Department of Computer Science
Series of Publications C
Report C-2002-13

Analytical Markovian Model of TCP Congestion Avoidance Algorithm Performance

Olga Bogoiavlenskaia, Markku Kojo,

Matt Mutka, and Timo Alanko

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O.Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 44441

Computing Reviews (1998) Classification: C.2.1, C.2.2
Helsinki 2002

Department of Computer Science
Series of Publications C
Report C-2002-13

Analytical Markovian Model of TCP Congestion Avoidance Algorithm Performance

Olga Bogoiavlenskaia, Markku Kojo,
Matt Mutka, and Timo Alanko

University of Helsinki
Finland

Analytical Markovian Model of TCP Congestion Avoidance Algorithm Performance

Olga Bogoiavlenskaia, Markku Kojo,
Matt Mutka, and Timo Alanko

Department of Computer Science
P.O. Box 26, FIN-00014 University of Helsinki, Finland

Technical report, Series of Publications C, Report C-2002-13
Helsinki, April 2002, ii + 26 + 11 pages

Abstract

We consider the performance of the TCP congestion avoidance algorithm by developing the Markovian model of the flow control algorithm, oftenly refered as Additive Increase Multiplicative Decrease. Analytically, we evaluate the steady state distributions of the congestion window size and the throughput. Further processing of the distributions yields moment of any degree i.e. expectation, variance, quantiles and other important performance metrics. Our model covers a wide range of networking environments. It relaxes several important restrictions presently accepted in analytical studies of TCP. Besides strict application in design and engineering, the model states usability bounds for simple estimations of the average TCP throughput. The paper includes a set of numerical examples that illustrate the scope of the model and state important properties of congestion avoidance. Our results are validated by experimental analysis. The values generated by the model are compared to that demonstrated by TCP connections in an emulated network environment and by other published models as well. The model demonstrates high stability and good agreement with the experiments.

Computing Reviews (1998) Categories and Subject Descriptors:
C.2.1 Computer-communication networks: Network Architectures and Design
C.2.2 Computer-communication networks: Network Protocols

General Terms:
Network protocols, Performance evaluation

Additional Key Words and Phrases:
Congestion avoidance, Markovian model, performance, TCP, throughput

Contents

Chapter 1

Introduction

The majority of flows, packets and bytes that travel over the Internet are controlled by the TCP protocol at the transport layer. Significant efforts have been directed towards developing models for analytical evaluation of TCP throughput. Analytical models enable the analysis of the performance behavior of existing protocols and the design of new protocols. An analytical model binds network parameters with protocol performance metrics to improve the understanding of the protocol sensitivities. Furthermore, an analytical model allows one to predict the bounds of the protocol performance on one environment and compare it to another environment. Quality of Service (QoS) prediction and management may be enhanced by using analytical models, such that the models become powerful tools for planning and traffic engineering. Regardless of the area of application of analytical modeling, there is a potential to save significant amounts of time, resources and effort in comparison to simulation tools.

We investigate the throughput of TCP congestion avoidance, which is often classified as an Additive Increase Multiplicative Decrease (AIMD) algorithm. The development of a throughput model for AIMD is important since a stable TCP connection spends most of its lifetime within the AIMD phase. Our evaluation comes from a sender's point of view to understand the rate that a TCP sender injects segments to the network. The model does not consider short TCP connections.

As a result of our investigation, we developed a Markovian model of TCP congestion window size and congestion avoidance throughput. Our new model yields several important new results concerning TCP congestion avoidance behavior:

- We obtained the steady state distributions of the AIMD congestion window size, cwnd, and congestion avoidance throughput (T_{CA}).

- These distributions allow us to calculate not only the average throughput of a connection, as has been done in some recent studies [AAB00, BH00, Flo99, MSMO97, PFTK01], but also its variance, moments of higher degree, quantiles, and interval estimations. This provides complete information concerning congestion avoidance behavior in a particular networking environment.

- Our model establishes quantitative dependence between the protocol performance metrics and the characteristics of the networking environment in which the protocol is implemented (e.g., segment loss probability and round trip time (RTT) distribution).
- Several studies have obtained estimations of TCP average throughput or its upper or lower bounds. All have restrictions on their applicability, which are not determined exactly since the bounds will vary depending on the properties of the networking environment. In addition to the derivation of the expectation of congestion avoidance throughput, our model provides the foundations for using simple estimations of average TCP throughput and may establish bounds of their applicability.

Our model provides complete characteristics of TCP congestion avoidance behavior and incorporates the probabilistic nature of the networking environment. It provides a new, powerful tool for planning and design of the TCP based transport layer, including issues of QoS. Our approach relaxes several important restrictions accepted in most of analytical TCP models:

- The model includes RTT variability. RTT is considered to be a random variable described by its distribution function;
- The model of cwnd size and throughput never exceeds their natural limits, which are parameters of the model;
- The model is valid for high and low bandwidth links, since it does not use "round" modeling (see details in [PFTK01] or Section 2 in this paper).
- RTT and segment loss probability may (or may not) depend on the congestion window size.

Restrictions that we relax become parameters of the model. This feature makes the approach highly flexible and applicable to a wide range of network environments. Detailed discussion of the modeling assumptions is given in Section 3.

All results, including the distributions mentioned above, are obtained in explicit analytical form. We have developed an algorithm that provides fast calculation of the cwnd distribution with linear complexity.

Our analytical results are validated by using observations of experimental TCP connections in an emulated network environment. In addition, we compare our model with other highly referenced models of congestion avoidance [Flo99, PFTK01]. The average TCP throughput provided by the model demonstrates high stability and good agreement with the experimental data (relative error within 5%-15% for certain experimental metrics).

The paper is organized as follows. Section 2 contains the formulation of the congestion avoidance throughput (T_{CA}) model. Related work and a comparison between our T_{CA} model and related work are discussed in Section 3. Section 4 presents an evaluation of the cwnd distribution and Section 5 evaluates the T_{CA} distribution. Several numerical examples are presented in Section 6, including an analysis of the behavior of T_{CA}, QoS and the dependence of the expected congestion avoidance throughput on RTT variability, sender's link capacity and the receiver window size. A presentation of the model validation is given in Section 7. Section 8 provides general conclusions and a discussion of the developed methods. The paper contains 14 illustrations that (except one) are placed in Appendix A. Appendix B presents validation results. Appendix C contains details of the congestion window size distribution derivation.

Chapter 2

TCP Congestion Avoidance Algorithm Modeling Assumptions

2.1 Basic assumptions

First we formulate the assumptions of our model.

- We only consider the AIMD portion of TCP congestion control. That is, the congestion window, cwnd, increases by one segment per RTT if no segment is lost. When the loss of a TCP segment is detected at the sender because it has received three identical ACKs from the receiver, then cwnd is reduced by one-half. No retransmission is considered.

- Segment losses occur according to a segment loss pattern. The segment loss pattern is defined by the distribution of the number of segments sent in succession between two consequent loss indications. Our assumptions allow any memoryless segment loss pattern. The memoryless property of the segment loss pattern means that the probability of "n consequent segments are delivered to the receiver's host" does not depend on how many segments have been delivered in the past (in virtual time), where "past" starts at the closest loss indication.

  For example, the Bernoulli scheme (or Bernoulli segment loss pattern) is memoryless. Let's denote p as the segment loss probability. The probability to deliver any particular segment is $(1-p)$, $0 < p < 1$. It does not depend on whether previous segments were delivered.

  We do not specify the reasons for losses. These may be due to congestion or link error. The model requires that the segment loss pattern remains unchanged over real time (or with the sequence number). However, the segment loss pattern may depend on the congestion window size, cwnd.

  In the case of congestion losses, note that congestion avoidance requires the protocol to decrease its window size at the first notification of possible congestion. This means that in terms of the model, the protocol intends to prevent changes of the segment loss pattern.

When heavy congestion is present, the loss events are likely to become correlated. In the case we expect that a connection experiences multiple timeouts. Hence, the connection reduces its cwnd to one, its throughput degrades, and the connection spends little time in the congestion avoidance phase. We do not treat this case of heavy congestion.

- We also include in our model the sender's link capacity as a parameter. TCP usually has a virtual link that buffers the segments sent. Therefore, TCP itself usually does not "feel" the limit of available capacity directly. Nevertheless the restriction is important for the following reasons. Let us suppose that the RTT is constant and equal to one time unit. If there are no losses, then cwnd increases by 1 each RTT. If we start from cwnd $= 1$, then during n rounds TCP sends

  $$1 + 2 + \cdots + n = \frac{1}{2}\left(n^2 + n\right)$$

  segments. Since RTT is 1 time unit, then n RTTs take n time units, which means that amount of data sent is a quadratic function of time. This relation never takes place in any real network because it requires an infinite network capacity. In practice, TCP reaches the link capacity with quadratic speed, then remains at the link capacity until the next segment loss is indicated. Hence, the actual number of data segments sent is linear or smaller than some linear function of time.

  Since the configuration of the end-to-end path as well as the capacity available to the connection are assigned randomly, they are difficult to predict. In most cases we cannot even characterize them by the distribution function. The distant bottleneck informs the TCP sender about its capacity by dropping segments and hence participates in the model through the segment loss pattern.

  Throughput cannot be higher than the sender's link capacity, which is a natural limit. We consider this restriction for our model when we analyze throughput from the sender's point if view. Nevertheless in many cases, this is a very rough estimation. If one can derive the bottleneck capacity of the end-to-end path, then it may replace the sender's link capacity.

- The maximal window size is restricted by some given value. It may be the receiver advertised window size (rwin) or any other restriction, as well[1].

From this set of assumptions we derive the distribution of cwnd and congestion avoidance throughput as functions of the segment loss pattern, the RTT distribution function, the maximal window size and the sender's link capacity.

There are several restrictions accepted in most T_{CA} models that *are not included* in our model. The most important one is "round" modeling [CSA00,

[1] For example high bandwidth-delay networks provide their own restrictions [LM97].

