [1,2] (with mean $SD=1.5\,\mathrm{s}$). The results are shown in Figure 7 and Figure 8.

Similar observations as made for $SD=7.5 \, \mathrm{s}$ in Figure 6 apply to the case here when $SD=3 \, \mathrm{s}$ in Figure 7, i.e., the spikes are long enough so that TCP's timer mechanism can not adapt to the spikes. However, the results for the case where $SD=1.5 \, \mathrm{s}$ reveal that now

^{*}Patches are available from http://www.cs.helsinki.fi/u/qurtov/ns/

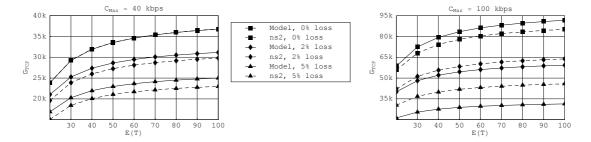


Figure 6: Comparison of model (solid lines) and simulations (dashed lines) in low (left) and higher (right) bandwidth delay product scenarios when delay spikes have long durations (SD = 7.5 s).

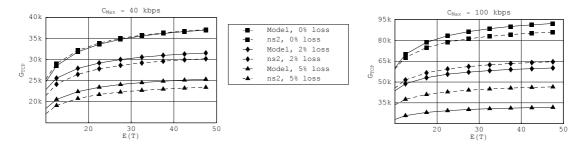


Figure 7: Comparison of model (solid lines) and simulations (dashed lines) in low (left) and higher (right) bandwidth delay product scenarios when delay spikes have somewhat shorter durations (SD = 3 s).

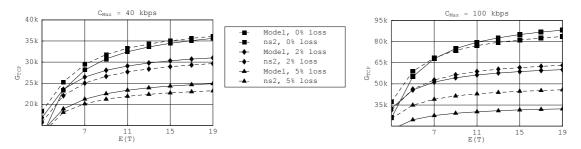


Figure 8: Comparison of model (solid lines) and simulations (dashed lines) in low (left) and higher (right) bandwidth delay product scenarios when delay spikes have short durations (SD = 1.5 s).

TCP's timer mechanism has indeed adapted to the spikes. This can be seen in Figure 8 especially clearly for the low bandwidth case for the lossless model (the topmost pair of lines). In earlier scenarios our model predicted the performance very accurately. Now the model underestimates the performance and this underestimation is due to that now in the simulation TCP does not have a slow start (causing loss of goodput) after every RTT spike (as is assumed in the model). This is especially clear when the spikes occur frequently with E(T) < 7. There the impact of the assumed slowstart after each RTT spike causes the goodput to drop quickly and the predicted

performance even becomes worse than for scenarios where we have congestion losses (remember that in the model with congestion losses there is no slowstart after an RTT spike). In fact, a better approximation in this case would be given by the model with congestion losses where loss probability p is given a value that is numerically close to zero (say, $p=10^{-6}$). On the other hand, for the lossy cases our model behaves again as earlier, i.e., the model slightly overestimates the performance. For high bandwidth scenario without packet losses we either overestimate or underestimate the goodput depending on the mean time between the RTT spikes. When packet

losses are present, the model underestimates the goodput, especially for the 5% loss case.

4.2. Sensitivity of performance to RTT spike distribution

Now we illustrate in Figure 9 the effect of the RTT spike distribution on the TCP goodput in the lossless model (left figure) and the model with packet losses (right figure). The x-axis is the mean of the given distribution and the y-axis is the corresponding estimate for the TCP goodput in bps obtained from the exact result for the lossless case or the approximation (10) for the case with packet losses.

