The Journal of Supercomputing, 20, 137–160, 2001

Towards More Complete Models of TCP Latency and Throughput

MICHAEL MITZENMACHER[*] michaelm@eecs.harvard.edu
Harvard University, Computer Science Department, 33 Oxford St., Cambridge, MA 02138

RAJMOHAN RAJARAMAN[†] rraj@ccs.neu.edu
College of Computer Science, Northeastern University, 161 Cullinane Hall, Boston, MA 02115

Abstract. Recently, several researchers have developed equations for modeling TCP behaviors, such as the expected throughput or latency, based on Markov chains derived from TCP with additional simplifying assumptions. In this paper, we suggest new directions for Markov chain analyses of TCP. Our first contribution is to closely examine not just the expectation but the entire cumulative distribution function of transfer times under various models. Particularly for short or medium transfers, the distribution is likely to be more useful than the expectation in terms of measuring end-user satisfaction. We find that the shapes of TCP cumulative distribution functions are remarkably robust to small changes in the model. Our results suggest that simplifying Markov analyses can be extended to yield approximations for the entire distribution as well as for the expectation.

Our second contribution is to consider *correction procedures* to enhance these models. A correction procedure is a rule of thumb that allows equations from one model to be used in other situations. As an example, several analyses use a Drop-Tail loss model. We determine correction procedures for the deviation between this model and other natural loss models based on simulations. The existence of a simple correction procedure in this instance suggests that the high-level behavior of TCP is robust against changes in the loss model.

Keywords: TCP latency, TCP throughput, Markov chains, Drop-Tail model

1. Introduction

Understanding and predicting TCP behavior remains a challenging problem, both because of the complexity of the protocol itself and the inherent complexity of the interactions between the protocol and the network.

Two important techniques have developed for understanding TCP behavior. The first is to use an event-driven simulation tool, such as `ns` [26], to simulate TCP behavior under preset conditions (e.g., [10, 13]). The `ns` simulator provides an infrastructure allowing realistic simulations of networks using TCP and other protocols, including aspects such as buffers with various queueing and drop policies, random delays corresponding to processing, and interaction among multiple flows. Data derived from such simulations can be studied to gain high-level insight into

[*] Supported in part by an Alfred P. Sloan Research Fellowship, NSF CAREER grant CCR-9983832, and an equipment grant from Compaq Computer Corporation.
[†] Supported in part by NSF CAREER award CCR-9983901.

TCP behavior. The simulation-based approach, however, does not provide an analytical and mathematical framework for studying TCP, making it difficult to extrapolate results or gauge the effect of changes.

Hence a second widely used approach is to study TCP by analyzing the event-driven process as a Markov process [7, 18, 20, 21, 24, 28]. This approach begins by developing a simplified model of TCP, with the goal of generating equations that describe the behavior of the model. For example, a natural goal is to derive an equation describing the functional relationship between the throughput rate, the round-trip time, and the loss probability; such relationships have been proposed as key features in designing other congestion control schemes that are fair to TCP traffic [6, 11]. Because of the inherent complexity of TCP and its environment, in order to derive a succinct equation, various simplifying assumptions are generally made to make the mathematics tractable. For example, a single stream is considered in isolation; all acknowledgments are assumed to arrive; losses or sequences of losses occur independently and randomly with some fixed probability; and packets are sent in groups over rounds.

In this paper, we suggest new directions for TCP analyses based on Markov processes. At this point, the work is primarily exploratory and based on simulations; we expect related mathematical analyses to follow in future work.

Our first direction is to expand the information sought from the TCP models. Thus far, the equation-based approach has primarily focused on finding expected throughput rates or transfer times. The expected transfer time may not be a reliable measure of important criteria such as end-user satisfaction, however. As a recent Fortune article (citing a Keynote systems study) states: “Everyone has a breaking point. For most Web surfers these days, it’s about eight seconds: If a page takes longer than that to load, most users won’t stick around” [8]. We therefore suggest that in order to compare properly various TCP models, it is imperative to study the *distribution* of the transfer time as well as the expectation, as different models might yield similar expectations but entirely different distributions. Hence, in our comparisons of various models, we primarily examine cumulative distribution curves.

One of our findings is that TCP distribution curves have robust shapes, in that in many cases varying the assumptions does not significantly change the overall shape of the distribution curve. We attempt to offer insight into why this is the case. We believe these results suggest that the equation-based models derived thus far will, with further work, extend to provide reasonable approximations of the entire distribution curve.

Our second related direction is to consider *correction procedures* that allow us to better understand the effect of various simplifying assumptions made for analysis. A potential problem with previous work based on the equation-based approach is that several simplifying assumptions are made, but their individual effects are not examined. Instead, the results of the end equation are tested against simulations. In this framework it is difficult to tell whether all the simplifying assumptions have a small effect, or whether the effects of various simplifications tend to cancel each other out.

Our goal is to consider the deviations introduced by the various simplifications in isolation to quantify what sort of errors they may cause. We emphasize that the purpose of this exercise is not to diminish the validity of the approach of determining TCP equations. Rather, we hope to enhance the TCP equations by recognizing what kinds of corrections may apply. Correction procedures can help us in several ways. For example, if we know that one simplifying assumption tends to increase the transfer time, while another tends to decrease the transfer time, we can take advantage of the fact that the effects tend to cancel each other. As a more specific example, we consider how the transfer time varies with the percentage of lost acknowledgments. This insight may allow us to use more faithfully a model where acknowledgments are not lost to predict behavior when acknowledgments are lost. In general, we find that there can be non-trivial differences between transfer rates for different models, suggesting that correction procedures or some other mechanisms are necessary to have accurate estimates of TCP behavior.

The idea of a correction procedure is also useful in understanding the complexity of TCP behavior. A simple correction procedure suggests that there is a feature of TCP that is robust to changes in the model. For example, we show that there appears to be a natural correction procedure among different loss models. We believe this should direct future equation-based work toward a universal analysis that holds for a variety of loss models. In contrast, the lack of a simple correction procedure between models would suggest a complex interaction between the TCP protocol and the underlying model that would need to be understood.

The remainder of this paper is organized as follows. Section 2 reviews previous work on TCP dynamics. Section 3 describes the loss models that we evaluate in our simulations. Section 4 presents the simulation environments used in our study. Section 5 presents our main results concerning the impact of different loss models on TCP performance. Section 6 considers the effect of lost acknowledgments. Section 7 studies the variance of transfer times for large bulk transfers. Finally, Section 8 presents concluding remarks and directions for future research.

2. Previous work

A significant amount of the traffic on the Internet currently uses TCP as the transport protocol. Even for applications for which TCP is not the transport protocol of choice, such as multicast and continuous media delivery, there is an increasing trend toward designing TCP-friendly transport protocols. Consequently, several simulation and analytical studies have been conducted to understand the start-up dynamics and the steady-state behavior of TCP bulk transfer and to quantify the TCP-friendliness of other transport protocols.

Earlier studies of TCP include the analysis of the basic congestion avoidance and control algorithms [9, 14] and simulations and trace-based analyses that detect phenomena such as ACK-compression, out-of-packet delivery, synchronization of losses, and pathological connections [17, 19, 30]. Analytical models developed in [15, 16] study the long-term behavior of a TCP connection. Consequently, they do not attempt to capture the impact of bursty losses, timeouts, slow start, and other

TCP characteristics. The start-up dynamics of TCP Reno are studied in [13], which also suggests changes to the implementation to improve its performance during the start-up epoch. The simulation-based study of [10] compares a number of different TCP implementations with respect to their response to multiple packet drops in a single window. One of the main results of the preceding study is that all of the most common TCP implementations that use cumulative acknowledgments react poorly to multiple packet drops in a single window since the TCP sender frequently incurs a timeout even if as few as two packets are dropped.

The significant difference in the effects of the two loss indications used by TCP, namely timeouts and triple duplicates, is a focus of [20, 21], which present a stochastic model for TCP congestion control and derive formulae for the expected steady-state throughput in terms of latency and packet loss. This model is further extended in [7] and [24] to include startup effects such as connection establishment and slow start, which have a significant impact on the latencies of short TCP transfers. All of these analytical studies [7, 21, 22, 24] adopt the Drop-Tail packet loss model described in Section 3. Recently, different models of packet loss have been proposed. One such model is presented in [18], which discards the "source-centric" model of parametrizing the individual packet loss probabilities and instead models the loss indications *received by* the source as a Poisson stream. More closely related to our work is the recent study in [28], which analyzes unicast and multicast packet loss measurements and evaluates the accuracy of multiple state Markov chain models for packet loss.

Models for analyzing TCP throughput that consider correlated packet losses have been recently studied in [2] and [31]. The correlated losses are represented by a Markovian process in [31], while [2] adopts a stationary ergodic random process. The emphasis of both studies, however, is on bounding the *average* throughput in terms of parameters characterizing the packet loss process; in contrast, our focus is on evaluating the impact of different loss models on the *distribution* of download times.