YR01, Flo99, MGT98, MSMO97, PFTK01, SKV01, SKVPE01]. The "round" assumption means that the TCP sender is assumed to transmit all segments falling within the current window back-to-back and then waits for the first acknowledgment for one of these segments. This ACK starts a new "round," i.e., the TCP sender transmits the next window back-to-back. Although this approximation in some sense contradicts the sliding window algorithm, it might be acceptable when the sender's link rate (the time needed to pass one segment to the network) is much smaller than RTT. This model is not suited for slow links (or small RTT's). Moreover it has effects on the limiting behavior of the resulting formulae (i.e., the T_{CA} thus computed tends to infinity for small loss probabilities even for a deterministic segment loss pattern).

Since we do not model recovery and retransmit algorithms, we do not require special assumptions on how the lost segments are distributed inside the round.

2.2 Parameters interdependence

Let us examine the interdependence between three key characteristics of the process, which are the segment loss pattern (or segment loss probability), cwnd and RTT. Consider the following example.

Let the network bottleneck capacity be much smaller than the sender's link capacity. Hence the TCP sender increases cwnd and its flow rate. Soon it over utilizes the bottleneck capacity. If the bottleneck device has significant buffer space, then the queueing component of RTT increases and creates dependence between the window size and RTT (i.e., a larger cwnd causes a larger RTT, and vice versa). If there is no buffer space at the bottleneck, then a larger cwnd will cause larger segment loss probabilities.

We provide examples in our analysis that show the shape of cwnd size distribution when segment loss probability depends on the size. Our model allows a dependence of RTT distribution on cwnd. However, it does not allow a dependence of the segment loss pattern or RTT distribution on time (real or virtual).

Chapter 3

Related work

There has been extensive research on TCP modeling, of which we present a selection of the research. Cardwell, et al. [CSA00] is a good starting point, although it does not include several recent papers. The motivations of much research have been to understand the impact of network and protocol properties on data communication performance [AAB00, ABBC, BH00]. Another important motivation is an attempt to distinguish between TCP-friendly and unresponsive best-effort flows with the aim to quantify the notion of TCP friendliness [YR01, Flo99, MGT98, PFTK01]. Efforts of several researchers yield expressions for TCP throughput expectation or its estimations (most of them are upper bounds). Another class of the models [CSA00, SKV01, SKVPE01] deal with short TCP connections that slightly suffer from segment losses.

Note that the expected value of a single random variable does not provide enough information for understanding TCP behavior. The outcome of a single random experiment is restricted only by the natural bounds that are implied by the experimental formulation. The throughput of a single TCP connection lays between zero and the bottleneck (or the sender) capacity. Any estimation placed between those natural bounds must be associated with the probability that expresses "how often" the estimation (including the expectation) is represented. It means that the constructed bound needs to be extended by the probability associated with it.

We consider the steady state behavior of congestion avoidance. We provide for the first time a complete set of characteristics of its behavior in the form of the cwnd distribution and the T_{CA} distribution. Further processing of the T_{CA} distribution yields its multiple important metrics, including expectation, standard deviation, QoS, and interval estimations.

The assumptions we make in our model compare with the assumptions made in the related work in the following ways:

- The restriction on the set of TCP algorithms under analysis by our model (i.e., the AIMD algorithm) is accepted in most investigations. In many cases only congestion avoidance is selected [Flo99, MSMO97]. In several works [MGT98, PFTK01], timeouts are included in the model and others [CSA00, YR01, SKV01] consider the slow start phase. Nevertheless these models accept other types of the restrictions that are relaxed in our study and will be detailed below.

- Most models use strong restrictions on the segment loss pattern. Several use a deterministic segment loss pattern, which means that the number of segments delivered between a consequent loss indication is fixed as a deterministic variable [Flo99, MSMO97] or the different loss scenarios are determined itself [ABBC, LM97, SKVPE01]. Many researchers accept the Bernoulli scheme [YR01, LM97, NHYKO01, PFTK01] with the fixed parameter. Misra et al. [MGT98] interprets segment losses as a Poisson flow (that is a limited version of the Bernoulli scheme) and Sikdar et al. [SKVPE01] use a uniform distribution.

  Several latest studies implement more generalized approaches. Baccelli and Hong [BH00] propose a framework that describes in detail the particular segment loss pattern (deterministic or random) in a generic manner. Altman, et al. [AAB00] introduce the most general approach describing segment losses by a general stationary ergodic point process over real time. The last is closest to our approach because the defined segment loss pattern becomes a parameter of the T_{CA} model. Hence, the same baseline model covers a range of networking environments.

  Our model accepts any memoryless segment loss pattern defined in terms of segments over virtual time. We do not specify reasons behind the losses in the pattern in detail by the argument we discuss above. The memoryless restriction covers many practical patterns and keeps the model tractable. Also, some non-memoryless patterns may be transformed to memoryless patterns. The segment loss pattern defined in our model allows dependence on the congestion window size cwnd.

- Restricted maximal cwnd size w_{max} is implemented in some of existing models [ABBC, BH00, CSA00, LM97, SKV01]. Some ignore it [YR01, MGT98]. Therefore, the deterministic segment loss pattern naturally implies the restriction. Most models accept that a random segment loss pattern treat cwnd as unrestricted. The restriction is set as an extension of the baseline model in studies [AAB00, PFTK01, SKVPE01].

- RTT is considered a constant value in most research. Our model treats RTT as a continuous random variable characterized by its distribution.

- With regard to the issue of limits on T_{CA}, we mention two approaches that have dominated recent studies. The first is typical for TCP performance modeling research that concentrates on the properties of the TCP algorithms. This analysis usually only partially relates to the properties of the environment in which TCP operates. Therefore T_{CA} is either unlimited or limited by the maximum cwnd value. The available network capacity is not considered in works [AAB00, Flo99, MSMO97, NHYKO01, PFTK01, SKVPE01].

  The second approach mainly concentrates on the properties of the end-to-end path, and combines them with some model (may be a rough estimate) of the protocol. For example, see [ABBC, BH00, LM97, SH01, WJ00]. Under this approach the network capacity itself is an object of the model and therefore the throughput is bounded by the model. Those studies focus on the behavior of multiple TCP connections or the bottleneck queue with TCP generated arrivals.

Nevertheless, it is extremely difficult to predict the end-to-end path that a TCP connection (or family of TCP connections) will traverse. Moreover, the end-to-end path may change during the connection lifetime. Therefore we introduce in our model a value that expresses an upper limit of T_{CA} and thus integrates characteristics of the network. The same is applied to the segment loss pattern (or segment loss probability).

The limit makes the model realistic and allows model layering. The network capacity limit as well as the segment loss pattern might be obtained through other models of the network. (An example of this approach is given in [REV01].) The sender's link capacity provides the simplest, although a rough estimate of the bottleneck link.

Chapter 4

Distribution of the cwnd size

Under our assumptions, the sequence of cwnd size w used by a consequent rounds forms a Markov chain. Its transition probabilities are defined by the segment loss pattern. Let us denote g^i the probability that more than i segments are successfully delivered in succession and g_i the probability that less than i segments are delivered (i.e., some of those i segments were lost). A simplified fragment of the Markov chain transition diagram is shown at Figure 4.1. Based on this interpretation we built the Kolmogorov equation corresponding to the Markov process and have obtained its solution in explicit analytical and recurrent forms.

$$\begin{tikzcd}
 & 2w \arrow[d, "g_{2w}"] & \\
w-1 \arrow[r, "g^{w-1}"] & w \arrow[r, "g^{w}"] \arrow[d, "g_w"] & w+1 \; , \\
 & w/2 &
\end{tikzcd}$$

Figure 4.1: The simplified fragment of transition diagram.

The recurrent form allows us to design a fast algorithm that calculates the solution of the Kolmogorov equation from the given set $\{g_i, g^i\}_{i=2}^{w_{max}}$, where w_{max} is rwin or another cwnd restriction. The solution is the distribution of cwnd size

$$w_i = f(g_i, g^i), \qquad i = 2 \dots w_{max} \tag{4.1}$$

The algorithm has $O(w_{max})$ complexity[1].

Note that the distribution constructed is discretized by rounds (is round-wise), i.e., the corresponding random process is embedded by the ends of each round. We are considering TCP throughput over real time and therefore the distribution needs further transformation. First we pass the moments that the segments were sent to the distribution (i.e., we consider the consequence of cwnd sizes each TCP segment was sent – segment-wise). Under this formulation the

[1] The algorithm implemented on a desktop computer finds a solution for $rwin = 10000$ within a second. Most typical values of $rwin$ in the Internet are about 128 segments.

larger windows *live longer* than the shorter ones. We have derived the multiplier that allows the calculation of the segment-wise distribution from the round-wise distribution. The relation between the segment-wise cwnd distribution and that over real time states Theorem 1 and Theorem 2 in the Appendix C.

Chapter 5

Distribution of the throughput

Let t_0 be the time that the TCP sender needs to inject one segment to the network. T_{CA} depends on the relation between wt_0 and RTT[1]. If $wt_0 > RTT$ then T_{CA} is equal to the sender's link capacity because when the ACK for the first segment of the round arrives, TCP (or its virtual link) still has segments to send and each ACK slides the window. If $wt_0 \leq RTT$ then T_{CA} is $T = w/RTT$. When the whole window has been sent, the sender waits for an ACK. Therefore, the segments are sent ACK-clocked.