In the Pareto distribution the shape parameter equals 1.5; the uniform distribution is located at the interval $[E(T) - \min(E(T), 5), E(T) + \min(E(T), 5)]$. Markov modulating process is illustrated by using exponential distributions with $\pi_0 = \pi_1 = 1/2$ and $\mu_0 = 0.25E(T)$ and $\mu_1 = 1.75E(T)$, i.e., in state 0 RTT spikes are more frequent than in state 1. The curves are calculated from (11) and (13). The RTT spike duration is SD = 5 and the other parameters are: maximum rate 40 kbps, packet size 576 bytes, propagation delay 0.6 sec, which give rise to R = 720 ms and $C_{\rm max} = 8.68$ pkts/s. In the model with packet losses the loss probability p = 0.02 is illustrated.

For the TCP goodput the worst case of RTT spikes in both models is the uniform distribution. This is because n bursty spikes yield a better goodput than n uniformly distributed spikes on a time interval of the same length. The benefit of bursts can be seen by comparing the Pareto and exponential distributions. Moreover, the exponential distribution and the Markov modulated exponential distribution have the same mean, but in the Markov modulated version spikes arrive frequently in state 0 and very seldom in state 1. Thus the modulated RTT process yields throughout a higher goodput. The impact of correlation in RTT spikes on the goodput is in agreement with the result on the TCP AIMD-component analyzed in [3].

In the model with congestion losses the packet loss probability in the range of 1-10% does not change appreciably how the different RTT spike distributions compare to each other. Only the general level of the goodput is lowered when the packet loss probability increases, as p is a scale factor in formula (10).

5. Conclusions

On a wireless link, the observed RTTs of TCP can be highly variable and the pattern of variability may contain sharp spikes, called delay spikes. These result in spurious timeouts that lower the TCP performance. We have provided a facile modeling framework to study the impact of such RTT spikes on the goodput of a TCP source.

Delay spikes, or RTT spikes, are triggered by random events occurring in the mobile's environment. Hence we have modeled the delay spikes as an outside disturbance. First, the case of i.i.d. spike intervals was considered in the framework of renewal reward theory and we derived an expression for the TCP goodput. In the modeling, we have first considered a lossless low bandwidth link on which the rate of successfully sent TCP packets is described by a) a silence for the mean duration of the delay spike, b) exponential increase (the slow start phase) and c) reaching the maximum rate due to bandwidth (window) limitation. Then a model has been presented that includes also congestion related packet losses by approximately taking into account the usual sawtooth pattern in the TCP sending rate, possible slow starts and congestion related timeouts between the RTT spikes, as well as a limited sending rate. Correlation between RTT spike intervals has been incorporated using a modulating embedded discrete time Markov chain and Markov renewal reward results.

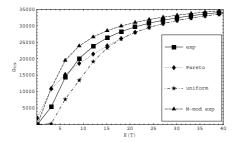
Validation with ns2 showed that our models closely approximate the TCP goodput in the presence of RTT spikes. In the lossless modeling scenario with a low bandwidth link the agreement with simulation is excellent. For a higher bandwidth link our model slightly overestimates the lossless goodput (as we allow exponential increase up to the bandwidth limitation). For the cases with congestion losses, the accuracy of the model is still very good for the low bandwidth case, but for the higher bandwidth case accuracy is still good up to 5% losses but with even higher loss rates our model is underestimating the performance somewhat. Finally, by studying the sensitivity of the goodput on the RTT spike interval distribution, we observed that in both models bursty spikes give higher goodput. This is similar to the result in [3] that studied the congestion control component of the TCP.

Currently the model is for a greedy TCP source, i.e., we assume that the TCP source has always data to send. One interesting direction for future research is to include a session structure in the model with thinking times and finite file sizes with random lengths.

Acknowledgement

We are grateful to Dr. Samuli Aalto and Dr. Esa Hyytiä for helpful discussions and comments.

Copyright © 0000 AEIT Prepared using ettauth.cls



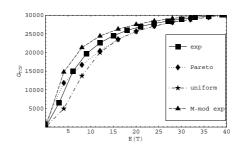


Figure 9: The impact of RTT spike interval distribution on TCP goodput for the lossless model (left) and model with congestion losses (right).