The assumptions in our model (and those made in earlier studies of TCP) have been influenced by a number of studies based on large-scale Internet measurements [5, 23, 25, 30]. For example, a key observation made in these studies is that packet losses are correlated. The Correlated model, described in Section 3, is largely motivated by the observations of [23] and [30]. In [30], it is argued that packet losses can be modeled by Bernoulli distribution of *loss episodes*, in which each loss episode is a sequence of consecutive losses, the length of which is drawn from a geometric distribution.

3. The loss model

The selection of a loss model is a key question in designing simplified models of TCP performance. In this section, we examine the most common loss models.

In all of the loss models we study, we make the assumption that packets and acknowledgments are sent in groups over rounds, and that losses are independent from round to round. This assumption, which is made in most analytical studies

[7, 22], is partially justifiable with the understanding that TCP tends to send packets in bursts in a manner similar to how our models send packets in rounds, and the round-trip time between rounds may be sufficient for most congestion to clear. The independence of packet losses occurring in different rounds is especially likely to hold for connections with moderate to high round-trip times since the time needed to send all the packets in a window is then much smaller than the round-trip time [2, 22]. We note that this assumption is not essential for our modeling approach; we adopt it for testing purposes in keeping with the main point of our study, which is to examine variations from the simplified Markov models studied thus far.

We focus on the following models:

— *Bernoulli*: Each data packet is independently lost with a fixed probability p.

— *Drop-Tail*: In each round, we consider the data packets sequentially. The first packet in the round is lost with probability p; for every other packet, if the previous packet was not lost, the packet is lost with probability p; if a packet is lost, then all subsequent packets in the round are lost.

— *Correlated*: In each round, we consider the data packets sequentially. The first packet in the round is lost with probability p; for every other packet, if the previous packet was not lost, the packet is lost with probability p; otherwise, it is lost with probability q.

The Bernoulli model is arguably the most basic model for packet loss. Owing to its simplicity, it lends an easier analysis than the other loss models. The Bernoulli model may be appropriate for modeling congestion arising in queues that implement the random early detection (RED) policy [11], since such queues respond to congestion by dropping packets uniformly at random. The Drop-Tail model is an idealization of the packet loss dynamics associated with a FIFO drop-tail queue. It is assumed in this model that during congestion, queues drop packets in bursts [4, 23], thus causing packets in the "tail" of a round to be lost. The Correlated model is somewhat less stringent. It characterizes the loss pattern as a Bernoulli distribution of loss episodes, each episode consisting of a group of consecutive packets, the length of which is approximated by a geometric distribution. Recent evidence for such correlated packet loss includes [23, 30]. We note that the Correlated model actually includes both the Bernoulli case (when $q = p$) and the Drop-Tail model ($q = 1$) as extreme cases. For each of the above models, we refer to p as the *loss episode parameter*.

4. Experimental setup

We used two simulation environments for our study: one a *round-based* TCP simulator that we have written, and the other the UCB/LBL/VINT simulator `ns`. The notion of a *round*, which is intrinsic to the Markov chain approach and is not captured in `ns`, defines the unit of time in the round-based simulator. In our simulator, a round is broken into sequential phases: the sender sends the round's worth of packets; packets are passed through a filter that may introduce loss; the receiver receives packets, and sends appropriate acknowledgments; acknowledgments are passed through a filter that may introduce loss; and the sender receives acknowledgments. The round-based simulator faithfully captures the analytic models proposed in [7, 21, 22]. Consequently, we are able to directly compare the different cumulative distribution functions obtained over a wide range of loss models. Furthermore, the simplicity of the round-based simulator allows us to vary parameters and models for testing with relative ease.

We use the round-based simulator to study the dynamics of file transfers, primarily focusing on transfers of 64 and 1024 packets from a TCP Reno source under various loss models. The two transfer sizes chosen are representative of short-to-medium size downloads. In our simulation, the receiver issues delayed acknowledgments; it sends an acknowledgment for every other packet or in the next round, whichever occurs earlier. We omit the effects of round-trip time smoothing calculations and the connection establishment phase. All experiments conducted with the round-based simulator involved 10,000 trials for each setting of the variables. In all our experiments, we assume that the maximum window and the initial slow start threshold are 24 and 42 packets, respectively.

We validate our round-based simulator by comparing the distributions obtained for the Bernoulli loss model with those obtained in `ns`. For this purpose, we modify the `ns` simulation to set the duration of the first timeout in any sequence of consecutive timeouts to match the corresponding value set in the round-based simulator, which is 4 times the round-trip time. (We note that the exponential backoff protocol used for setting the retransmission timers in the event of a sequence of consecutive timeouts is the same in both round-based and ns TCP simulations.) The `ns` experiments simulate a TCP `Reno` sender and a `DelAckSink` receiver, which sends an ack for every other packet or when a 100 ms timer expires, whichever occurs earlier. Our data for the `ns` simulations are based on 1,000 trials. Figures 1 and 2 compare the cumulative distribution plots for the number of rounds needed to transfer 64 packets and 1024 packets, respectively, under different Bernoulli packet loss probabilities. For the data obtained from `ns` simulations, the y-axis represents the ratio of the transfer time to the round-trip time. We note that the relative error between the two plots is less than 3%. Figure 3 gives the cumulative distributions of timeouts for 1024 packet transfers, which also match very closely.

The small discrepancy between the round-based and `ns` distributions in the Bernoulli model is due to some subtle differences in the two simulations. We discuss one such difference here. During the fast recovery phase in TCP Reno, a sender artificially inflates the congestion window in response to duplicate acknowledgments to account for the fact that packets have left the network. In the round-based simulator, if the TCP sender is in fast recovery and incurs a timeout, then the slow-start threshold (`ssthresh`) is set to half the *inflated* congestion window (`cwnd`); in this we have followed Wright and Stevens [27]. On the other hand, in the `ns` version, the threshold is set to half the *uninflated* congestion window, which corresponds to half of the congestion window at the instant that the fast recovery phase was initiated. This appears to follow RFC 2581 [1]. Discrepancies also arise due to minor differences in the particular implementation of delayed acknowledgments. (As an aside, we note that seemingly minor differences can have

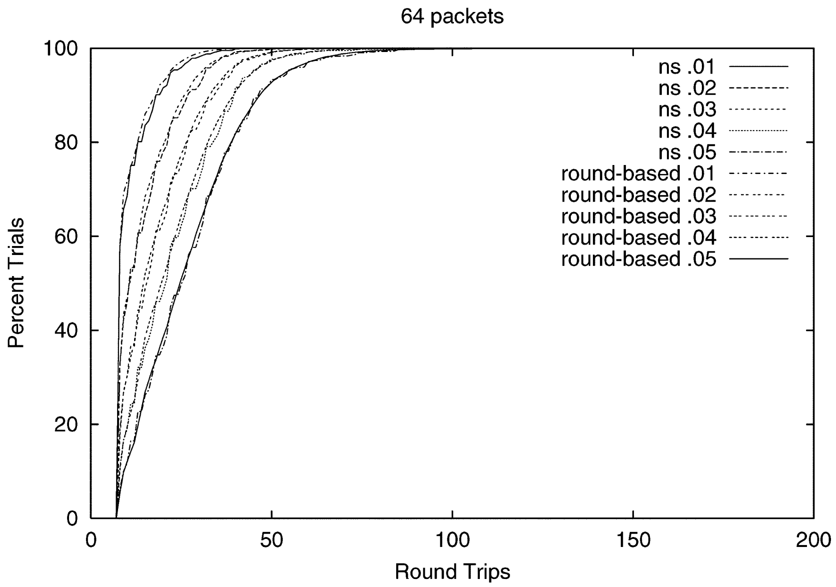

Figure 1. Comparison between ns and round-based simulations for the Bernoulli loss model with respect to rounds; 64 packets, packet loss 1–5%.

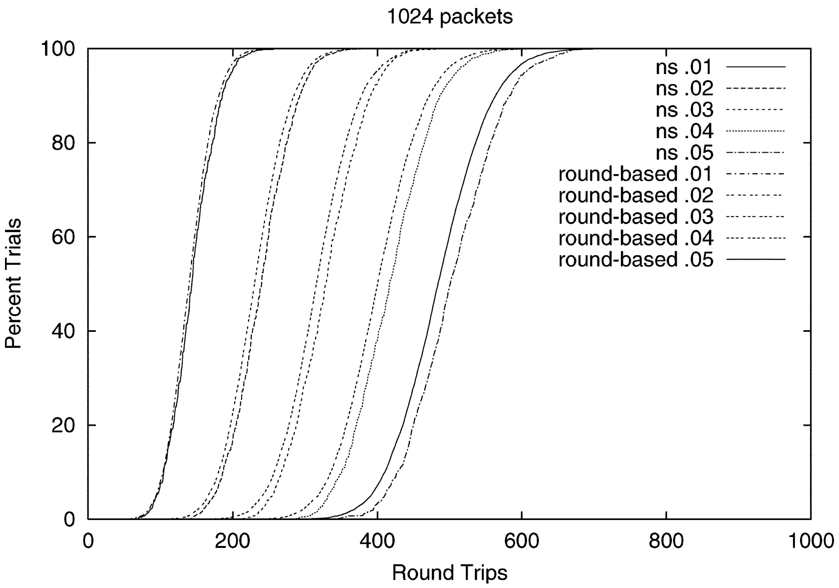

Figure 2. Comparison between ns and round-based simulations for the Bernoulli loss model with respect to rounds; 1024 packets, packet loss 1–5%.