Let us consider the second case in detail. We denote $R_w(x)$ as the distribution function of RTT, which may or may not depend on w. We also denote the sequence $\{\omega_i\}_{i=2}^{w_{max}}$ where $\omega_i = \mathcal{P}r(w = i)$. Since $T = w/RTT$, we must consider all those w and RTT such that their relation gives T_0 and sum their joint probabilities to calculate the probability of T_{CA} at the particular value T_0. We have set w as an independent random variable and RTT may depend on it. Therefore

$$\mathcal{P}r(RTT|_w) = \frac{\mathcal{P}r(RTT \cup w)}{\mathcal{P}r(w)} \tag{5.1}$$

and

$$\mathcal{P}r(RTT \cup w) = \mathcal{P}r(w)\mathcal{P}r(RTT|_w)$$

If RTT is independent on w then $\mathcal{P}r(RTT|_w) = \mathcal{P}r(RTT)$ and

$$\mathcal{P}r(RTT \cup w) = \mathcal{P}r(w)\mathcal{P}r(RTT)$$

The throughput distribution function is defined as

$$\begin{aligned} T(x) = \mathcal{P}r(T < x) = \mathcal{P}r(w = 2)\mathcal{P}r(RTT > 2/x) + \\ \mathcal{P}r(w = 3)\mathcal{P}r(RTT > 3/x) + \\ \cdots + \mathcal{P}r(w = w_{max})\mathcal{P}r(RTT > w_{max}/x) \end{aligned} \tag{5.2}$$

[1] This relation also appears in [LM97].

or, rewritten as

$$T(x) = \sum_{i=2}^{w_{max}} \omega_i \left(1 - R_w\left(\frac{i}{x}\right)\right) = \\ 1 - \sum_{i=2}^{w_{max}} \omega_i R_w\left(\frac{i}{x}\right) \tag{5.3}$$

Formula (5.3) is true if $x < L$, where $L = 1/t_0$ is the sender's link capacity and $T(L) = 1$. Therefore, the following steps are needed to calculate the performance metric of T_{CA}

1. Define w_{max} and segment loss pattern.

2. Calculate cwnd size distribution $\{\omega_i\}_{i=2}^{w_{max}}$. With the aim one must define collection $\{g_i, g^i\}$, find the distribution of equation (4.1), transform it to the packet-wise distribution and then apply Theorem 1 or apply Theorem 2 directly to the segment-wise distribution. (See Appendix C for details).

3. Define $R_w(x)$, the distribution of RTT. The shape of the distribution might be set according to the environment properties as well as empirically derived from observations. If RTT depends on the cwnd size, then the $R_w(x)$ distribution should be defined for each value of cwnd.

4. Construct the distribution of equation (5.3).

5. Calculate the necessary metrics (i.e., moment, quantile etc.) from the distribution of equation (5.3).

Chapter 6

Numerical examples obtained on the base model

We present in this section several examples of the evaluation of TCP characteristics based on the model presented above. All figures referenced in this section are presented in Appendix A. The examples demonstrate the scope of the model. The model lets one analyze a wide range of T_{CA} characteristics, including not only T_{CA} expectation, but its variance and other moments, quantiles, and interval estimations. It also lets us establish quantitative functional dependencies of those metrics on the parameters of the networking environment and considers their variability. Several figures show how the expectation and the variance of RTT affect T_{CA} expectation and variance.

For several examples RTT is specified to be distributed according to a normal distribution function with expectation 60 ms and a standard deviation of 30 ms. The values and the shape of the distribution are set arbitrarily for the purpose of the example and are not due to model restrictions. Other parameters of the RTT distribution are clearly given in the legend of the correspondent figures. We use the Bernoulli scheme as the segment loss pattern. For most of the examples the segment loss probability p is constant. Nevertheless it depends on cwnd for several examples.

6.1 Congestion window size distribution

Figures 8.1-8.4 present the cwnd distribution for rwin $w_{max} = 120$ segments and $p = 0.00045$, $p = 0.0006$, $p = 0.001$ and $p = 0.0045$, respectively. Here p is the probability of segment loss detection. For this section all numerical presentations are obtained using the algorithm mentioned in Section 4.

The figures demonstrate three basic types of the cwnd behavior. For small values of p, cwnd is likely to reach w_{max} and stay there for a long time. This creates a peak at w_{max}. If a segment is lost at cwnd=w_{max}, then cwnd is halved and this creates another smaller peak of the distribution at $w_{max}/2$.

If p becomes larger, then cwnd demonstrates unstable behavior and fluctuates around a large range of sizes. Therefore the distribution loses its maximum (local or global) at w_{max}. The shape of the distribution depends on the relation between w_{max} and p. As w_{max} becomes larger, then p must be smaller to enable a stable cwnd. For values $p > 0.1$, the distribution is concentrated in the area of small windows (less than 10 segments) and hence $w_{max} > 10$ does not affect its shape.

Figure 8.5 presents the cwnd expectation and a standard deviation as a function of p for $w_{max} = 70$. For each value of p we have calculated the corresponding cwnd distribution and then its expectation and standard deviation. The expectation crosses the y axes at w_{max}. Our calculation shows that the cwnd standard deviation depends very slightly on w_{max}. It has a maximum between $p = 0$ and $p = 0.05$. Obviously the standard deviation must be zero for $p \approx 1$ or $p = 0$ and it is nonzero between these two points. Therefore, it has at least one maximum as a function of the segment loss probability. The maximum corresponds to the case of cwnd fluctuating around a wide range of values.

Figure 8.6 shows different cases of dependence between p and cwnd. We introduce a multiplier α, which defines the degree of the dependence. We suppose that p depends on the cwnd size w as $p(w) = \alpha^w q$, where $0 < q < 1$ is some initial value of the segment loss probability. The distribution of Figure 8.6 is obtained for $\alpha > 1$.

The case of $0 < \alpha < 1$ looks rather theoretical, although it is not beyond the scope of this model. We cannot provide an example of the corresponding environment. The case of $\alpha > 1$ might be provided by the environment that cannot properly process incoming segments at a particular arrival rate (see Section 2.2).

6.2 Expectation of throughput

Here we present the expectation of T_{CA} (identified as ET) as a function of p for different environments. We have used the procedure described in Section 5 for the calculations.

Fig. 8.7 shows ET for $w_{max} = 70$ and different sender's links, i.e., 100 segments per second (segs/s), 200 segs/s and 500 segs/s. Fig. 8.8 shows ET for link capacity 1000 segs/s and different w_{max}, i.e., 20, 40, 60. Note that for $p > 0.03$, the parameter w_{max} does not affect T_{CA} since large windows are never reached.

The presented model of T_{CA} never exceeds the given link capacity. Our results show whether T_{CA} utilizes the link capacity for the given combination of w_{max} and RTT distribution. These and similar figures let us design and/or exploit the protocol and the environment more efficiently since they bind network parameters and the maximum of reachable throughput.

We also mention that the expectation of cwnd and ET have the same shape. Characteristics of T_{CA} are highly determined by its cwnd size. This feature highly correlates with practical experience.

Figures 8.11 and 8.12 illustrate the influence of RTT distribution parameters on the expectation and standard deviation of T_{CA}. Both figures are generated using $w_{max} = 30$ segments and the sender's link capacity at 300 segs/s. As expected, an increase of RTT decreases the expected T_{CA} and an increase of the RTT standard deviation increases T_{CA} for all significant p. The impact of retransmitted segments is believed to be negligible.

6.3 Standard deviation of the throughput

The deviation of T_{CA} plays a significant role as it shows the stability of a connection and may be crucial for the QoS. Our investigation shows that the variance of cwnd and the variance of RTT mainly contribute in the variance of T_{CA}. Like the cwnd deviation, the T_{CA} deviation may have a maximum in the same range between $p = 0$ and $p = 0.05$, but it may also demonstrate monotonous behavior. This happens if RTT is large and highly variable, and the sender's link is fast or cwnd is small enough and cannot utilize the link. In these cases the impact of RTT on deviation increases. Therefore, to decrease the deviation for a fast link one must not only decrease the segment loss probability but also increase w_{max}. Our model provides not only qualitative but quantitative analysis, as well.

Figures 8.9 and 8.10 present the standard deviation of T_{CA} as a function of the segment loss probability p. Figure 8.10 shows the T_{CA} deviation for the same link capacity 550 segs/s and a different w_{max}, i.e., 30, 50 and 70 segments. Figure 8.9 shows the T_{CA} deviation for different links, i.e., 100, 200 and 300 segs/s and $w_{max} = 50$ segments. One may see that the deviation reaches significant values (about 1/3 of expectation) at peaks. Note also that for $0 < p < 0.05$, slight changes of p lead to significant changes of T_{CA} deviation and may have a crucial effect on the QoS, which can be predicted using the model.

6.4 Quality of Service.

The T_{CA} model may be used to characterize QoS of a network environment. Probabilities that T_{CA} is above and below a given level (e.g., half of the link capacity) are shown in Figure 8.13. These values might be also used to construct the probability that T_{CA} lays between certain bounds. The lines cross at probability 0.5 and their sum is always equal to 1. As Figure 8.13 shows, slight changes of p may have significant effect on the QoS for $p < 0.05$. The presented model provides quantitative analysis of the phenomena.

Chapter 7

Validation of the model

We validated our model of the expected value of T_{CA} using a TCP network emulator. We do this because the expected value of T_{CA} is an easily observable metric and many TCP analyzing tools provide the corresponding values. In addition, the expected value of T_{CA} lets us compare the results provided by our model with other models that analyze average TCP throughput. Therefore, we compare values of average throughput provided by our and two other models with values derived experimentally.

7.1 Arrangement of the experiments

The experiments were conducted on Seawind [KGMSAR01], the real-time software network emulator running on Linux developed at the Department of Computer Science of the University of Helsinki. Seawind enables measurements of protocol implementations with parameters that the analyst may manage. Seawind intercepts the traffic flow between the client and server hosts. Then, the desired link characteristics are emulated by delaying, dropping and modifying packets in the flow.

The link is modeled as a direction specific channel that is maintained separately for the uplink (from sender to receiver) and the down-link (from receiver to sender). Packets that arrive at the emulator are placed into the *input queue*, which may be interpreted as a router queue and thus inspects the effect of congestion based losses. The link behavior is simulated by the number of consequent *delays* (i.e., transmission and propagation delay in our case) and transmission errors. Therefore, it allows us to study the combination of congestion losses and link error losses.

Nine sets of experiments were conducted. The first of them contains ten connections and the remaining contain three connections. As a result, we have analyzed 35 TCP connections. For all connections we have set a propagation delay of 200 ms. The input queue holds three packets with a drop tail queue management policy. Transmission errors on a link were evaluated on a per-packet basis in the *drop* mode, which means that after a transmission error the corrupted packet is dropped.

For the first set of tests (we note it as 1) we used a link rate of 28800 bits/sec for both directions, a packet transmission error probability of 0.005 and a MTU of 296 bytes. The next three sets (2 – 4) were conducted with the packet transmission error probabilities of 0.025, 0.005 and 0.0025, respectively. The remaining five sets (5 – 9) were conducted with the link rate of 0.5 MBits/sec for both transfer directions and an MTU of 576 bytes. The packet transmission error probabilities were 0.025, 0.005, 0.0025, 0.0005 and 0.00025. A bulk transfer workload was generated using the *ttcp* tool [SRH90].