REFERENCES

- A. A. Abouzeid and S. Roy. Stochastic modeling of TCP in networks with abrupt delay variations. ACM/Kluwer Wireless Networks, 9(5), 2003.
- A. A. Abouzeid, S. Roy, and M. Azizoglu. Stochastic modeling of TCP over lossy links. In *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000.
- E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. In *Proceedings of ACM SIGCOMM*, Stockholm, Sweden, August 2000.
- U. Ayesta, K. Avrachenkov, E. Altman, C. Barakat, and P. Dube. Multilevel approach for modeling short TCP sessions. In Proceedings of 18th International Teletraffic Congress (ITC 18), Berlin, Germany, August 2003.
- F. Baccelli and D. R. McDonald. A stochastic model for the throughput of non-persistent TCP flows. In *Proceedings of Value Tools* 2006, Pisa, Italy, October 2006.
- F. Baccelli, D. R. McDonald, and J. Reynier. A mean-field model for multiple TCP connections through a buffer implementing RED. *Performance Evaluation*, 49, 2002.
- P. Benko, G. Malicsko, and A. Veres. A large-scale, passive analysis
 of end-to-end TCP performance over GPRS. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.
- S. Fu and M. Atiquzzaman. Modeling TCP reno with spurious timeouts in wireless mobile environment. In *Proceedings of* 12th International Conference on Computer Communications and Networks, Dallas, Texas, USA, October 2003.
- A. Gurtov. Effect of delays on TCP performance. In *Proceedings* of *IFIP Personal Wireless Communications*, Lappeenranta, Finland, August 2001.
- A. Gurtov and S. Floyd. Modeling wireless links for transport protocols. ACM Computer Communications Review, 34(2), 2004.
- A. Gurtov, M. Passoja, O. Aalto, and M. Raitola. Multilayer protocol tracing in a GPRS network. In *Proceedings of the IEEE Vehicular Technology Conference*, Vancouver, BC, Canada, September 2002.
- I. Kaj and J. Olsén. Throughput modeling and simulation for single connection TCP-tahoe. In *Proceedings of 17th International Teletraffic Congress (ITC 17)*, December 2001.
- J. Kilpi and P. Lassila. Micro- and macroscopic analysis of RTT variability in GPRS and UMTS networks. In *Proceedings of Networking* 2006, Coimbra, Portugal, May 2006.
- J. Korhonen, O. Aalto, A. Gurtov, and H. Laamanen. Measured performance of GSM HSCSD and GPRS. In *Proceedings of the IEEE Conference on Communications*, St. Petersburg, Russia, June 2001.

- A. Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4), 1998.
- P. Kuusela and P. Lassila. Modeling the impact of delay spikes on TCP performance on a low bandwidth GPRS wireless link. In Proceedings of 16th ITC Specialist Seminar, Antwerpen, Belgium, September 2004.
- R. Ludwig and R. H. Katz. The eifel algorithm: making TCP robust against spurious retransmissions. ACM Computer Communications Review, 30(1), 2000.
- M. A. Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello. Using partial differential equations to model TCP mice and elephants in large IP networks. *IEEE Transactions on Networking*, 13(6), 2005.
- M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. ACM Computer Communications Review, 27(3), 1997.
- S. Misra, W.B. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proceedings of ACM SIGCOMM*, August 2000.
- J. Olsén and I. Kaj. Slowstart window modeling for single connection TCP-tahoe. Technical Report, University of Uppsala, Sweden, Report 2001:21, 2001.
- J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. In Proceedings of ACM SIGCOMM, September 1998.
- J. Roberts. A survey on statistical bandwidth sharing. Computer Networks, 45, 2004.
- P. Sarolahti, M. Kojo, and K. Raatikainen. F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts. ACM Computer Communications Review, 33(2), 2003.
- R. Wolff. Stochastic Modeling and the Theory of Queues. Prentice-Hall. 1989.
- M. Zorzi, A. Chockalingam, and R. R. Rao. Throughput analysis of TCP on channels with memory. *IEEE Journal on Selected Areas in Communications*, 18(7), 2000.