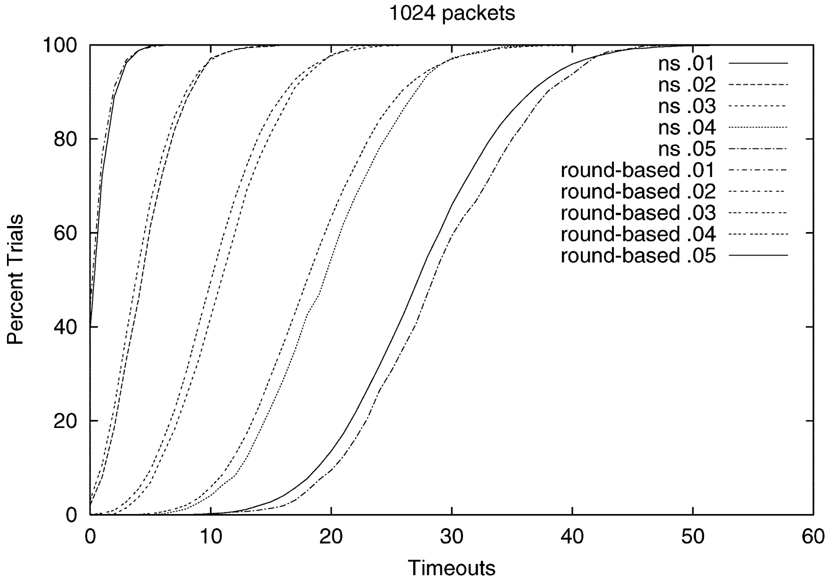

Figure 3. Comparison between ns and round-based simulations for the Bernoulli loss model with respect to timeouts; 1024 packets, packet loss 1–5%.

a significant impact. For example, in [1], it is stated that when deciding whether to use slow start or congestion avoidance, the case where cwnd equals ssthresh is an ambiguous case. We have found that the specific choice makes a small but clearly noticeable difference.)

A similar validation for the Correlated and Drop-Tail models poses problems. A plausible approach to implement these models in ns is to use a variant of the two-state Markov chain, that is, the TwoState link error model offered in ns. The two states in this model represent the error and error-free states. The parameters of the model are the duration of time that a link spends in a particular state before switching to the other state. A significant drawback of the model thus described is that it is oblivious to the notion of rounds, and consequently, introduces dependencies between losses across rounds. Independence of packet loss across rounds is an important assumption made in previous models [7, 20, 21], and has also been observed in measurement studies [4]. To address the above problem, we modified the ns implementation to introduce the notion of a round by maintaining a round number that is incremented whenever the time difference between the sending of two consecutive packets exceeds the current smoothed round-trip time value. With this we can approximately capture the behavior of the Correlated and Drop-Tail models, although there is variance in the round-trip time as measured by the TCP sender because of delayed acknowledgments. A comparison of the cumulative distribution plots of the number of rounds for transfers of 64 and 1024 packets is given in Figures 4 and 5, respectively.

To summarize, we have developed our own round-based implementation of TCP, which we use in all further experiments. The primary reason to prefer our

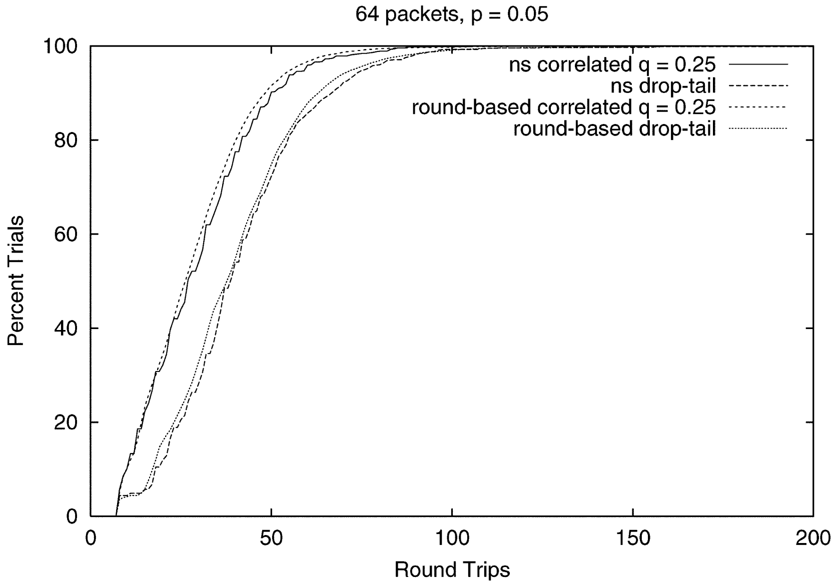

Figure 4. Comparison between ns and round-based simulations for the Correlated and Drop-Tail models with respect to rounds, 64 packets.

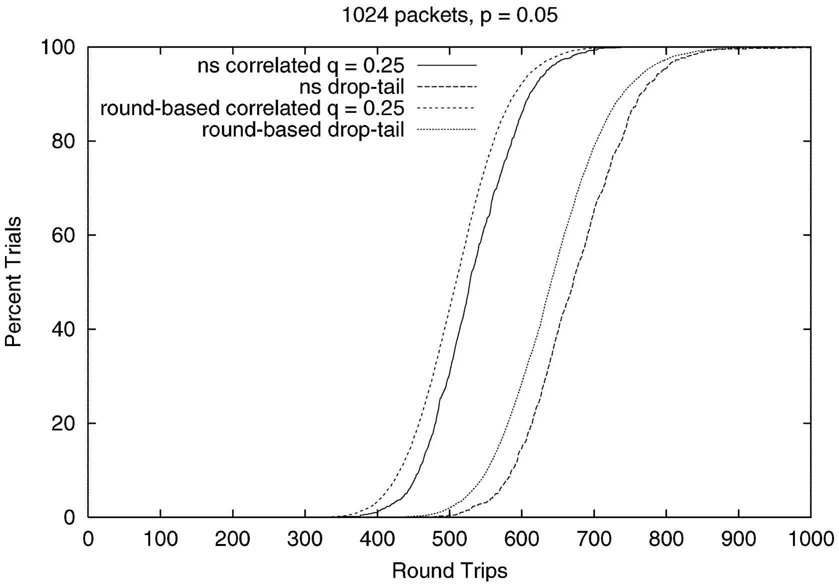

Figure 5. Comparison between ns and round-based simulations for the Correlated and Drop-Tail models with respect to rounds, 1024 packets.

implementation over ns is that our purpose is to study round-based Markov models. Our implementation is more faithful to these Markov models, and offers us more control and flexibility in experimenting with variations of these models. Our implementations differ a small amount but not significantly from ns. It is worth noting that the resolution of seemingly small ambiguities in the TCP protocol can lead to noticeable deviations in TCP performance, highlighting the difficulty of accurately modeling TCP.

5. Analysis of the loss models

In this section we examine the impact of the loss models discussed in Section 3 on the cumulative distributions of transfer times. In keeping with our goal of isolating the effects of various assumptions, we assume in this section that there are no losses of acknowledgment packets and focus on the effect of lost data packets.

We begin by considering the graphs showing the behavior of the cumulative distribution function when $p = 0.05$ for all three of the models, namely, Bernoulli, Correlated, and Drop-Tail; this is representative of the behavior of loss rates between 1 and 10%. (We focus on this range of loss rates, as it is most likely to lead to interesting behavior, and it appears representative in practice [23].) For the Correlated model, we show the behavior for $q = 0.25, 0.5$ and 0.75. To the first order (barring boundary effects), this equalizes the number of loss episodes seen by all three models. As one might expect, the greater the correlation, the greater the time; this seems natural since the same number of loss episodes leads to more overall losses and timeouts when there is correlation. (See Figures 6 and 7.) However, it is interesting to note that although the expected transfer time changes with the models, the distributions all have the same approximate shape.

