The Effects of Systemic Packet Loss on Aggregate TCP Flows

Thomas J. Hacker
Center for Advanced Computing
hacker@umich.edu
University of Michigan

Brian D. Noble
Electrical Engineering & Computer Science
bnoble@umich.edu
University of Michigan

Brian D. Athey
Michigan Center for Biological Information
bleu@umich.edu
University of Michigan

Abstract

The use of parallel TCP connections to increase throughput for bulk transfers is common practice within the high performance computing community. However, the effectiveness, fairness, and efficiency of data transfers across parallel connections is unclear. This paper considers the impact of systemic non-congestion related packet loss on the effectiveness, fairness, and efficiency of parallel TCP transmissions. The results indicate that parallel connections are effective at increasing aggregate throughput, and increase the overall efficiency of the network bottleneck. In the presence of congestion related losses, parallel flows steal bandwidth from other single stream flows. A simple modification is presented that reduces the fairness problems when congestion is present, but retains effectiveness and efficiency.

1 Introduction and Motivation

Recently, the high-performance computing community has found that bulk TCP transfers over Abilene[1] do not approach network capacity. This is true despite the lack of congestion; instead, such flows experience systemic, random loss events that prevent full utilization. In response, the community has turned to the use of *parallel* TCP connections to carry what was formerly a single flow, striped across all of them. Empirically, such parallel flows significantly outperform singletons [HaAth2001], and have been integrated into a variety of libraries supporting parallel and grid computing, including bbcp [HaTr2001], GridFTP [AlBe2001], Storage Resource Broker [BaMo1998], and Globus [FoKe1997].

Measurements of the Internet at large [StPa2000] [NiTier99] [Tos2001] suggest that losses caused by circumstances other than congestion are a broader problem, not confined to Abilene. This raises the possibility that parallel TCP connections could be useful more broadly. In the absence of congestion, parallel TCP transmissions are more robust in the face of systemic, random loss. However, in the presence of congestion, parallel TCP transmissions effectively defeat TCP's congestion control mechanisms, leading to unfairness and, potentially, congestion collapse.

This paper evaluates the effectiveness, fairness, and efficiency of parallel TCP flows in the presence of systemic, random loss. Effectiveness is whether parallel flows increase throughput. Fairness is defined as equal sharing of the bottleneck by flows. Efficiency is whether the use of parallel flows improves the utilization of the network.

We measured congestion-free losses in Abilene, and used these observations to derive an accurate loss model. We incorporated this loss model into the ns2 simulator [NS2], and simulated the behavior of parallel streams in competition with themselves and single-stream flows, both with and without congestion. The results of the simulation are consistent with analytical predictions and empirical observations of performance.

Predictably, parallel TCP flows are more robust when systemic, random losses are present. The observed loss process limits a single stream to just over 25% of a 100 Mb/s link, but six streams in parallel achieve more than 80% utilization. However, when parallel flows compete with single-stream ones, the former steal bandwidth from the latter. To address this problem, we introduce *fractional congestion control.* In it, a single TCP stream is told to increase its congestion window by only one packet for every N acknowledged packets, but decreases its window in the normal way. Fractional congestion control can be used to reduce the aggressiveness of parallel streams in the presence of congestion, but preserve much of their effectiveness in its absence. The effects of fractional congestion control are similar to the effects of long round trip times on TCP flows, which cannot compete effectively with short round trip time flows [QiZhKe2001].

The next section of this paper will briefly review the factors that affect aggregate TCP performance. The sections following will discuss the choices made in equipping the simulation, and address the effectiveness, fairness, and efficiency questions using results from the simulation.

[1] Abilene is an advanced backbone network linking academic and research centers in the United States. Abilene is operated by the University Consortium for Advanced Internet Development based in Ann Arbor, Michigan.

0-7695-1524-X/02 $17.00 (c) 2002 IEEE

2 Background and Related Work

The TCP congestion avoidance algorithm, first described by Jacobson [Jac88], was designed to prevent congestion on shared public networks. When a network becomes congested, packets are dropped by routers and switches due to queue overflows. The congestion avoidance algorithm responds to packet drops by reducing the transmission rate of the TCP sender. After reducing the transmission rate, the TCP sender enters a recovery phase in which the number of packets in flight is increased by one every round trip time until the network again becomes congested, at which point the transmission rate is again reduced.