Appendix: Derivation of τ_{CA} , τ_{SS} and β

In the pure sawtooth case, the rate evolution during $\tau_{\rm CA}$ is determined by a linear increase from $x_0^{(1)}$ to $2x_0^{(1)}$ with a slope 1/R (see Figure 10), i.e., rate is halved upon each loss. Thus, we have that $2x_0^{(1)} = x_0^{(1)} + \tau_{\rm CA}/R$ and the area under the curve should equal 1/p. In the cutted

Copyright © 0000 AEIT Prepared using ettauth.cls

sawtooth case (see Figure 10), the rate increases linearly from $x_0^{(2)}$ to $C_{\rm max}$ after which the congestion window still increases linearly (the dashed line) but packets are sent only at the rate $C_{\rm max}$. We again have the relations that $2x_0^{(2)} = x_0^{(2)} + \tau_{\rm CA}/R$ and the area under the curve equals 1/p (number of sent packets), and $\tau_{\rm CA}$ can be deduced from these. In the rate limited case, $\tau_{\rm CA}$ simply equals the time it takes to send 1/p packets with rate $C_{\rm max}$. After some algebraic manipulations, we arrive at

$$\tau_{\text{CA}} = \begin{cases} \tau_{\text{CA}}^{(1)} = \sqrt{\frac{2R}{3p}}, \\ \text{if } 0 \le x_0^{(1)} < \frac{C_{\text{max}}}{2}, \end{cases}$$

$$\tau_{\text{CA}} = \begin{cases} \tau_{\text{CA}}^{(2)} = R\left(2C_{\text{max}} - \sqrt{3C_{\text{max}}^2 - \frac{2}{pR}}\right), & \text{(14)} \end{cases}$$

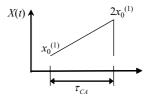
$$\text{if } \frac{C_{\text{max}}}{2} \le x_0^{(2)} \le C_{\text{max}}$$

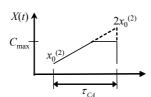
$$\tau_{\text{CA}}^{(3)} = \frac{1/p}{C_{\text{max}}}, & \text{otherwise},$$

where

$$\begin{cases} x_0^{(1)} = \sqrt{2R/(3p)}, \\ x_0^{(2)} = 2C_{\text{max}} - \sqrt{3C_{\text{max}}^2 - 2/(pR)}. \end{cases}$$
 (15)

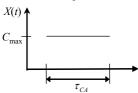
Note that when $x_0^{(1)}=C_{\rm max}/2$ then $x_0^{(2)}=C_{\rm max}/2$, i.e., the limits in (14) are continuous.





Mode 1: pure sawtooth

Mode 2: cutted sawtooth



Mode 3: rate limited

Figure 10: Illustration and notation for different τ_{CA} cycles.

As discussed earlier, associated with each $\tau_{\rm CA}$ is $\tau_{\rm SS}$, the time it takes to send 1/p packets assuming a constant silence period, TO, a slow start and, after that, the same rate evolution as in the corresponding $\tau_{\rm CA}$ cycle. Let

us consider fist the case that $\tau_{\rm CA}=\tau_{\rm CA}^{(1)}$, i.e., the pure sawtooth case. The corresponding timeout-slow start cycle is determined by a constant silence period of length TO, an exponential rate increase obeying $x(t)=2^{t/R}$ up to $x_0^{(1)}$ as given by (15), a linear increase with slope 1/R until x' at which point 1/p packets have been sent (see Figure 11). In this case, $\tau_{\rm SS}=TO+t_1+t_2$, where t_1 is the length of the slow start period and t_2 the length of the linear increase period. The time $t_1=R\log_2 x_0^{(1)}$ and during this time $a_1=R(x_0^{(1)}-1)/\ln 2$ packets are sent. During t_2 , $a_2=t_2x_0^{(1)}+t_2^2/(2R)$ packets are sent. During t_1+t_2 altogether 1/p packets are sent, which results in a second order equation for t_2 , whose only positive solution equals

$$t_2 = -R \left(x_0^{(1)} - \sqrt{\left(x_0^{(1)}\right)^2 - \frac{2(x_0^{(1)} - 1)}{\ln 2} - \frac{2}{pR}} \right)$$