Seawind provides filter logs, which shows the relevant protocol information about the segments injected to to the network. The output of the logs is compatible to output generated by *tcpdump* [JLM97]. The log-files created by Seawind were analyzed by the *tcptrace* tool [OS]. We extracted RTT samples and set of the values characterizing connection behavior from the log-file.

7.2 Processing data

The following parameters must be estimated using empirical observations to calculate the T_{CA} expectation: segment loss probability, RTT distribution, receiver advertised window and sender's link capacity. The last one is determined exactly as an emulation parameter and the receiver advertised window is given explicitly in the *tcptrace* report.

There are several possible estimations of the segment loss probability. The natural estimation is the relation between the number of lost segments and the total number of segments sent. The number of triplicate acknowledgements in most cases provides the lower bound of the number of lost segments, especially if several segments are lost in a row or if a timeout occurs. Nevertheless, the number of the retransmitted segments in most cases overestimates number of lost segments due to unnecessary retransmissions under the cumulative acknowledgments scheme. Moreover segments might be lost during retransmission and hence retransmitted several times. This number may provide the best estimate when using the TCP SACK option (Selective Acknowledgements). For our study we specify the number of loss segments as a value between the number of post-loss ACKs and the number of the triplicate ACKs.

Rigorously, the RTT distribution function is continuous just as time is a continuous variable. In practice, we typically use piecewise distributions. We know that RTT may not be smaller (or larger) than some natural bounds and hence its distribution must have breaks at those points. Fitting a piecewise function to observations is a rather complicated problem. Using a continuous function for piecewise data creates computational errors that contribute to the T_{CA} estimation. For those reasons we use an empirical distribution and use frequencies built from the base of RTT samples as a cdf and pdf of RTT.

There exist a large variety of TCP implementations. The properties of a particular implementation may be out of the scope of the documented requirements or even strictly contradict them. This certainly creates significant difficulties for modeling. In our experiments we used a Linux implementation of TCP. Following IETF requirements, Linux TCP tries to avoid burst. With that goal, it drops cwnd to 2 segments if the pointer of the left bound and the pointer inside the sliding widow coincide. This feature has a significant effect on cwnd. In most cases for the T_{CA} evaluation we used a round-wise distribution that is stochastically "smaller" than the true one. Three connections from our series of experiments did not implement this feature and they are calculated with the true cwnd distribution.

7.3 Results of data analysis

We present our results using a series of tables. Table 7.1 is given below and the remaining are placed in Appendix B. We consider several metrics to characterize the throughput of real TCP connections, which will be compared with the model's value of throughput of congestion avoidance. These are

- Average throughput ATp. The value is provided by the *tcptrace* tool

  $$ATp = \frac{\text{Total data delivered}}{\text{Elapsed time}}$$

  This metric is the classical one.

- Sender's data transfer throughput ATpXmit

  $$ATpXmit = \frac{(\text{Unique bytes sent} + \text{Retransmitted data bytes})}{\text{Data transfer time}}$$

  This metric counts all data sent by the TCP sender over real time cleared from handshaking and other additional procedures.

- Throughput "without timeouts" NoTOTput

  $$NoTOTput = \frac{\text{Average cwnd}}{\text{Average RTT}}$$

  If the TCP connection is not idle and cwnd is small such that it does not reach the sender's link capacity, then the connection sends w segments during one RTT and therefore its throughput is

  $$T = \frac{w}{RTT}$$

  If cwnd and RTT are independent, the T_{CA} expectation is

  $$\mathsf{E}(T) = \mathsf{E}(w) * \mathsf{E}\left(\frac{1}{RTT}\right)$$

  Since the function $1/x$ is convex, according to the expectation properties

  $$\frac{1}{\mathsf{E}(RTT)} \leq \mathsf{E}\left(\frac{1}{RTT}\right)$$

  and hence for connection activity periods

  $$\mathsf{E}(T) \geq \frac{\mathsf{E}(w)}{\mathsf{E}(RTT)}$$

> This metric defines the throughput that the connection could have if there were no timeouts or other kinds of delays. Therefore, it is closer to T_{CA}. The error provided by the metric increases when the TCP connection throughput approaches the link capacity, which is the case that breaks the round modeling assumption. Under the "round modeling" assumption, the following relation takes place $ATp \leq ATpXmit \leq NoTOTput$.

We used three models of TCP throughput to compare results with the experimental data and with each other. Values provided by our model are marked ET. FF is T_{CA} estimation from Floyd and Fall [Flo99], and $PFTK$ is T_{CA} estimation from Padhey et al. [PFTK01]. The estimations that are larger than the maximum link capacity are marked by bold font.

The sets of tests are marked by numbers and the link rates and packet transmission error probabilities are given in titles of the tables. Connections inside one replication set are marked by a dual number. Each connection lists two strings in the table. The first one presents absolute values and the second one gives relative error in percents between ET and the three experimental metrics. Values calculated with the true cwnd distribution are marked with an asterisk symbol.

The experiments show that our model has the best accordance with the NoTOTput metric when it approaches T_{CA}. The other two experimental metrics demonstrate variable accordance that depend on experimental conditions. For the ATp metric, 22 observed connections have relative error within 0%-20%, 10 are between 20%-35% and three connections have relative error approaching 50%.

We see that the estimation of ET is never significantly worse than FF and PFTK. In addition, ET demonstrates very reliable and confident behavior. For the sets 6 – 9 (Tables 8.5 – 8.8 of Appendix B), all three estimations differ only slightly. Here rwin of 64 segments is almost never reached due to the high packet loss probabilities. Even if it is reached, it happens at the slow start phase but not in congestion avoidance. At the same time the throughput reached by the connections is far from the link capacity. Therefore, we have the case of an unrestricted window size and pure "round" modeling accepted for FF and PFTK. All three models provide very close results.

In set 5 (Table 8.4 of Appendix B), due to the small packet loss probability congestion avoidance reaches and spends most of its time at the receiver advertised window. Therefore our model provides the best estimation. The difference grows with a reducing packet loss probability since FF and PFTK tend to infinity in this case.

The ET estimation also demonstrates very good properties in sets 1 – 4 (Table 7.1 and Tables 8.1 – 8.3 in Appendix B). Its relative error with the NoTOTput metrics stays within 10%. It is within 5% in most cases. At the same time for the sets 1, 3 and 4, FF and PFTK provide useless results since their numbers are larger than the maximum link capacity. Hence the capacity itself provides a better estimation than FF and PFTK.

Several test expose properties of FF and PFTK estimations, which are generated by considering RTT as a constant value. We used the mean of RTT sam-

Link rate 28800 bits/sec (**14.118** segments/sec).
Set 1. Packet transmission error probability 0.005.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
1.1	11.33 *6.15%*	11.56 *4.02%*	12.03 *0.06%*	12.02	**15.50**	**15.34**	segs/s
1.2	10.46 *14.85%*	10.62 *13.12%*	12.22 *1.73%*	12.01	**14.89**	**14.69**	segs/s
1.3	11.51 *1.85%*	11.69 *0.28%*	12.01 *2.43%*	11.72	**14.95**	**14.82**	segs/s
1.4	9.48 *26.17%*	9.66 *23.85%*	11.56 *3.42%*	11.96	**14.85**	**14.63**	segs/s
1.5	10.71 *6.66%*	10.85 *4.94%*	11.85 *6.23%*	11.32	13.06	12.91	segs/s
1.6	10.04 *6.70%*	10.20 *5.00%*	11.42 *6.20%*	10.71	10.84	10.69	segs/s
1.7	10.33 *14.68%*	10.54 *12.40%*	11.83 *0.14%*	11.85	**14.39**	**14.19**	segs/s
1.8	10.6 *9.72%*	10.78 *7.92%*	12.01 *3.09%*	11.64	**15.72**	**15.51**	segs/s
1.9	11.65 *7.95%*	11.88 *5.93%*	12.18 *3.30%*	12.58	**17.97**	**17.83**	segs/s

Table 7.1: Experimental and modeling T_{CA} (Absolute values and relative error). Values exceeding the bandwidth capacity are marked by bold font.

p	FF	ET, rwin=10	ET, rwin=40	ET, rwin=80
0.9	1.90	3.72	3.72	3.72
0.5	2.55	3.73	3.73	3.73
0.1	5.70	7.21	7.22	7.22
0.05	8.07	9.99	10.33	10.33
0.01	18.03	14.79	24.01	24.01
$5 * 10^{-3}$	25.51	15.56	34.05	34.31
10^{-3}	57.04	16.17	55.49	71.50
$5 * 10^{-4}$	80.66	16.26	59.60	87.55
10^{-4}	180.38	16.32	62.99	103.79
10^{-5}	570.41	16.34	63.76	107.46
10^{-6}	1804.00	16.34	63.83	107.75
10^{-7}	5704.00	16.34	63.84	107.79
10^{-8}	18040.00	16.34	63.84	107.79

Table 7.2: Models comparison. Throughput units are segments/sec.

ples for their RTT parameter since there is no other recommendation available. The effect appears very clearly for connection 3.1, 4.1 and 5.1 (Tables 8.2, 8.3, and 8.4 in Appendix B). Here RTT samples have outliers that have a large value. This population significantly shifts the RTT mean and hence unexpectedly reduces the FF and PFTK estimations. Note that the estimation provided by our model stays stable in all those cases since it accounts for all RTT values with their frequencies. This effect also keeps estimations of FF and PFTK under the link capacity in set 2 since there are RTT frequencies that look like two-“hump” functions.

Table 7.2 provides further comparison of the FF (model of [Flo99]) and ET (our model) estimations. It contains T_{CA} expectation values for rwin 10, 40 and 80 segments and a maximum link capacity of 115 segs/s.

We have demonstrated that if the connection satisfies the restrictions of [Flo99] and [PFTK01], all three models provide very close results. Obviously estimations of FF and PFTK are computationally simpler than ET. However, usability of simple T_{CA} estimations varies. Our analysis shows that the same packet loss probability may or may not fit this area of usability, which depends on the network environment under analysis. Therefore, our model provides an apparatus to use a simple average throughput estimation and to discover bounds of their correctness for the particular implementation.