We now consider a correction procedure for the relationship of the expected transfer time among the various models over the range of p from 1 to 10%. We have determined the ratio between the average transfer time for each model with the Bernoulli model for these loss probabilities. We call this ratio the *correction factor*. The correction factor provides a way of translating the results from one loss model to another. For example, using the equations determined in [20] for the expected goodput of the Drop-Tail model, we can use the correction factors to estimate the expected goodput for other models. The correction factors are charted in Figure 8. We note that the differences between the models are fairly significant, especially between the Bernoulli and Drop-Tail models, where the difference ranges from roughly 30% to 50% in this range of p. This fact suggests that understanding which loss model is appropriate and relevant is important for accurately predicting TCP behavior. Unfortunately, the correction factor does not appear to have a simple functional form; currently we can only estimate the correction factor through experimentation.

Interestingly, although we determined the correction factor simply from the average transfer time, the correction factor appears closely tied to the entire cumulative transfer distributions. Indeed, if we consider the distribution curve for 1024 packets and loss episode parameter $p = 0.05$ from Figure 7, but rescale all the transfer

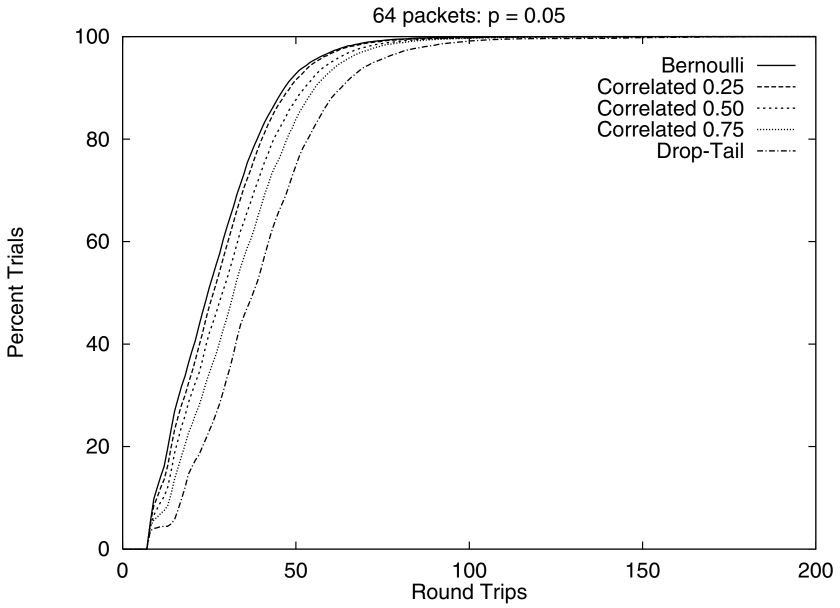

Figure 6. Cumulative distribution function for various loss models; 64 packets.

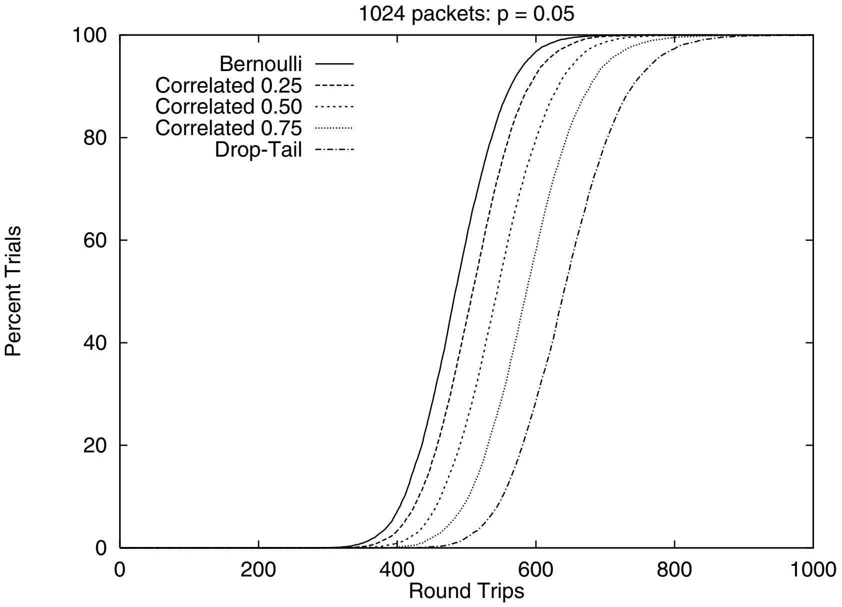

Figure 7. Cumulative distribution function for various loss models; 1024 packets.

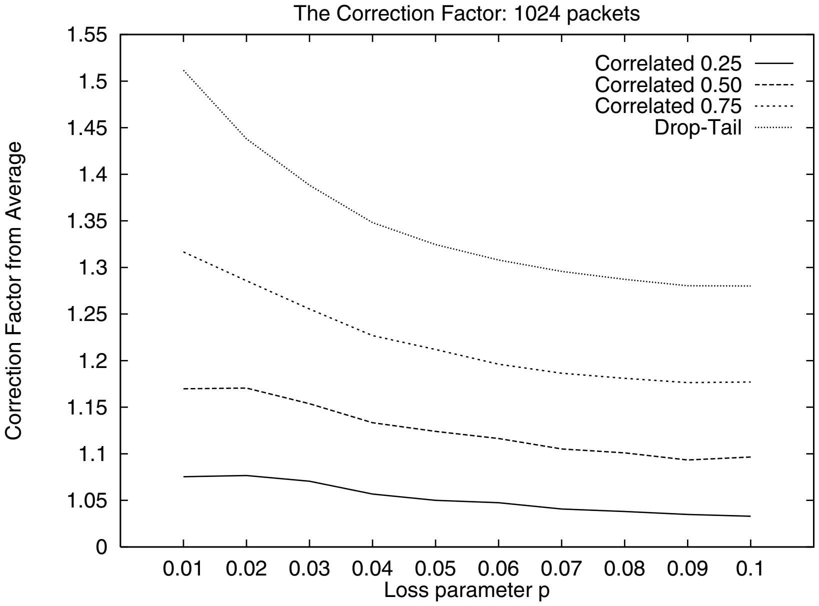

Figure 8. The correction factor versus p.

times downward for all of the models (besides the Bernoulli model) according to the appropriate correction factor, then the distributions themselves are remarkably close, as seen in Figures 9, 10, and 11. Hence our initial determination that the distribution curves have the same shape has now taken a concrete form: the difference between pairs of models can be succinctly summarized by a single number, the correction factor.

We attempt to explain this observation. Our data suggest (not surprisingly) that the transfer times are highly correlated with the number of timeouts in a linear relationship for all of the loss models. For example, we examined the correlation coefficient between the number of timeouts and the transfer time. For the Bernoulli model, the correlation coefficients for 1024 packets are 0.81, 0.94, and 0.90 with loss episode parameters 0.2, 0.5, and 0.8, respectively. For the Drop-Tail model, the corresponding numbers are 0.96, 0.94, and 0.81 respectively. These numbers remain relatively constant as the number of packets varies. Hence, for a fixed loss episode parameter, the transfer time is roughly a linear function in the number of timeouts. In all of the models, the number of timeouts has a nearly normal distribution, as seen in Figure 12, suggesting that each loss episode has some near-fixed probability of leading to a timeout. Hence it is reasonable for all of the loss models to have the same shape distribution, and for these distributions to scale.

An interesting line for future research is to try to approximate timeout characteristics with an appropriate normal model; this would allow a simplified Markov model that be used to derive not only approximate expected transfer times under TCP, but also approximations for the full distribution. Indeed, although we are working with simplified models, the utility of understanding the variance is

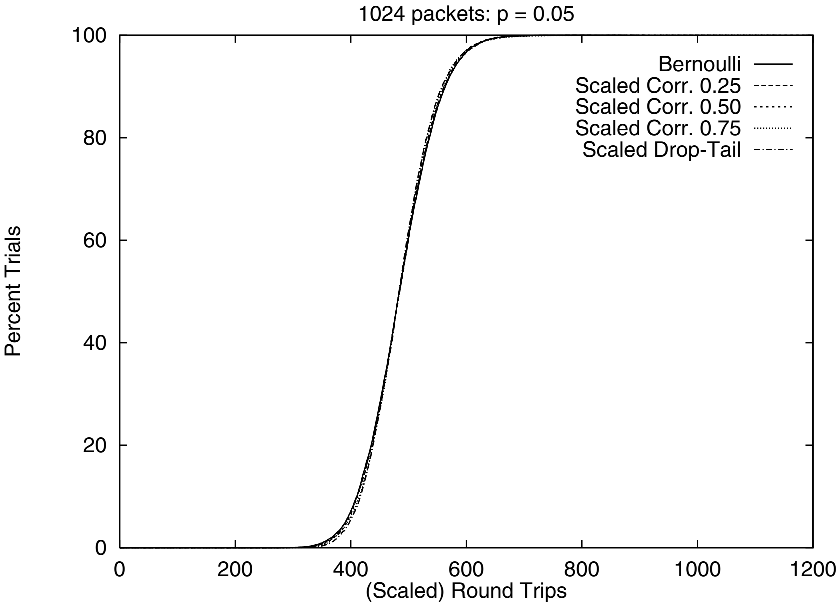

Figure 9. The correction factor applied to the cumulative distribution; 1024 packets, $p = 0.05$.