If $a_1 \ge 1/p$, then the cycle ends during slow start and $t_1 = R \log_2(\ln 2/(pR) + 1)$. Thus, $\tau_{\text{SS}}^{(1)}$ is given by

$$\tau_{\text{SS}}^{(1)} = \begin{cases} TO + R \left(\log_2 x_0^{(1)} - x_0^{(1)} + \sqrt{\left(x_0^{(1)}\right)^2 - \frac{2(x_0^{(1)} - 1)}{\ln 2} - \frac{2}{pR}} \right), \\ \text{if } \frac{R(x_0^{(1)} - 1)}{\ln 2} \le \frac{1}{p}, \end{cases}$$

$$TO + R \log_2 \left(\frac{\ln 2}{pR} + 1 \right),$$
otherwise.

Next consider the cutted sawtooth case. The slow start cycle is determined by a silence of length TO, an exponential increase up to $x_0^{(2)}$ as given by (15), a linear increase with slope 1/R until $C_{\rm max}$ and after that a constant rate until 1/p packets have been sent (see Figure 11). As earlier, the length of the slow start phase is $t_1 = R\log_2 x_0^{(2)}$ and during this time $a_1 = R(x_0^{(2)} - 1)/\ln 2$ packets are sent. The length of the linear increase part is $t_2 = R(C_{\rm max} - x_0^{(2)})$ and during this time $a_2 = R(C_{\rm max}^2 - (x_0^{(2)})^2)/2$ packets are sent. Thus, if $a_1 + a_2 < 1/p$, the length of the constant rate period is given by $t_3 = (1/p - a_1 - a_2)/C_{\rm max}$, and the length of the cycle equals $TO + t_1 + t_2 + t_3$. If $a_1 < 1/p \le a_1 + a_2$, the cycle stops during the linear increase part. Thus, the total time equals the same as in the pure sawtooth case, as given in (16). If $a_1 \ge 1/p$, the cycle ends during slow

Copyright © 0000 AEIT Prepared using ettauth.cls

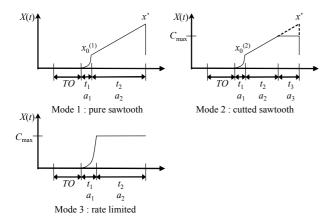


Figure 11: Illustration and notation for different τ_{SS} cycles.

Table 1: Summary of τ_{CA} , τ_{SS} and β .

be expressed as

, CIT/ 88 1			
TCP mode	$ au_{\mathrm{CA}}$	$ au_{ m SS}$	β
$ \begin{array}{c c} 0 \leq x_0^{(1)} < \frac{C_{\max}}{2} \\ \frac{C_{\max}}{2} \leq x_0^{(2)} < C_{\max} \\ \text{otherwise} \end{array} $	$ \begin{array}{c} \tau_{\text{CA}}^{(1)} \text{ in (14)} \\ \tau_{\text{CA}}^{(2)} \text{ in (14)} \\ \tau_{\text{CA}}^{(3)} \text{ in (14)} \end{array} $	$ \begin{array}{c} \tau_{\rm SS}^{(1)} \ {\rm in} \ (16) \\ \tau_{\rm SS}^{(2)} \ {\rm in} \ (17) \\ \tau_{\rm SS}^{(3)} \ {\rm in} \ (18) \\ \end{array} $	$\beta(2x_0^{(1)}R, p)$ $\beta(C_{\max}R, p)$ $\beta(C_{\max}R, p)$

start, which was also already considered earlier in the pure sawtooth case. Thus, $\tau_{SS}^{(2)}$ can be expressed as