Chapter 8

Conclusion and discussion of the results

We developed an analytical Markovian model of TCP AIMD congestion avoidance throughput. The model yields the distribution of the congestion window size and congestion avoidance throughput (T_{CA}). Further processing of the distribution provides multiple metrics of T_{CA}, including its expectation, variance, other moments, and interval estimations. The distributions are obtained in explicit analytical form. We also obtained the recurrent expression for the congestion window size distribution. On that basis, we developed the algorithm that allows the calculation of the cwnd distribution with $O(w_{max})$ complexity, where w_{max} is the receiver advertised window or other restriction on cwnd.

The paper contains multiple numeric examples demonstrating the scope of the model and exposing important features of AIMD algorithm revealed by the modeling. Our analytical results are validated by observations on real TCP connections run in an emulated environment. For our model, we have input parameters that are

1. Memoryless segment loss pattern.
2. Distribution function of RTT.
3. Maximal window size.
4. Sender's link (bottleneck) capacity.

The results of modeling produce

1. Distribution of the congestion window size, cwnd.
2. Distribution of the congestion avoidance throughput.

The distribution of cwnd lets us understand the behavior of cwnd under a given segment loss pattern, and compare the cwnd behavior among different segment loss patterns. It also shows which cwnd sizes appear more frequently and which do not appear at all. Further processing of the cwnd distribution provides its moment of any degree, quantiles and interval estimations.

Although the set of available segment loss patterns is restricted by the memoryless property, it essentially widens most previous approaches. For instance, the Bernoulli scheme for segment loss may depend on window size. Also some non-memoryless models may be transformed to memoryless models.

The new results of our model reveals the influence of different factors on congestion avoidance throughput. It allows us to compare the throughput behavior between different segment loss patterns, the same segment loss patterns with different parameters and different RTT distributions or different parameters of the same RTT distribution. The presented model has a realistic limiting behavior since it does not tend to infinity for small segment loss probabilities. It is valid for fast and slow links, as well as those that have a high bandwidth-delay product. The model also provides QoS characteristics.

A comparison of our model with other existing models shows that the models use a more narrow set of parameters and wider set of assumptions. For example, the segment loss pattern is assumed to be deterministic or a Bernoulli scheme, and RTT distribution participates only through its expectation. The maximal window size or maximum available capacity is not considered at all in many models.

As our model relaxes restrictions, it demonstrates high flexibility and applicability and affords a new level of design and planning for the TCP transport layer. The presented model provides foundations for using computationally simpler estimations of average TCP throughput and may exactly obtain bounds of their correctness.

Acknowledgements

We express our gratitude to Pasi Salorahti for his friendly help in conducting the experimental evaluations.

Bibliography

[AAB00] Altman E., Avrachenkov K., Barakat C., A Stochastic model of TCP/IP with Stationary Random Losses, Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, 2000, pp. 231-242.

[ABBC] Altman E. Bolot J., Brown P. Collnge D., Performance Modeling of TCP/IP in Wide-Area Network, ISSN 0249-6399 apport de recherche Iustitut National de Recherche en Informatique et en Automatique

[BH00] Baccelli F., Hong D. TCP is Max-Plus Linear and what it tells us on its throughput, Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, 2000, pp.219-230.

[CSA00] Cardwell N., Savage S., Anderson T. Modeling TCP Latency, Proceeding of INFOCOM 2000.

[YR01] Yeom I., Reddy N. Modeling TCP Behavior in a Differentiated Service Network, IEEE Transactions on Networking 9(1) 2001, pp.31-46.

[Flo99] Floyd S., Fall F. Promoting the use of end-to-end congestion control in the Internet. IEEE/ACM Transactions on Networking, 7(4) 1999, pp.458-472.

[LM97] Lakshman T.V., Madhow U., The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Trans. Networking 5 (3) (1997) pp.336-350.

[MGT98] Misra V., Gong W., Towsley D. Stochastic Differential Equation Modeling and Analysis of TCP-Window size Behavior, Technical report, Univ. of Massachusetts, 1998.

[MSMO97] Mathis M., Semke J., Mahdavi J. Ott T. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, Computer Communication Review, 1997(8).

[NHYKO01] Naito K., Hiraku O., Yamazato T., Katayama M., Ogawa A., New Analytical Model fro the TCP Throughput in Wireless Environment, Proceedings of VTC Spring 2001, pp.2128-2132.

[PFTK01] Padhey J., Firoiu V., Towsley D., Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation, IEEE/ACM Transactions on Networking 8(2) 2001, pp.133-145.

[REV01] Roughan M., Erramilli A., Veitch D., Network Performance for TCP Networks. Part 1: Persistent Sources, Proceedings of Seventeenth International Teletraffic Congress, Salvador da Bahia, Brazil, September 24-28, 2001.

[SKV01] Sikdar B., Kalayanaraman S., Vastola K.S. Analytic model and comparative study of the latency and steady-State throughput of TCP Tahoe, Reno and SACK, GLOBECOM'01, Vol.3 2001 pp.1781-1787.

[SKVPE01] Sikdar V., Kalyanaraman S, Vastola K.S. An integrated model for the latency and steady-state throughput of TCP connections, Performance Evaluation 46 (2001) pp.139-154.

[SH01] Schwefel H. Behavior of TCP-like elastic traffic at a buffered bottleneck router, Proceedings of INFOCOM 2001.

[WJ00] Warland J., A Transaction-Level Tool for Predicting TCP Performance and for Network Engineering, Proceedings of 8th MASCOTS Conference, 2000, pp.106-112.

[KGMSAR01] Kojo M., Gurtov A., Manner J., Sarolahti P., Alanko T., Raatikainen K. Seawind: a Wireless Network Emulator In Proceedings of 11th GI/ITG conference on Measurement, Modeling and Analysis (MMB 2001), Aachen, Germany, September 2001.

[JLM97] Jacobson V., Leres C., McCanne S. tcpdump. Available at: http://ee.ll.gov/, June 1997.

[OS] Ostermann S. tcptrace. Available at: http://jarok.cs.ohio.edu/software/tcptrace/tcptrace.html

[SRH90] Stine R.H. FYI in a network management tool catalog: Tools for monitoring and debugging TCP/IP internets and interconnected devices. IETF RFC 1147, April 1990.

Appendix A. Illustrations for the numerical examples

In this Appendix one may find all figures referenced in Section 6. Figures are placed in the order of reference.