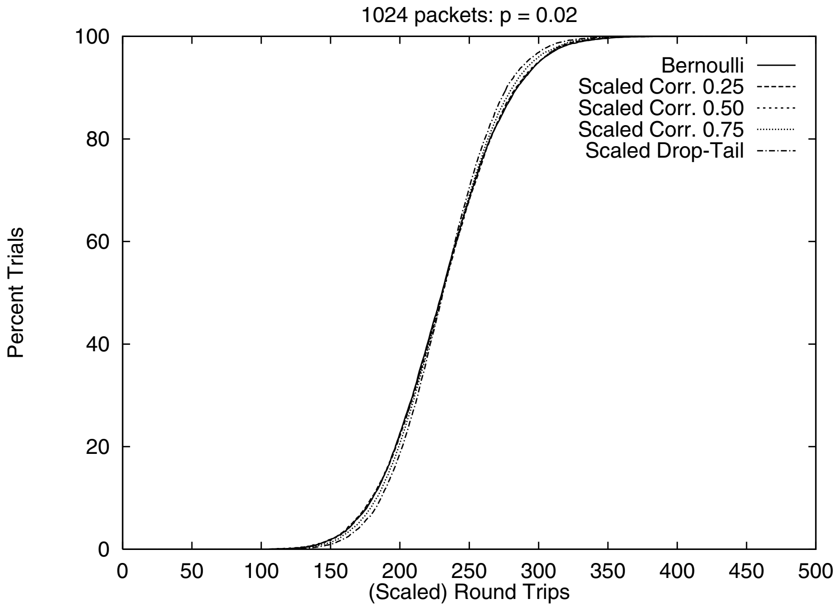

Figure 10. The correction factor applied to the cumulative distribution; 1024 packets, $p = 0.02$.

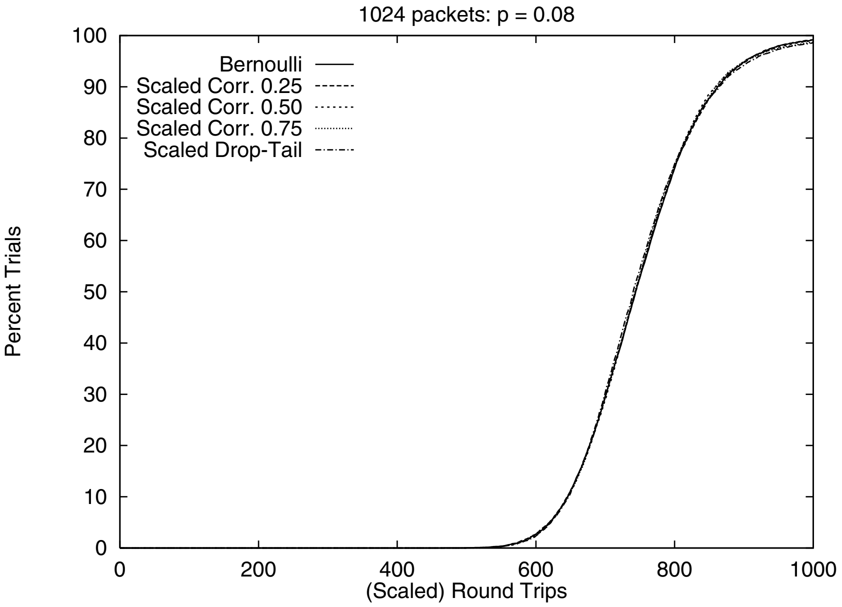

Figure 11. The correction factor applied to the cumulative distribution; 1024 packets, $p = 0.08$.

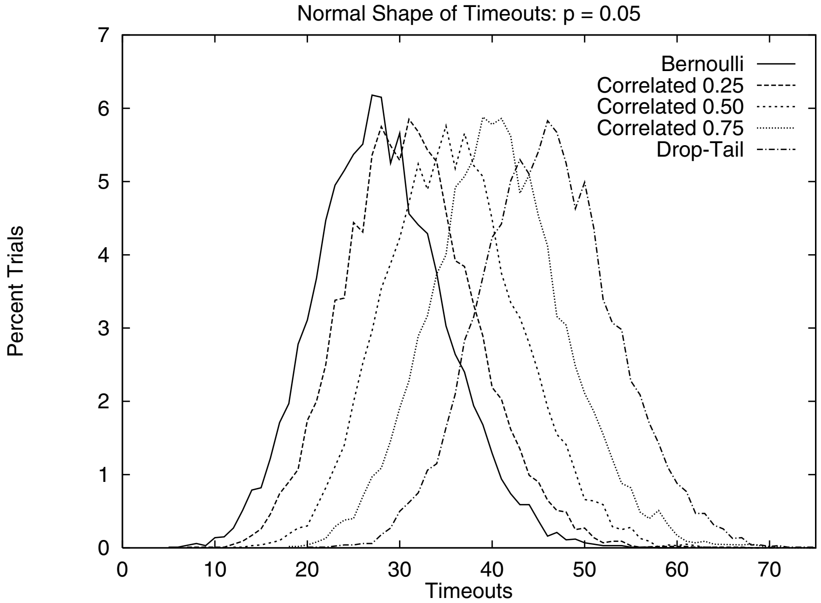

Figure 12. Normal distributions for timeouts.

highlighted by recent work by Barford and Crovella, who suggest that for short and medium downloads, the variability of timeouts is the primary cause of variability in transfer time [3].

Also, we believe further understanding of the correction factors would be very useful, as they relate the Drop-Tail model and the perhaps more realistic Correlated model. We note some caveats. Our experiments suggest that the correction factor must be increased for shorter transfers, and slightly decreased for larger transfers. This behavior is interesting, since it suggests that the convergence of the throughput rate to its expectation may require significantly long transfers; we explore this further in Section 7 where we return to the question of variance. Also, these correction factors depend on parameter choices in the model; when delayed acknowledgments are not used, for instance, the appropriate correction factors increase substantially.

Because starting with the same loss episode parameter p leads to more losses when losses are correlated, it may seem more fair to attempt to equalize the models for the same overall fraction of lost packets, instead of equalizing for loss episodes. As shown in Figure 13, which compares the Drop-Tail model with the Bernoulli model, such accounting dramatically punishes the Bernoulli model. When $p = 0.02$ for the Drop-Tail model, for example, the total fraction of lost packets is 7.02%; for the correlated model with $p = 0.05$ and $q = 0.5$, the total fraction of lost packets is about 6.91%. Comparing the cumulative distribution curves for transfer times for these two situations with the Bernoulli model with $p = 0.07$, we see that the correlated models appear much better when we equate the packet loss probability. The explanation is that TCP Reno handles very well long sequences of losses such as those that occur with the Drop-Tail model; a contiguous sequence

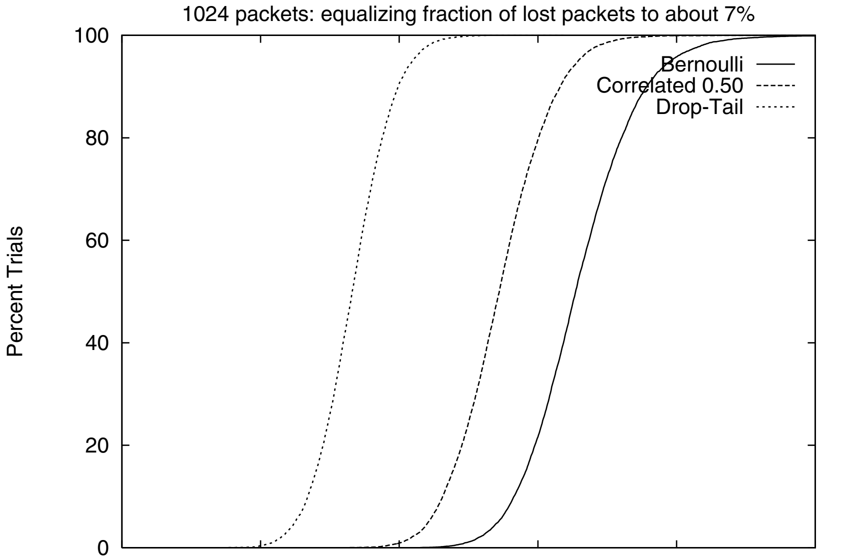

Figure 13. Cumulative distributions when overall packet loss percentages are equalized.

of losses occurring within a single window tends to cause a single retransmission timeout, after which all the lost packets are resent. The same number of randomly distributed losses, however, is likely to spread over multiple windows and may cause multiple timeouts.