The TCP congestion avoidance algorithm is a simple control system in which packet loss packet represents feedback from the system under control [ChJa89]. The implicit assumption in the congestion avoidance algorithm is that packet losses are due only to network congestion. The goals of the congestion avoidance algorithm are to provide fair sharing of bandwidth between competing TCP flows, to maximize the overall use of available network bandwidth, and to prevent network congestion.

Several expressions that relate TCP throughput to various factors have been derived. Bolliger [BoGrHe99] provides an excellent overview of these expressions and their relative accuracy. The Mathis equation [MaSe1997] describes the relationship between frame size (MSS), RTT and packet loss rate that regulates peak TCP throughput for a single TCP flow. Bolliger found that the Mathis equation is essentially as accurate as the Padhye equation [PaFi1998] for packet loss rates less than 1/100.

Three empirical factors affect the TCP transmission rate. The first, maximum segment size (MSS), is dependent on the static configuration and architecture of the end-to-end network. The second factor, Round Trip Time (RTT), ranges from a minimum value determined by distance and network architecture to a maximum value that is a function of the total size of the network and network queues between two hosts. If the network is experiencing low traffic load, RTT will not tend to grow large. However, under heavy load, RTT can become a dominant factor in determining peak TCP throughput.

The third factor, packet loss rate, is the ratio of the number of dropped packets to the total number of packets transmitted. This factor is the most dynamic of the three when the network is not overloaded and there are non-congestion packet drops. The packet loss rate factor used in the Mathis equation is a long-term approximation based on a constant probability distribution of packet loss. The effects of packet loss on smaller time scales are also affected by the statistical distribution (i.e. "burstyness") of packet loss events. Empirical measurements on the Internet [Slac2001] have shown that packet loss is persistently high and limits TCP throughput. Low packet loss rates (1 in 10^6) are necessary to efficiently utilize modern high-speed network equipment. Unfortunately, wide-area public networks do not currently demonstrate the low level of packet loss necessary for good throughput [InTr2001].

The predominant view is that packet loss is caused exclusively by routers dropping packets when queues overflow. A recent study [StPa2000] found that between 1 packet in 1,100 and 1 packet in 32,000 fails the TCP checksum and is dropped, even when the data-link CRC checksum passes. A thorough investigation of this failure in the study revealed at least 40 different sources of error. Another example source of packet loss not due to congestion is described in a report [Tos2001], in which a large amount of packet loss in a cable modem was caused by a hardware bug. Nitzan and Tierney [NiTier99] found ATM cell drops due to hardware problems limited TCP performance over OC-12 links. Floyd [FlRa2002] [Fl2002] found that the hardware bit error rate of fiber optic media is large enough to prevent TCP from fully utilizing bandwidth on 10 Gb/sec networks.

Given this evidence, the belief that packet losses are caused only by queue overflows cannot be supported. There are many pieces of network equipment and carriers involved in a wide-area network data transmission – a bug or error in any of these pieces of equipment may corrupt or drop a packet.

With such a potentially high packet loss rate, it is reasonable to consider the scenario in which systemic packet loss slow TCP flows to the point at which it is impossible to reach the delay bandwidth product of the connection and the congestion point in the network. This scenario is most likely to occur in high speed wide-area networks, such as Abilene, that have substantial amounts of available bandwidth with no congestion. In a previous study by Hacker [HaAt2001], information provided by the Abilene NOC and the MichNet NOC [RoAb2001] indicated that no packets were dropped on either Merit or Abilene routers due to transfer experiments. For this study, the campus NOC [Kos2001] reported zero router discards for the network bottleneck during the collection of trace data[2]. This evidence indicates that the network is underutilized, and that packet drops are caused by factors other than congestion.

Concurrent flows have long been viewed as unfair competitors to "ordinary" TCP transmissions.