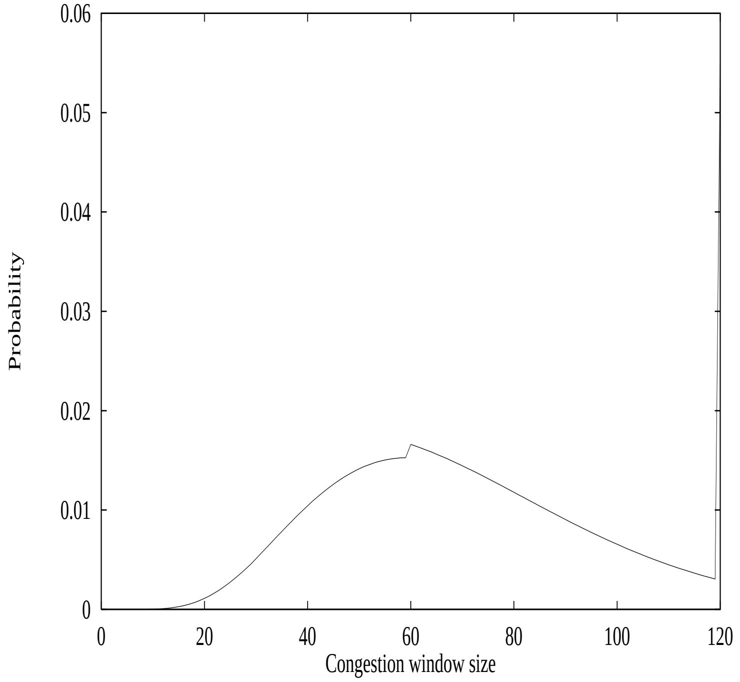

Figure 8.1: cwnd distribution. p=0.00045

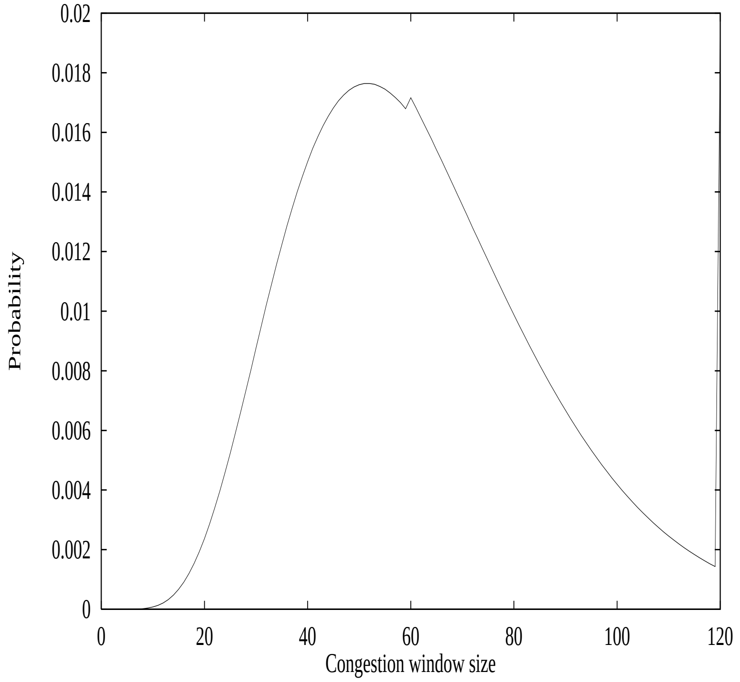

Figure 8.2: cwnd distribution. p=0.0006

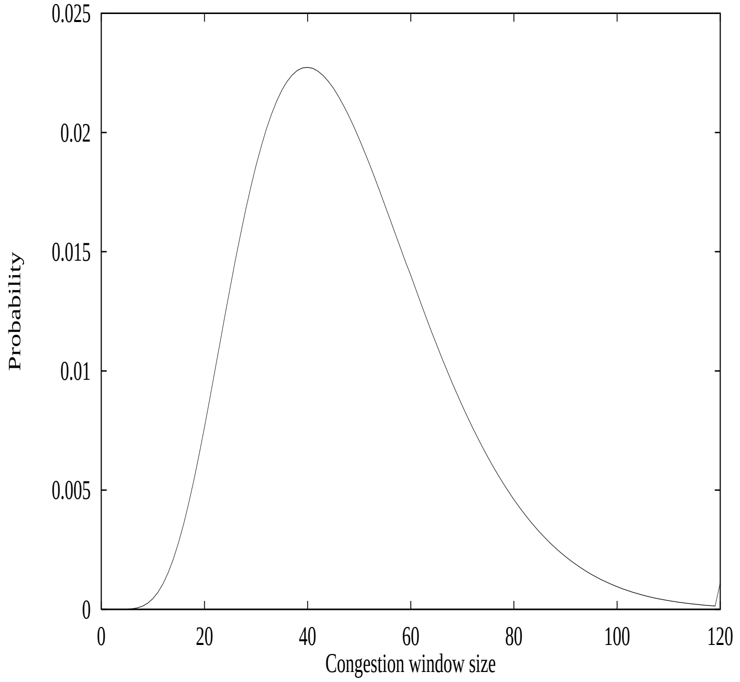

Figure 8.3: cwnd distribution. p=0.001

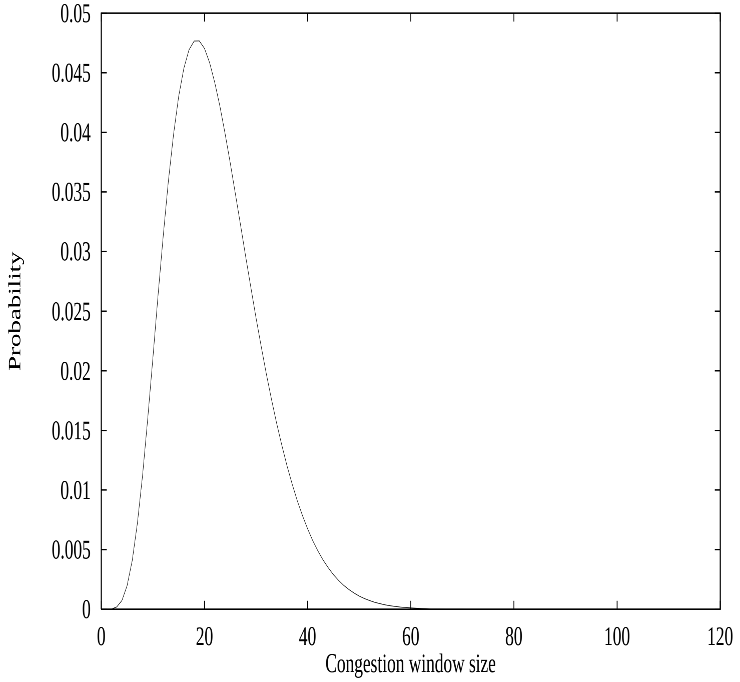

Figure 8.4: cwnd distribution. p=0.0045

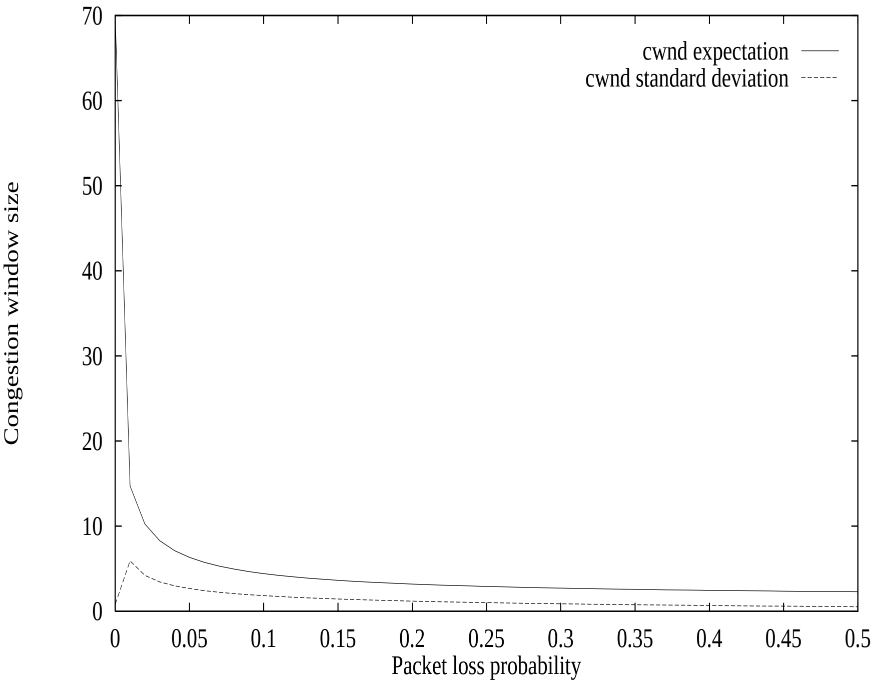

Figure 8.5: Expectation and variance of cwnd.

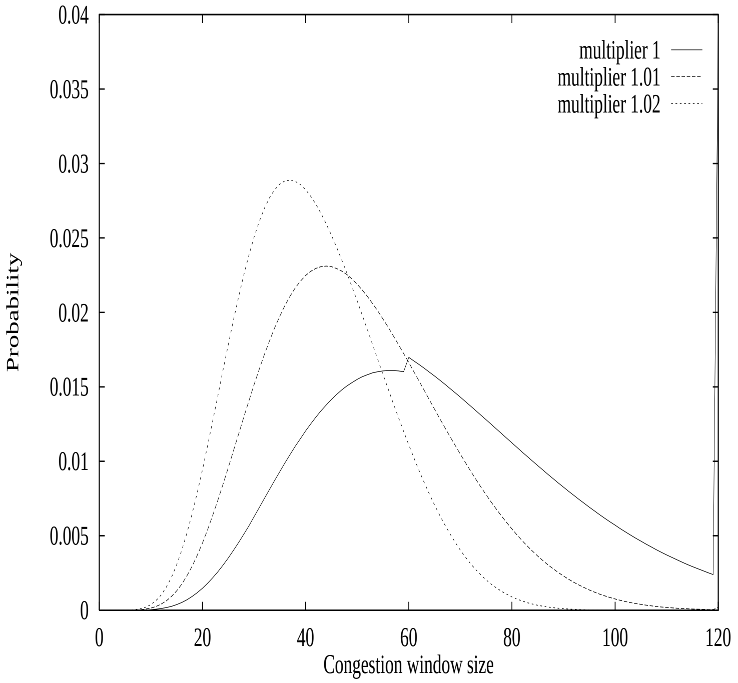

Figure 8.6: cwnd distribution. Case $\alpha > 1$

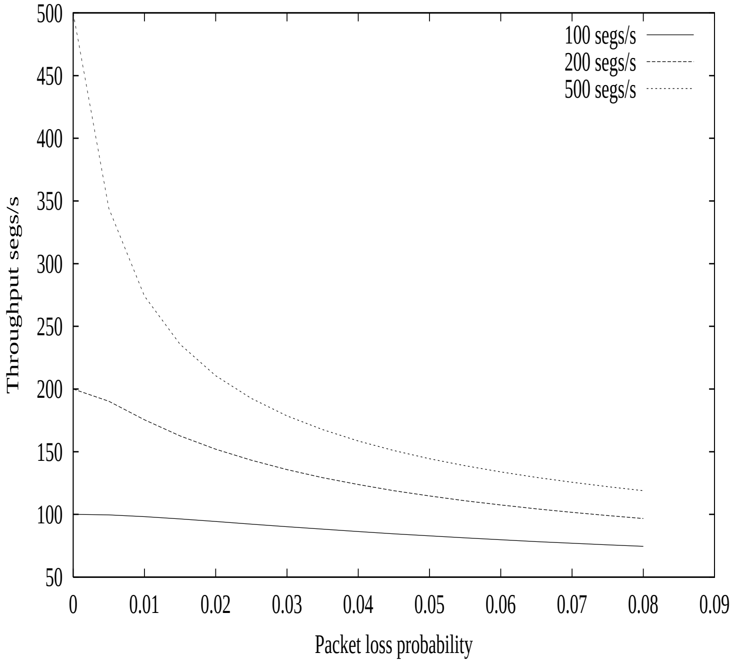

Figure 8.7: Expectation of T_{CA} for different sender's link capacities. $w_{max} = 70$ segments.

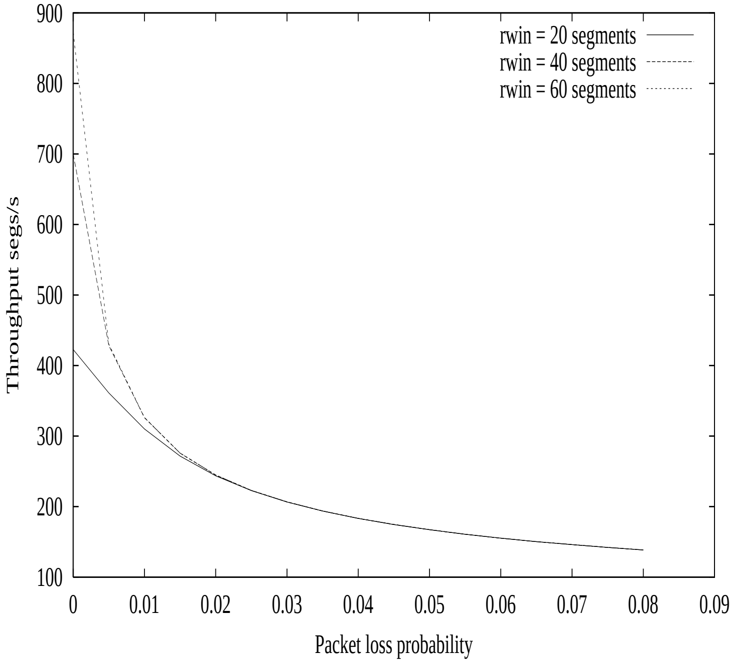

Figure 8.8: Expectation of T_{CA} for different w_{max}. Sender's link capacity is 1000 segs/s.