Our conclusion is that the choice of loss model has a significant effect on the expected transfer time and throughput rate. If we equalize the models so that a loss episode begins at any point according to the same loss episode parameter p, models with correlated losses have noticeably longer transfer times. The shapes of the corresponding cumulative distribution functions, however, are quite similar. In particular, we have shown that a correction factor based on the expected transfer time appears to correct for the differences between the entire distribution. We believe this suggests a richer model yielding approximate distributions for certain loss models will be possible.

6. Lost acknowledgments

In most analyses of TCP based on Markov processes, acknowledgments are assumed to arrive so that effects of lost acknowledgments can be ignored in the analysis. In this section we attempt to determine the effect of this assumption. We focus here on the model with Bernoulli data packet and acknowledgment losses. More highly correlated losses will have similar effects, depending on the loss parameters. (The same loss episode parameter p will lead to more severe effects with more correlated models, since a larger number of packets will be lost.)

We begin by examining the cumulative distribution curves for transfers of 64 and 1024 packets and a 5% data packet loss rate with varying acknowledgment loss rates in Figures 14 and 15. Although the distribution curves keep the same overall shape, the effect of acknowledgment losses is to noticeably slow the transfer. The difference between no acknowledgment loss and a 5% acknowledgment loss is a 21% increase in the average time for 64 packet transfers and a 19% increase in the average time for 1024 packet transfers. Since smaller downloads spend proportionally more time in slow start, it is reasonable that acknowledgment losses would have a slightly larger effect.

It is interesting to determine how the lost acknowledgments affect performance, since they affect the system in multiple ways. In some cases, a lost acknowledgment can directly lead to a timeout in an instance where the arrival of the acknowledgment might have led to a normal continuation or a fast retransmit. For example, if the first pair of packets sent are received but the acknowledgment is lost, the sender will continue to wait for an acknowledgment until a timeout occurs. Lost acknowledgments also slow down the overall rate at which the sending window size increases. Note that this effect may indirectly lead to additional timeouts, as a smaller window may preclude a fast retransmit for a lost packet later on in the process. To see the impact of each of these effects, we modified our TCP simulation so that when an acknowledgment was lost, the simulation would act as though the acknowledgment arrived, except that it would not increase the cwnd parameter. That is, we removed the possibility of a lost acknowledgment directly causing

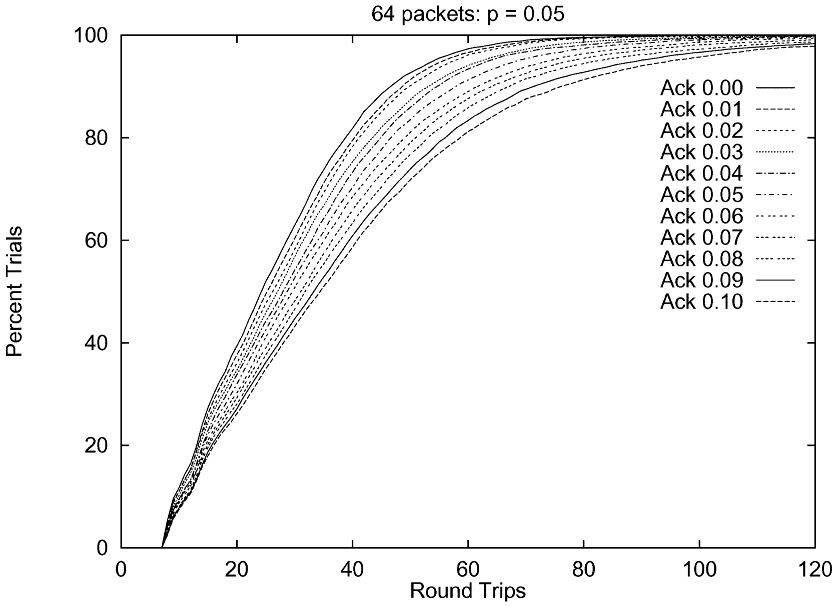

Figure 14. Cumulative distribution functions as lost acknowledgments increase: 64 packets.

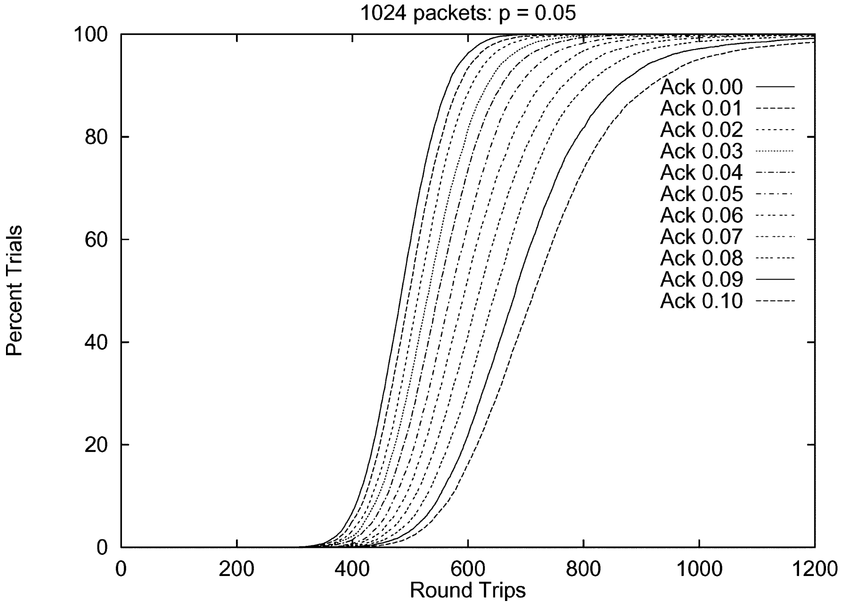

Figure 15. Cumulative distribution functions as lost acknowledgments increase: 1024 packets.

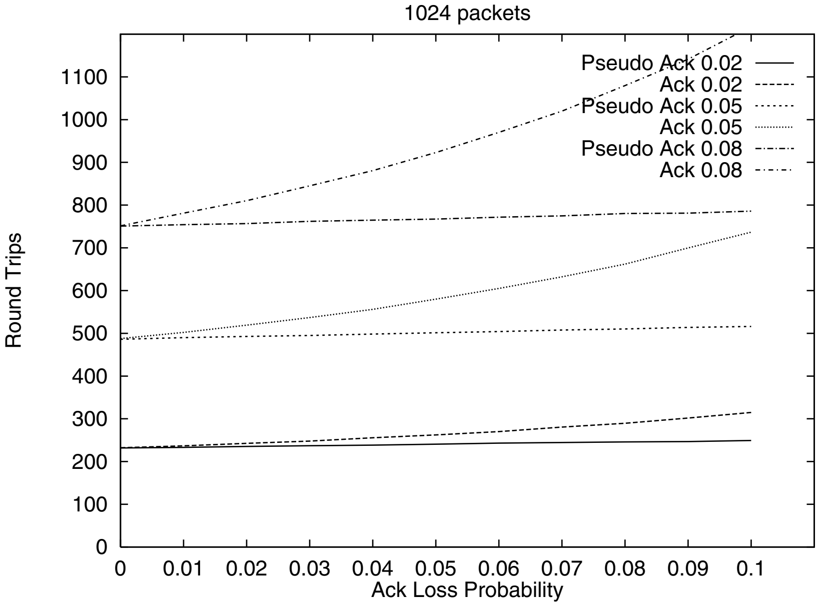

Figure 16. The effects of lost acknowledgments versus pseudo-ack loss.

a timeout, while keeping the failure to increase the sending window. We call this a *pseudo-ack loss*.

In Figure 16, we compare the increase in the average transfer time from acknowledgment loss for lost acknowledgments and pseudo-ack loss. We show results for data packet loss probabilities $p = 0.02, 0.05$, and 0.08. As can be seen, pseudo-ack loss causes only a small increase in the transfer time, suggesting that the important effect of lost acknowledgments is to increase the number of timeouts. The relative importance of this effect grows with the packet loss probability. Again, this offers some corroboration of the suggestion of [3] that timeouts are the significant cause of variability in short and medium downloads.