---

[2] The correct operation of the mechanism that detects router discards in the network bottleneck was verified by forcing the network into congestion with a large number of UDP and TCP streams.

Balakrishnan [BaRaSe99] presents a software tool called the Connection Manager that provides an API for application controlled congestion management on HTTP and UDP flows. Such a tool could be used to collectively manage a set of concurrent flows in order to provide effective yet fair parallel flows. Rather than centralized control, our approach provides for control local to each flow.

3 Parallel TCP Throughput

This section reviews the factors that affect aggregate throughput for hosts that utilize parallel TCP streams. The reader should refer to Hacker [HaAt2001] for a more complete discussion.

When an application uses s multiple TCP streams between two hosts, the aggregate bandwidth of all s TCP connections can be derived from the Mathis equation, in which MSS_i, RTT_i, and p_i represent the relevant parameters for each TCP connection i:

$$BW_{agg} \leq C\left[\sum_{i=1}^{s} \frac{MSS_i}{RTT_i\sqrt{p_i}}\right] \qquad (1)$$

Since the value for MSS is determined on a system wide level by a combination of network architecture and MTU discovery, it is reasonable to assume that each MSS_i value is identical and constant across all simultaneous TCP connections between the two hosts. We may reasonably assume that RTT will be equivalent across all TCP connections, since the entire network path traversed by packets for each TCP connection will likely take the same network path and experience the same end-to-end queuing delays.

If parallel TCP streams are used on a network with unused bandwidth, and there are systemic packet losses, the packet loss rate p_i experienced by each TCP stream will effectively determine the maximum throughput of each stream. As the number of parallel TCP streams increases, the behavior of each packet loss factor p_i should be unaffected as long as few packets are queued in routers or switches at each hop in the network path between the sender and receiver. This progresses until the traffic reaches capacity. Thereafter, the loss process is primarily driven by congestion.

Examining equation (1), some features of parallel TCP connections become apparent. First, an application opening s multiple TCP connections is in essence creating a large "virtual MSS" on the aggregate connection that is s times the MSS of a single connection. Second, a single loss event causes only one of s streams to decrease their window size. Together, these allow a parallel flow to put more packets "in flight" on the network during the congestion avoidance algorithm increase phase than a single stream flow. This has significant effects on throughput on wide-area networks with long round trip times, since single stream flows with large RTTs cannot compete on an equal basis with flows with small RTTs. If we factor RTT and MSS out of equation (1):

$$BW_{agg} \leq C\frac{MSS}{RTT}\left[\sum_{i=1}^{s} \frac{1}{\sqrt{p_i}}\right] \qquad (2)$$

Given the relatively stable nature of the values of MSS and RTT compared with the dynamic nature of p, the packet loss rate p is an essential factor in the aggregate throughput of a parallel TCP connection. Given the importance of this factor, simulations used to investigate parallel TCP flows must pay special attention to the accurate modeling of packet loss.

3.1 Fractional Parallel TCP Flows

When a network bottleneck is fully utilized with n single stream flows, the TCP congestion avoidance algorithm allows the single stream flows to converge to an optimal fairness and efficiency point in the absence of systemic non-congestion packet loss [ChJa89]. When an application uses s parallel flows out of the n total flows, the application unfairly competes with the remaining (n-s) single stream flows for bandwidth. However, when packet losses are primarily non-congestive, even a modest number of single stream flows are incapable of fully utilizing the network. In such cases, parallel flows consume the unused bandwidth, rather than stealing bandwidth from competing single stream flows.

The aggressiveness of parallel flows is due to two factors: a faster aggregate recovery rate and a larger resistance to loss. When a router drops packets across all flows, the aggregate recovery rate of s parallel flows is s times the recovery of a single stream flow. When there are non-congestion packet losses, parallel flows absorb the loss on a subset of the flows, while allowing the remaining flows to continue to increase their congestion window.

Our goal is to modify parallel TCP behavior to remove the undesirable fairness problems, while retaining the desirable effectiveness and efficiency characteristics. Intuitively, parallel flows compete unfairly because they collectively open their congestion window more aggressively than an equivalent single stream. One can tame this aggression by limiting the rate of window increase. We considered two approaches to doing so.