$$\tau_{\text{SS}}^{(2)} = \begin{cases} TO + R \log_2 x_0^{(2)} + R(C_{\text{max}} - x_0^{(2)}) + \\ \frac{1}{pC_{\text{max}}} - \frac{R(x_0^{(2)} - 1)}{C_{\text{max}} \ln 2} - \frac{RC_{\text{max}}}{2} + \frac{R(x_0^{(2)})^2}{2C_{\text{max}}}, \\ \text{if } \frac{1}{p} > \frac{R(x_0^{(2)} - 1)}{\ln 2} + \frac{R}{2} \left(C_{\text{max}}^2 - \left(x_0^{(2)} \right)^2 \right), \\ TO + R \left(\log_2 x_0^{(2)} - x_0^{(2)} + \\ \sqrt{\left(x_0^{(2)} \right)^2 - \frac{2(x_0^{(2)} - 1)}{\ln 2} - \frac{2}{pR}} \right), \\ \text{if } \frac{R(x_0^{(2)} - 1)}{\ln 2} < \frac{1}{p} \leq \frac{R(x_0^{(2)} - 1)}{\ln 2} + \\ \frac{R}{2} \left(C_{\text{max}}^2 - \left(x_0^{(2)} \right)^2 \right), \\ TO + R \log_2 \left(\frac{\ln 2}{pR} + 1 \right), \\ \text{otherwise.} \end{cases}$$

$$To fix the value of β , the probability of a slow state cycle, we approximate the window size w needed in the probability of the window size at the end of the corresponding π of the probability of the probability of the window size at the end of the corresponding π of the probability of the window size at the end of the corresponding π of the probability of the window size is obtained by the window size at the end of the corresponding π of the probability of the window size is obtained by the window size at the end of the corresponding π of the probability of the probability of the window size at the end of the corresponding π of the probability of the window size at the end of the corresponding π of the probability of the probability$$

Finally, we consider the rate limited case. The slow start cycle is now determined by an exponential increase up to $C_{\rm max}$, after which packets are sent at a constant rate $C_{\rm max}$ until 1/p packets have been sent (see Figure 11). The length of the slow start phase is $t_1 = R \log_2 C_{\text{max}}$ and during this time $a_1 = R(C_{\rm max} - 1)/\ln 2$ packets are sent. The remaining packets $a_2 = (1/p - a_1)$ are sent at the rate $C_{\rm max}$, and $t_2 = a_2/C_{\rm max}$. If $1/p < a_1$, the cycle ends during slow start (as discussed earlier). Thus, $\tau_{SS}^{(3)}$ can

Copyright © 0000 AEIT Prepared using ettauth.cls

$$\tau_{\text{SS}}^{(3)} = \begin{cases} TO + R \log_2 C_{\text{max}} + \frac{1}{pC_{\text{max}}} - \frac{R(C_{\text{max}} - 1)}{C_{\text{max}} \ln 2}, \\ \text{if } \frac{R(C_{\text{max}} - 1)}{\ln 2} \leq \frac{1}{p}, \\ TO + R \log_2 \left(\frac{\ln 2}{pR} + 1\right), \\ \text{otherwise.} \end{cases}$$
(18)

To fix the value of β , the probability of a slow start cycle, we approximate the window size w needed in (5) by the window size at the end of the corresponding $\tau_{\rm CA}$ cycle. Given a rate r, the window size is obtained by the familiar relation w = rR. Thus, in the pure sawtooth case we have that $\beta \approx \beta(2x_0^{(1)}R, p)$. For the cutted sawtooth case, $\beta \approx \beta(C_{\text{max}}R, p)$, and similarly for the rate limited case $\beta \approx \beta(C_{\text{max}}R, p)$.

In Table 1 we have summarized the results for obtaining $\tau_{\rm CA}$, $\tau_{\rm SS}$ and β . Finally, note that the actual condition for the "otherwise" in the table is that the discriminant of $x_0^{(2)}$ in (17) becomes negative, i.e., $3C_{\text{max}}^2 - 2/(pR) < 0$.