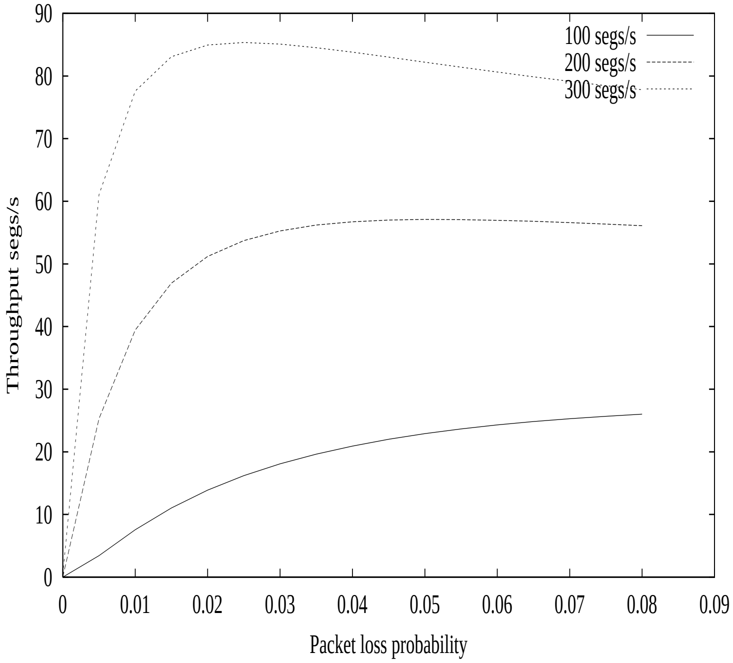

Figure 8.9: Standard deviation of T_{CA} for different sender's link capacities. $w_{max} = 50$ segments.

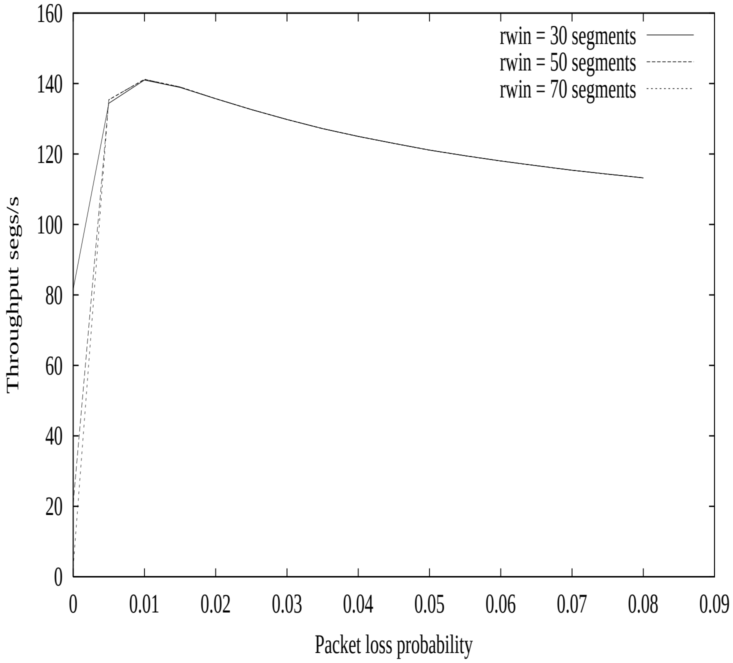

Figure 8.10: Standard deviation of T_{CA} for different w_{max}. Sender's link capacity is 500 segs/s.

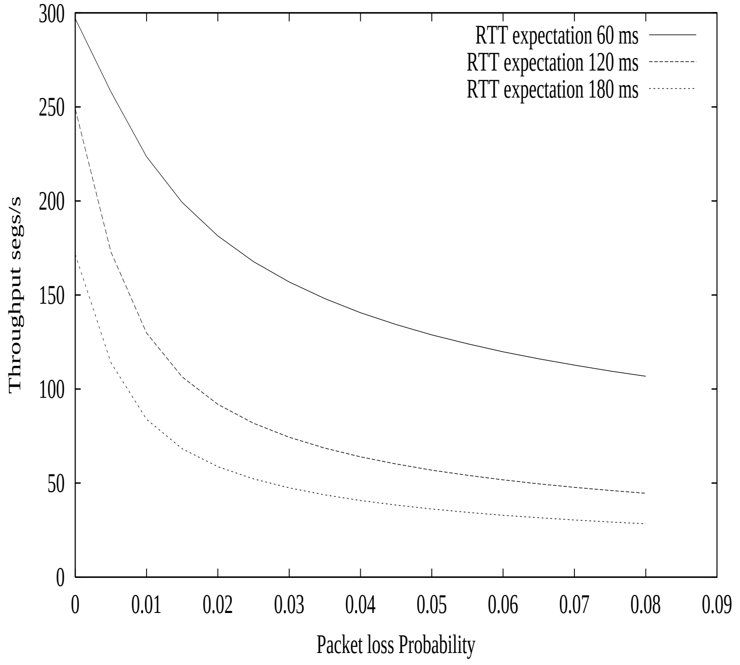

Figure 8.11: Expectation of T_{CA} for different RTT expectations. $w_{max} = 30$ segments, sender's link capacity 300 segs/s.

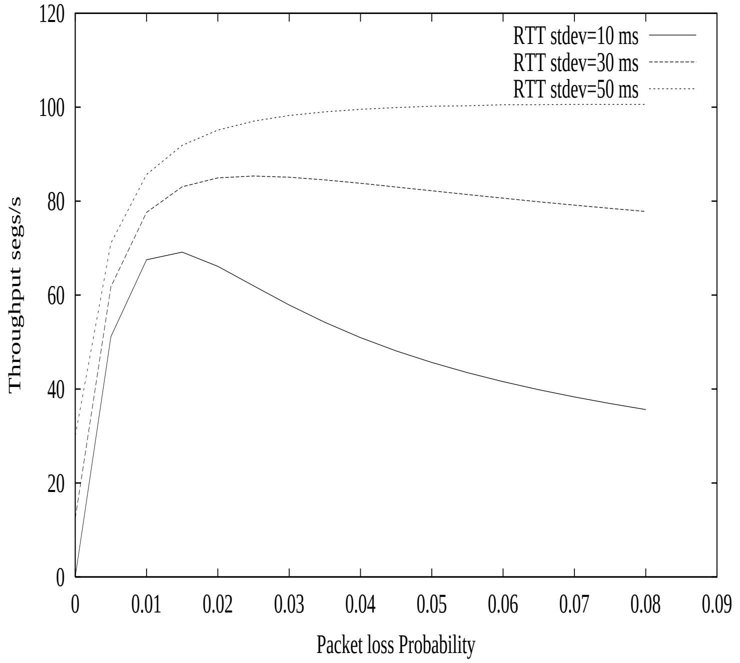

Figure 8.12: Standard deviation of T_{CA} for different RTT standard deviation. $w_{max} = 30$ segments, sender's link capacity 300 segs/s.

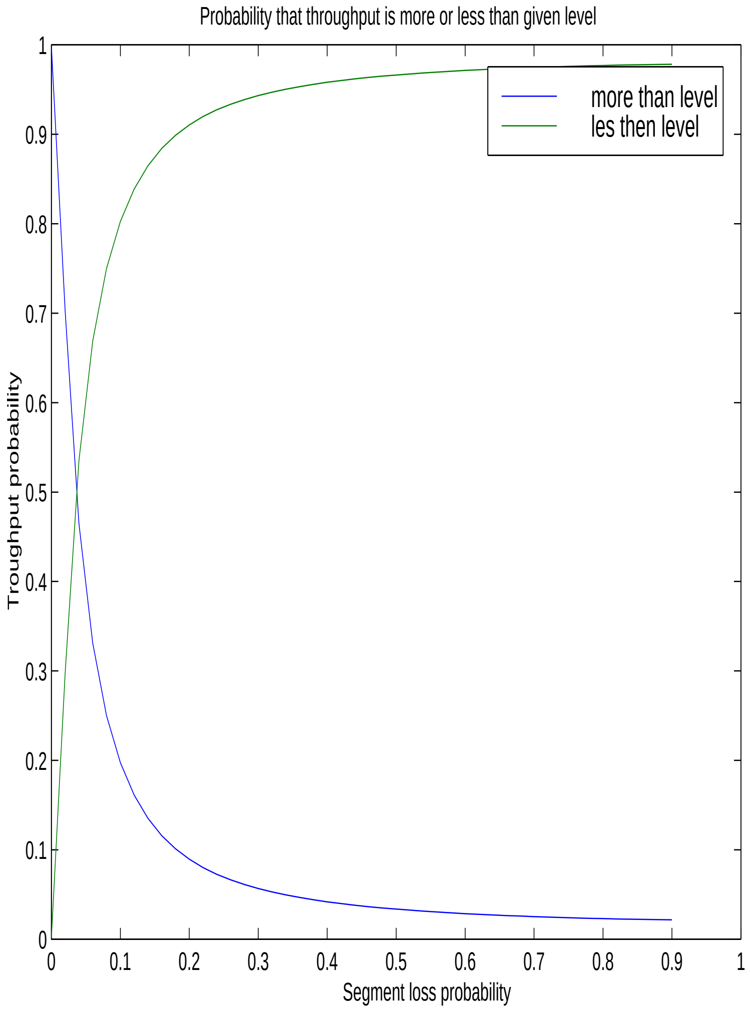

Figure 8.13: T_{CA} QoS. Level is 0.5

Appendix B. Validation results

Link rate 28800 bits/sec (**14.118** segments/sec).
Set 2. Packet transmission error probability 0.005.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
2.1	9.20 *19.99%*	9.49 *16.37%*	11.38 *2.98%*	11.04	13.65	13.37	segs/s
2.2	9.38 *22.29%*	9.69 *18.41%*	11.68 *1.72%*	11.48	13.61	13.34	segs/s
2.3	9.29 *23.09%*	9.58 *19.33%*	11.52 *0.74%*	11.43	13.49	13.22	segs/s

Table 8.1: Experimental and modeling T_{CA} (Absolute values and relative error).