In the first approach, we limit the collective rate of increase to be equivalent to that of a single flow. In this scheme, each one of n flows increases its conges-

tion window by one packet per *n* packets acknowledged. This scheme can still tolerate non-congestive losses, because only one of *n* parallel streams will decrease its window in response to a loss event. We refer to this as the *fractional approach*. This reduces, but does not eliminate, the unfairness of parallel flows.

Our second approach constructs a parallel flow by augmenting a standard TCP flow with parallel flows that open their window very conservatively. Each of the *n* conservative flows increases its congestion window by one packet for every *jn* it receives, for some positive integer *j*. Intuitively, when there is slack bandwidth available, these conservative flows should be able to consume it. However, in the presence of congestion, the conservative flows cannot compete, and throttle themselves. Because it combines standard TCP with fractional congestion control, we call this the *combined approach*. The intuition behind this approach is that, when not all bandwidth is consumed, the fractional flows can expand to do so. However, during times of congestion, the fractional flows cannot compete with single stream flows, and are effectively eliminated.

The fractional and combined approaches are equivalent to a standard TCP flow with a round trip time longer than competing single stream flows. Since TCP flows with long round trip times cannot compete with flows with short round trip times, the fractional and combined approaches can consume unused bandwidth when there is systemic loss present, but cannot effectively compete with standard flows that have a shorter round trip time.

3.2 Throughput of Fractional TCP Flows

To understand the effects of fractional flows on parallel TCP throughput, deriving an expression for fractional parallel flow bandwidth allows us to estimate the upper bound on the amount of bandwidth fractional parallel flows can consume. To perform this derivation, we will use the geometric construction technique of Mathis [MaSe1997]. We assume that the packet loss rate for each TCP flow is equivalent, and that the packet loss rate is constant for each flow of an *n* flow parallel stream.

To begin, consider Figure1, which shows the pattern of TCP throughput as a function of time for an individual flow of a fractional stream. With the fractional congestion avoidance algorithm, the throughput (or number of packets in flight) will increase for a period of time equivalent to *n* times the RTT of a single flow before a packet drop.

The number of packets that are sent within the time cycle of length *na* is equal to the area of Figure1:

$$A = na^2 + \frac{na^2}{2} = \frac{3na^2}{2} \quad (3)$$

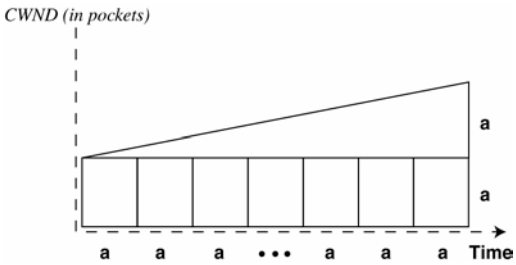

Figure 1. Geometric Construction for Fractional Stream of *n* Parallel Streams

Figure 2 shows the situation for a non-fractional single stream flow in which time *y* elapses before a packet is dropped and the congestion window is reduced by half.

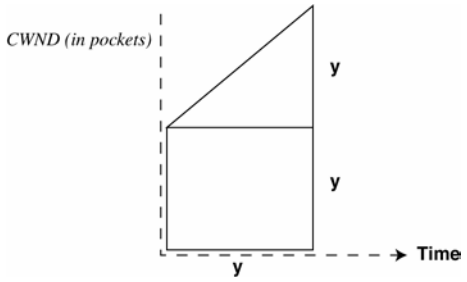

Figure 2. Geometric construction for a Single TCP flow

The area of Figure2 is $Y = \frac{3y^2}{2}$

Since the packet loss rate per flow as a function of the number of packets transmitted is constant, Figure1 and Figure2 represent the same number of packets transmitted over different time scales. Since the number of packets is the same, $X=Y$ and thus:

$$\frac{3na^2}{2} = \frac{3y^2}{2} \Rightarrow a = \frac{y}{\sqrt{n}}$$

Bandwidth is the number of packets transmitted per unit time. Bandwidth for the fractional single stream is