We again attempt a correction procedure for varying acknowledgment losses by using a correction factor. Consider a fixed loss event parameter $p = 0.05$ and a fixed number of packets in the message, 1024. We determined the ratio between the average transfer times for various acknowledgment loss probabilities, and then scaled the entire distribution curve by this factor. The result appears in Figure 17. Although the resulting distribution curves do not completely overlap, they are very close; we see that larger acknowledgment losses lead to more extreme tails than scaling would suggest. This suggests higher acknowledgment loss probabilities have a greater than linear effect on the overall transfer time. A suitable correction procedure would need to take this into account. From a more theoretical standpoint, this suggests that a richer equation-based model that accurately accounted for lost acknowledgments would need to include a non-linear term in the acknowledgment loss probability. However, as a first approximation, the cumulative distribution for no acknowledgment loss along with a correction factor determined by the expectations

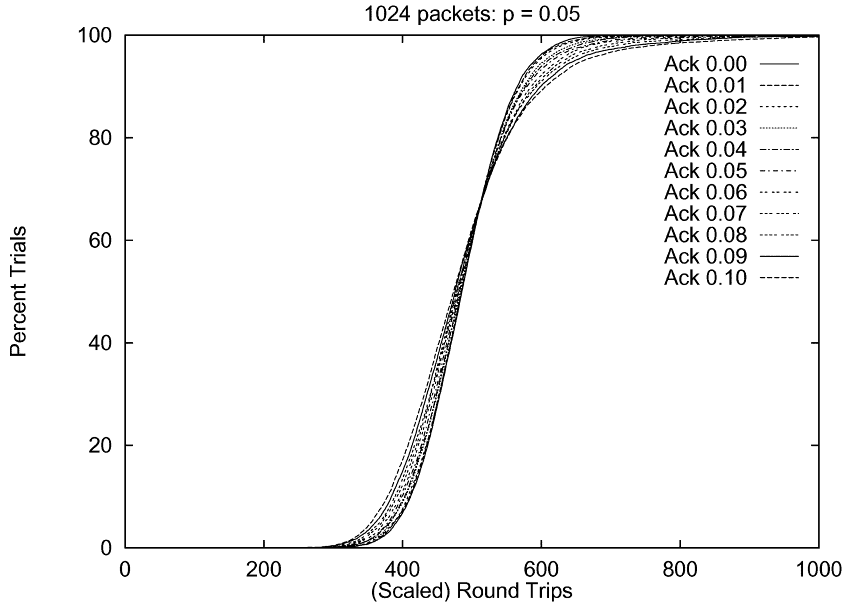

Figure 17. The correction rule applied to the cumulative distribution; 1024 packets.

appears sufficient to approximate the cumulative distribution for various amounts of acknowledgment loss.

Again, these results are dependent on the TCP variables used in these experiments. We specifically note that the effects of lost acknowledgments are less substantial when delayed acknowledgments are not in use, as one would suspect. However, they are still quite noticeable, as seen in Figure 18. At a 5% data packet loss rate, the difference in average transfer time between no lost acknowledgments and a 5% loss rate of acknowledgments is still almost 12%.

7. Variance and convergence

In this section, we examine how the variance in the transfer times changes as the size of the transfer increases. We are motivated by several works which use the round-by-round analysis approach to approximate the average throughput and goodput of a connection. For example, in [21] the authors develop a simplified Markov chain for TCP and design an equation to approximate the long-term expected throughput. In [20] the limiting distribution of the Markov chain is determined numerically in order to determine the average throughput rate.

While over the long haul we expect convergence toward the steady state rate, our results show that for short and medium transfers there can be significant variance, depending on the quantity and location of losses encountered. Hence an interesting question is how the variance of the transfer time changes with the size of the file, so that we may have some idea as to how good an approximation the steady state average throughput rate is. For convenience here we again assume no lost

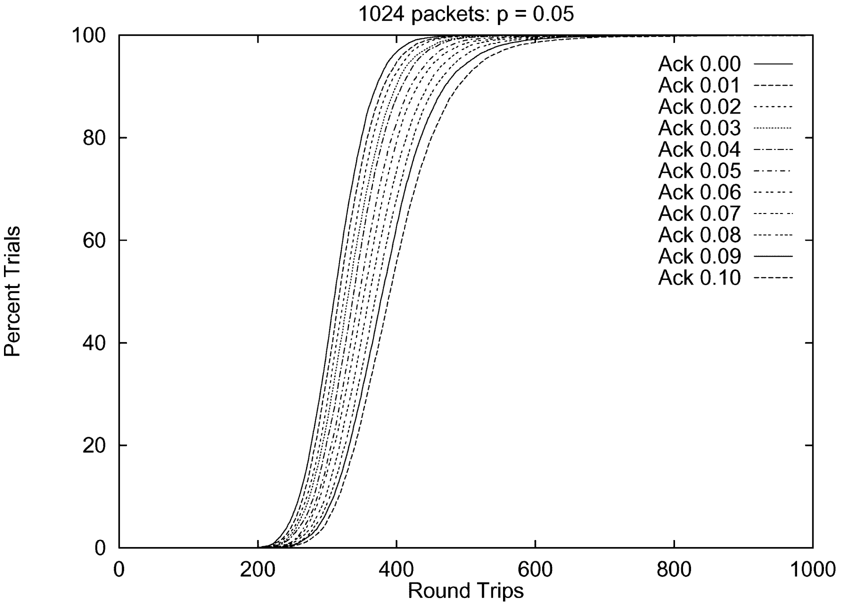

Figure 18. Cumulative distribution functions as lost acknowledgments increase without delayed acknowledgments.

acknowledgments. We use the Drop-Tail model for packet loss, although our results generally hold for all of the loss models we consider.

Figure 19 shows the probability density functions for the number of rounds to transfer 1024 packets. As can be seen, when the transfer is sufficiently long, the transfer time (like the number of timeouts) appears normally distributed. (We note that for short downloads such as 64 packets with small loss probabilities, the distribution of loss episodes appears more like a discrete Poisson distribution, and the transfer time also has approximately that form.) For a fixed loss episode parameter p, the standard deviation in the number of loss packet episodes when n data packets are transmitted is approximately $\sqrt{np(1-p)}$. (This is exact, modulo retransmissions.) Hence we would expect for fixed p that the standard deviation in the transfer time would grow proportionally to $\sqrt{n}$, and this is seen in our experiments, as shown in Figure 20. While this means that as n grows large the standard deviation of the transfer time becomes a vanishingly small fraction of the average, it also means that for short and medium downloads the variation remains significant. For 1024 packets, one standard deviation in the transfer time is still over 10% of the average; for 16,384 packets, it is only 3%.

Noting the normal distribution of the download time allows us to devise a useful correction mechanism. Given the average and the standard deviation, we expect about 1/3 of all trials to fall outside one standard deviation, and about 5% of all trials to be within two standard deviations. (The exact numbers are actually 31.74% and 4.56%.) Note that for a fixed loss episode parameter p we can approximately determine the average and standard deviations using a small number of packets n and scale up, if desired. We see in Table 1 that this rule of thumb is approximately

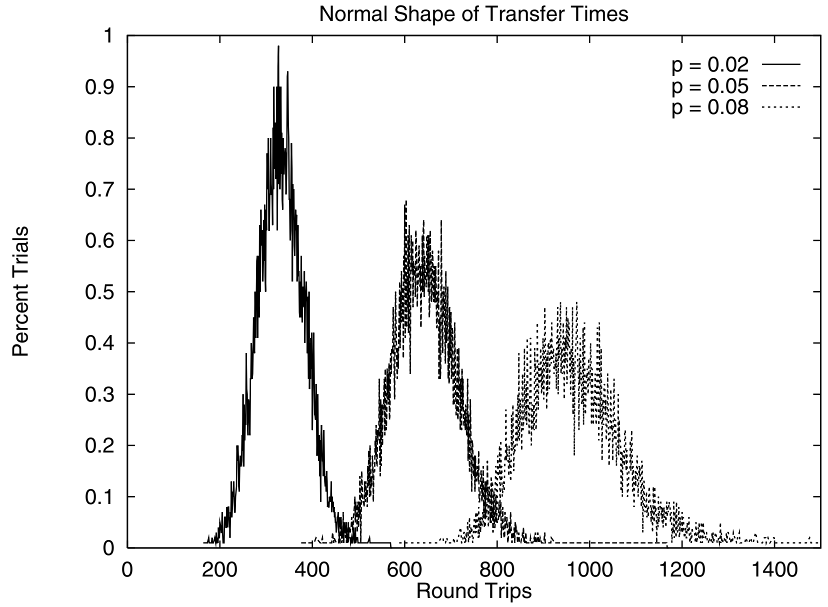

Figure 19. Density functions appear normal.