Link rate 28800 bits/sec (**14.118** segments/sec).
Set 3. Packet transmission error probability 0.0025.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
3.1	11.89 *5.37%*	12.04 *6.53%*	13.63 *17.47%*	11.25	11.23	11.10	segs/s
3.1*	11.89 *2.12%*	12.04 *0.88%*	13.63 *10.93%*	12.14	11.23	11.10	segs/s
3.2	11.97 *0.27%*	12.06 *0.50%*	12.85 *6.62%*	12	**15.77**	**15.59**	segs/s
3.3	12.00 *0.14%*	12.10 *0.69%*	12.86 *6.52%*	12.02	**15.88**	**15.70**	segs/s

Table 8.2: Set 3. Experimental and modeling T_{CA} (Absolute values and relative error). Values exceeding the bandwidth capacity are marked by bold font.

Link rate 28800 bits/sec (**14.118** segments/sec).
Set 4. Packet transmission error probability 0.0005.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
4.1	12.04	12.10	12.92	11.35	8.46	8.39	segs/s
	5.74%	*6.22%*	*12.18%*				
4.1*	12.04	12.10	12.92	12.03	8.461	8.386	segs/s
	0.06%	*0.57%*	*6.89%*				
4.2	12.11	12.19	12.64	11.61	**15.9**	**15.77**	segs/s
	4.20%	*4.82%*	*8.14%*				
4.3	12.05	12.13	12.56	11.71	**16.01**	**15.89**	segs/s
	2.88%	*3.52%*	*6.78%*				

Table 8.3: Experimental and modeling T_{CA} (Absolute values and relative error). Values exceeding the bandwidth capacity are marked by bold font.

Link rate 0.5 Mbits/sec (**119.27** segments/sec).
Set 5. Packet transmission error probability 0.00025.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
5.1	52.09	52.16	54.38	61.54	65.92	65.66	segs/s
	18.15%	*17.97%*	*13.17%*				
5.2	52.29	52.39	54.27	64.32	75.55	75.28	segs/s
	23.01%	*22.78%*	*18.51%*				
5.3	52.68	52.79	54.6	62.75	77.95	77.70	segs/s
	19.12%	*18.89%*	*14.93%*				

Table 8.4: Experimental and modeling T_{CA} (Absolute values and relative error).

Link rate 0.5 Mbits/sec (**119.27** segments/sec).
Set 6. Packet transmission error probability 0.0005.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
6.1	51.85 *16.22%*	52.00 *15.89%*	54.20 *11.18%*	60.26	63.35	63.08	segs/s
6.2	50.30 *16.64%*	50.46 *16.27%*	53.16 *10.38%*	58.68	58.97	58.70	segs/s
6.3	49.90 *28.18%*	49.97 *27.98%*	52.64 *21.50%*	63.96	63.10	62.80	segs/s

Table 8.5: Experimental and modeling T_{CA} (Absolute values and relative error).

Link rate 0.5 Mbits/sec (**119.27** segments/sec).
Set 7. Packet transmission error probability 0.0025.

N	’Classical’	Sender’s	Throughput	Our model	Other models		
	throughput	throughput	without TO	ET	FF	PFTK	
	ATp	ATpXmit	NoTOTput				
7.1	32.57 *27.86%*	32.78 *27.06%*	36.52 *14.03%*	41.65	39.52	39.15	segs/s
7.2	34.43 *20.84%*	34.64 *20.10%*	39.71 *4.76%*	41.60	39.24	38.87	segs/s
7.3	30.85 *16.85%*	31.19 *15.59%*	35.49 *1.59%*	36.05	34.38	34.02	segs/s

Table 8.6: Experimental and modeling T_{CA} (Absolute values and relative error).

Link rate 0.5 Mbits/sec (**119.27** segments/sec).
Set 8. Packet transmission error probability 0.005.

N	'Classical' throughput ATp	Sender's throughput ATpXmit	Throughput without TO NoTOTput	Our model ET	Other models		
					FF	PFTK	
8.1	24.31 *29.04%*	24.58 *27.61%*	27.8 *12.82%*	31.36	30.30	29.93	segs/s
8.2	25.80 *26.88%*	26.08 *25.53%*	30.34 *7.88%*	32.73	31.43	31.06	segs/s
8.3	25.04 *31.17%*	25.30 *29.81%*	28.55 *15.05%*	32.84	31.70	31.33	segs/s

Table 8.7: Experimental and modeling T_{CA} (Absolute values and relative error).

Link rate 0.5 Mbits/sec (**119.27** segments/sec).
Set 9. Packet transmission error probability 0.025.

N	'Classical' throughput ATp	Sender's throughput ATpXmit	Throughput without TO NoTOTput	Our model ET	Other models		
					FF	PFTK	
9.1	12.02 *50.30%*	12.45 *45.08%*	15.30 *18.03%*	18.06	18.19	17.84	segs/s
9.2	11.97 *46.99%*	12.42 *41.60%*	14.90 *18.06%*	17.59	17.77	17.41	segs/s
9.3	12.08 *50.27%*	12.51 *45.11%*	15.64 *16.12%*	18.16	18.27	17.91	segs/s

Table 8.8: Experimental and modeling T_{CA} (Absolute values and relative error).

Appendix C. Evaluation of congestion window size

Let us consider the pair $\xi = (w, n)$, where w is a current cwnd size and n is the number of the segments sent under this cwnd since it has changed last time. For each w, $n = 1, \ldots, w$. Let us denote $w(t)$ and $n(t)$ the values that define ξ at the moment $t > 0$. Then according to our assumptions $\nu(t) = \{(w(t), n(t))\}_{t>0}$ is semi-Markovian random process (SMP).

Let τ_i be the time when ith TCP segment was sent. Note that here i is not a sequence number but is a unique natural number of each segment sent regardless of its content. The segments are numbered according to the following rule: $i_1 > i_2$ if $\tau_{i_1} > \tau_{i_2}$. Then sequence $\xi(\tau_0), \xi(\tau_1), \ldots, \xi(\tau_n), \ldots$ forms a Markovian chain embedded in the process $\nu(t)$. According to our assumptions the space of states of the chain $\xi_i = \xi(\tau_i)$ is finite as $w = 2 \ldots w_{max}$. Let us denote the space X. We also denote $p^k_{\xi\eta}$ the probability of the transition from state ξ into state η by k steps for any $\xi, \eta \in X$. Since set X is finite then

$$p^k_{\xi\eta} \xrightarrow[k\to\infty]{} \pi_\eta$$

and

$$\sum_{\xi\in X} \pi_\xi = 1$$

Let us denote $P_\xi(t)$ the probability of $\nu(t) = \xi$ and lets α_ξ be the expectation of the segments inter-departure time. The following theorem takes place

Theorem 1 *If RTT distribution has a finite expectation, then*

$$P_\xi(t) \xrightarrow[t\to\infty]{} \frac{\alpha_\xi \pi_\xi}{\sum\limits_{\xi\in X} \alpha_\xi \pi_\xi} \tag{8.1}$$

The proof is based on the Smith renewal theorem.

The presented theorem shows that the ergodic distribution of the process $\nu(t)$ is completely determined by the values α_ξ and π_ξ. Note that values π_ξ obviously are solution of the Kolmogorov equations for the Markov chain ξ_i. The equations are complicated. The immediate finding of their explicit solution is a rather difficult problem. Therefore we divide the problem into two parts. We consider the time moments when TCP assigns the new cwnd. (We remind the reader that only the congestion avoidance phase is considered here.) Thus, the chosen sequence of cwnd sizes ξ' also forms a Markovian chain whose partial fragment of the transition diagram is given in Section 4. The chain ξ'_i is embedded in the chain ξ_i. In fact the solution of the Kolmogorov equations for the chain ξ'_i gives us the distribution π'_ξ for all pairs $\xi' = (w, 1)$, i.e. $\pi'_\xi = \pi_w$. The distribution π_w is obtained by us in explicit analytical form. This simplifies the source Kolmogorov equations for the chain ξ_i and provides the π_ξ distribution.

Nevertheless the final closed forms are cumbersome and hence the calculation of π_ξ and especially the calculation of the sum in the right side of (8.1) may require significant efforts. Therefore we have obtained the recurrent presentation for the distribution π_w and on its base we have developed the numerical algorithm for the fast computation of π_w and π_ξ. The algorithm has linear complexity.

Note that random process $\mu(t) = \{w(t)\}_{t>0}$ also is SMP, and ξ'_i is a Markovian chain embedded in it. Let U_w be the duration of the round. Then

$$U_w = \begin{cases} RTT & \text{if} \quad wt_0 > RTT \\ wt_0 & \text{otherwise} \end{cases} \tag{8.2}$$

Let us denote β_w the expectation of U_w. Also let $P_w(t)$ be the probability of $\mu(t) = w$. Then the following theorem on the SMP $\mu(t)$ takes place.

Theorem 2 *If RTT distribution has a finite expectation, then*

$$P_w(t) \xrightarrow[t\to\infty]{} \frac{\beta_w \pi_w}{\sum\limits_{w=2}^{w_{max}} \beta_w \pi_w} \tag{8.3}$$

The values α_ξ might be derived from the two different assumptions:

1. α_ξ does not depends on $n \in \xi$;

2. α_ξ depends on $n \in \xi$.

Obviously in practice α_ξ always depends on $w \in \xi$. The first assumption simplifies analysis although it provides computational error as in the case obviously

$$\alpha_{w,n} = \alpha_w = \frac{\mathsf{E}(RTT)}{w} \tag{8.4}$$

The cwnd size is taken out of the expectation operator as it is fixed for any α_w. Nevertheless in practice the first assumption typically is not held by TCP instances but in many cases the final error caused by the assumption is negligible. If it is not, then for Bernoulli type schemes, accepting the second assumption does not lead to analytical difficulties in the derivation of α_ξ.

Let us denote $p_{w,n} = p_\xi = \lim\limits_{t\to\infty} P_\xi(t)$. The distribution of the TCP window size in the congestion avoidance has the following form

$$\omega_i = \sum_{j=1}^{i} p_{i,n} \tag{8.5}$$

This distribution is denoted ω_i in Section 5 and it is used for the construction of the numerical examples presented in this report. Obviously $\omega_i = \lim\limits_{t\to\infty} P_w(t)$.

Helsinki 2002