$$BX = \frac{3a}{2} = \frac{3y}{2\sqrt{n}} \quad (4)$$

Since the bandwidth for the single stream flow is known, substituting from equation (4):

$$BY = \frac{3y}{2} \Rightarrow BX = \frac{1}{\sqrt{n}} BY \quad (5)$$

Thus, the throughput of a single fractional stream of an *n* stream parallel flow is related to a single stream TCP flow by 1 over the square root of the number of fractional streams. To compare *n* fractional parallel streams

to a single stream, we simply add up all of the fractional terms

$$BW_{parallel} = \sum_{1}^{n} \frac{BY}{\sqrt{n}} = \sqrt{n} BW_{unistream}$$

This is the best possible case, in which all fractional streams are adding together to create an aggregate TCP flow that overcomes the effects of systemic packet loss when there is available network bandwidth. Without loss of generality, if a fractional TCP stream is *jna* units (*j* a positive integer),

$$BW_{parallel} = \sqrt{\frac{n}{j}} BW_{unistream} \qquad (6)$$

If j > 1, more round trip times are required to increase the congestion window. From equation (2), it is apparent that this is functionally equivalent to a *virtual* round trip time, since the number of packets necessary to increase the congestion window by one has been increased. Since the virtual round trip time is longer, fractional streams with j > 1 are less effective at increasing throughput, but are fairer, since they cannot effectively compete with flows with short round trip time.

4 Methodology

To evaluate the effectiveness, fairness, and efficiency of parallel TCP, the ns2 network simulator [NS2] was used to create a simulated network environment. The simulation was used to conduct over 5400 simulation experiments. The impact of *s* single stream flows, *s* fractional flows (fractional approach), and a single stream flow coupled with *(s-1)* fractional flows (combined approach) on effectiveness, fairness, and efficiency was evaluated using the simulation.

This paper has four remaining sections. In Section 5, the outfitting of the simulation and selection of the packet loss model is discussed. In Section 6, the effectiveness, fairness, and efficiency of parallel TCP flows are assessed using the results of simulation experiments. In Section 7, the implications of these assessments are discussed along with proposed future work.

5 Outfitting the Simulation

This section describes the simulation used for experiment in detail, and discusses the selection of the packet loss model for the simulation. Finally, an assessment of the accuracy of the simulation will be presented.

5.1 Network Configuration

The network configuration used in the simulation is shown in Figure 3. The link speeds and propagation delays were based on network measurements using traceroute and Pchar [Mah] between two sites on either side of the United States connected by the Abilene and CalREN networks.

In the configuration in Figure 3, six nodes are connected to a bottleneck node with 100 Mb/sec connections. The bottleneck node (B in Figure 1) then connects with a 100 Mb/sec connection to a high-speed node. One of the six nodes (marked P in Figure 1) played the role of a host that performed simulated FTP transfers over parallel TCP flows to the terminus node (marked T in Figure 1). The remaining five end nodes (marked S in Figure 1) played the role of single TCP stream hosts sharing the bottleneck link with the parallel transmission node P.

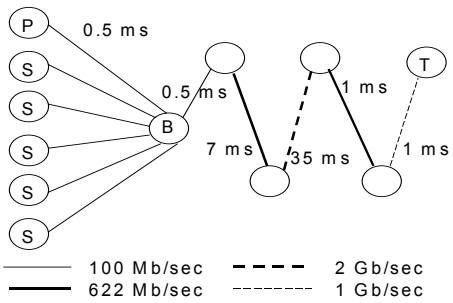

Figure 3. Simulation Network.

Careful selection of the queue length of the bottleneck node B is important to assess the impact of non-congestive and congestive losses on the effectiveness, fairness, and efficiency of parallel flows. In one set of simulations, the queue length was set very large, to assess the effects of non-congestive loss. We will refer to this set of simulations as the *systemic loss simulation*. Another set of simulations used a queue length of 50 packets for the bottleneck node B. We will refer to this set of simulations as the *congestive simulation.* In the congestive simulation, losses were caused both by the systemic loss model and by queuing losses. This distinction was necessary to differentiate the effects of systemic loss from pathological queuing behaviors (such as lock-out and phase effects) and congestive loss. All flows were TCP SACK flows with large TCP buffers. Explicit Congestion Notification (ECN) was not used.