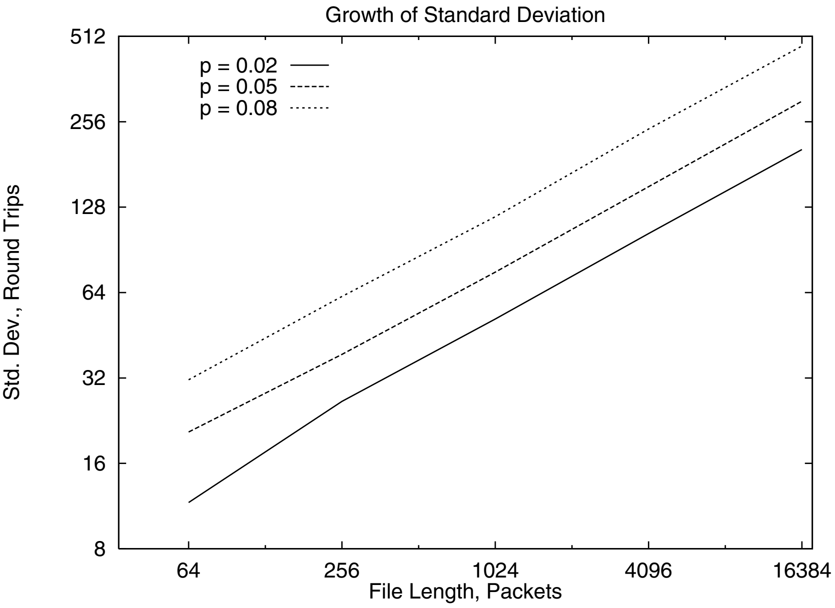

Figure 20. Standard deviations grow proportional to $\sqrt{n}$.

Table 1. Percentage of trials two standard deviations or more smaller than the mean, one deviation or more smaller, one deviation or more larger, and two deviations or more larger

Number of packets	p	< 2 sd	< 1 sd	> 1 sd	> 2 sd
64	0.02	0.00	28.4	16.5	3.7
	0.05	0.00	14.5	12.9	2.8
	0.08	0.00	8.0	10.0	3.0
1,024	0.02	2.1	15.6	15.8	2.7
	0.05	1.6	14.9	14.9	3.0
	0.08	0.6	13.6	13.2	3.2
16,384	0.02	2.2	15.8	15.8	2.3
	0.05	2.0	15.7	15.2	2.6
	0.08	1.5	16.1	15.8	2.8

Values converge to the normal distribution for sufficiently many packets.

true, when the download is sufficiently large. It is also interesting to note that there is a skew toward tails with very long download times that decreases with file size.

The relationship as p changes is less clear, as the transfer rate and the standard deviation in the transfer rate do not appear to have a linear relationship with the loss episode parameter p. As losses increase, timeouts become even more common in a superlinear way, because the timeouts arise from the interaction of several losses. Understanding in detail the loss patterns that can lead to a timeout, as described in part in [10], could shed light on this effect.

8. Conclusion

We have studied the impact of different loss models on the cumulative distribution of TCP transfer times by simulating associated Markov processes. Our simulations show that while the choice of the loss model (Bernoulli, Correlated, or Drop-Tail) has a significant effect on the actual distribution function obtained, the shape of the function is robust to changes in the model. We have quantified the preceding observation by showing that the differences among the *entire* distributions obtained for two different models can be characterized by a *single* scaling factor, which is dependent on the two models and their associated parameters. The effectiveness of such a simple correction procedure suggests that a simplified Markov model can be used to derive approximations of the full distributions under realistic loss models. We plan to explore this line of research further.

A primary reason for the effectiveness of the correction procedure is that the transfer times have an approximately linear relationship with the number of timeouts in all the loss models, and the distributions of timeouts in these models are approximately normal with different mean values. A promising direction for future research is to derive a better characterization of timeouts during TCP transfers, both in terms of the number of occurrences as well as the total duration of the timeouts. In this vein, it will also be interesting to consider alternative models for the duration of the first timeout in any sequence of consecutive timeouts. Presently, this duration is assumed to be fixed both in our simulations and in the Markov chain approaches, whereas in practice it is a random variable depending on round-trip time measurements. The difference will likely change the overall distribution of transfer time.

We have also considered how the distributions of transfer times are affected by lost acknowledgments, which are often ignored in analytical approaches for simplicity. Simulations indicate that although the distribution curves maintain the overall shape, lost acknowledgments slow the transfer considerably. The increase in transfer times appears directly traceable to a significant increase in timeouts while waiting for an acknowledgment. It would be interesting to use tools for studying critical paths of TCP, such as those developed by Barford and Crovella [3], to examine these effects in more detail.

Finally, we have studied the relationship of transfer size to the standard deviation in the distribution of transfer times. We have observed that for a fixed packet loss probability, the standard deviation is proportional to $\sqrt{n}$ for a transfer of n packets, and is significant for small to medium downloads. The growth rate of the standard deviation stems from the fact that transfer times appear to be approximately normally distributed for sufficiently long files. An interesting open problem is to characterize the standard deviation not only in terms of the file size but also in terms of the loss model parameters. Such a characterization would provide another mechanism for deriving useful approximations for full distributions of transfer times under diverse loss models.

At a higher level, we have argued that Markov models for TCP should be subjected to more detailed scrutiny by determining the effects of various simplifying assumptions. The choice of the loss model, for example, can have a significant effect on overall download time. While the ultimate goal may be a general analysis that applies to all models, a useful practical alternative we suggest is to use a correction procedure between models. In some cases, it appears that simple correction procedures can be derived experimentally. The search for correction procedures also provides insight into TCP behavior under different models that can guide the search for improved equation-based analysis.

Acknowledgments

We would like to thank Trevor Mendez for several helpful discussions concerning the `ns` simulations.

References

1. M. Allman, V. Paxson, and W. Stevens. TCP congestion control. IETF RFC2581, 1999.
2. E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. In *Proceedings of SIGCOMM*, pp. 231–242, 2000.
3. P. Barford and M. Crovella. Critical path analysis of TCP transactions. In *Proceedings of SIGCOMM 2000*, pp. 127–138, 2000.
4. J. Bolot and A. Vega-Garcia. Control mechanisms for packet audio in the internet. In *Proceedings of IEEE Infocom'96*, pp. 232–239, 1996.
5. J.-C. Bolot. Characterizing end-to-end packet delay and loss in the internet. *Journal of High-Speed Networks*, 2:305–323, 1993.

6. J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver. FLID-DL: congestion control for layered multicast. In *International Workshop on Networked Group Communication*, pp. 71–81, 2000.
7. N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. In *Proceedings of IEEE Infocom'00*, 2000.
8. Y. Chen and G. Lindsay. How to lose a customer in a matter of seconds. *Fortune*, 1, 2000. At www.fortune.com/fortune/technology/2000/06/12/ega.html.
9. D. Chiu and R. Jain. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks and ISDN*, 17:1–14, 1989.
10. K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP'. *Computer Communication Review* 26:5–21, 1996.
11. S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000*, pp. 43–56, 2000.
12. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE Network*, 1:397–413, 1993.
13. J. Hoe. Start-up dynamics of TCP's congestion control and avoidance schemes. Master's thesis, MIT, 1995.
14. V. Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM'88*, pp. 314–329, 1988.
15. J. Mahdavi and S. Floyd. TCP-friendly unicast rate-based flow control. Manuscript, 1997. URL http://www.psc.edu/networking/papers/tcp_friendly.html.
16. M. Mathis, J. Semske, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27:67–82, 1997.
17. T. Mendez. Detection of pathological TCP connections using a segment trace filter. *Computer Communication Review*, 22:28–35, 1992.
18. V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of TCP windowsize behavior. In *Proceedings of Performance'99*, Appears as Technical Report ECE-TR-CCS-99-10-01, 1999.
19. J. Mogul. Observing TCP dynamics in real networks. In *Proceedings of SIGCOMM'92*, pp. 305–317, 1992.
20. J. Padhye, V. Firoiu, and D. Towsley. A stochastic model of TCP Reno-congestion avoidance and control, Technical report 99-02, Department of Computer Science, University of Massachusetts at Amherst, 1999.
21. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8:133–145, 2000.
22. V. Paxson. Automated packet trace analysis of TCP implementations. In *Proceedings of SIGCOMM'97*, pp. 167–179, 1997.
23. V. Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Transactions on Networking*, 7:277–292, 1999.
24. B. Sikdar, S. Kalyanraman, and K. Vastola. An integrated model for the latency and steady state throughput of TCP connections. In *Proceedings of IFIP Symposium on Advanced Performance Modeling*, 2000.
25. K. Thompson, G. J. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics *IEEE Network*, 11:10–23, 1997.
26. UCB/LBNL/VINT. Network simulator ns-2, Simulation software, 1998.
27. G. R. Wright and W. R. Stevens. *TCP/IP Illustrated* Vol. *2: The Implementation*. Addison-Wesley, Reading, Mass., 1995.
28. M. Yajnik, S. B. Moon, J. Kurose, and D. Towsley. Measurement and modeling of the temporal dependence in packet loss. In *Proceedings of IEEE Infocom'99*, pp. 345–352, 1999.
29. L. Zhang, S. Shenker, and D. Clark. Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. In *Proceedings of SIGCOMM'91*, pp. 133–147, 1991.
30. Y. Zhang, V. Paxson, and S. Shenker. The stationarity of Internet path properties. Technical report, AT&T Center for Internet Research at ICSI, 2000.
31. R. Zorzi, A. Chockalingam, and R. Rao. Throughput analysis of TCP on channels with memory. *IEEE Journal on Selected Areas in Communications*, 18:1289–1300, 2000.