5.2 Simulation Loss Models

Choosing a loss model that accurately represents observed loss behavior is critical for accurate simulation. We generated and measured traffic on Abilene over two days, confirming the absence of congestive loss. The Iperf utility [GaWa2000] was used to perform the

data transmission experiments. Tcpdump was used to collect every packet transmitted and received from the remote Iperf server. Tcptrace was used with locally developed perl utilities to generate the number of bytes and wall clock time that elapsed between packet loss events in the transmissions. Over the two days, we observed 9,263 losses out of 62,376,519 total packets, yielding a loss rate of 0.01%. This modest loss rate limits the throughput of a single TCP stream to 25 Mb/sec; far below what one would expect given the network capacity and utilization.

5.2.1 Analysis of Packet Loss Data

Figure 4 shows the distribution of elapsed time between packet loss events. Viewed casually, a Pareto distribution appears to fit the empirical observations adequately. This is tempting, as it would be consistent with results observed in the work by Borella [BoSw1998], characterizing wide-area losses likely due to congestion. However, close examination reveals that the interloss data contains (at least) two distributions. The dominant distribution was the intraloss times within a packet loss burst event. A latent distribution was the time that elapsed between packet loss burst events (interloss times). A similar bi-modal distribution was observed by Ferrari and Cammarata [FerCa02] in tests from INFN in Bologna, Italy to the University of Michigan in Ann Arbor, Michigan and NIKHEF in Amsterdam, Netherlands.

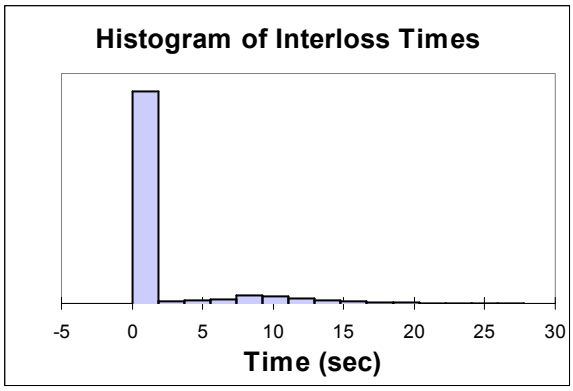

Figure 4. Histogram of Elapsed Time between Packet Loss Events

If we cleave the dataset at the 1-second point, and examine the latent distribution on the right, the distribution appears to be a normal distribution. Figure 5 shows the histogram for this distribution with a fitted random distribution overlay. A P-P plot for the fit of a random distribution to this data indicates a good fit.

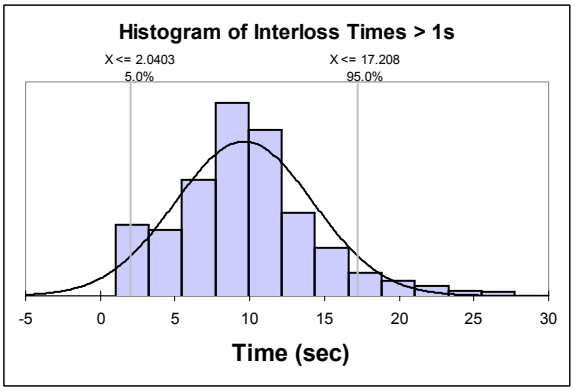

Figure 5. Histogram of Elapsed Time Between Packet Bursts.

If we now turn our attention to the left side of the distribution in Figure 4, the dominant distribution appears to be exponential, but in fact it is a combination of many different exponential distributions from the packet loss interarrival time within each packet loss burst event. Zhang [ZhPaSh2000] observed this behavior in aggregate packet loss data collected over long time scales. Due to the large number of packet loss bursts, it was not feasible to analyze all of the individual bursts to determine the fraction of these bursts that exhibited exponentially distributed inter loss elapsed time. To statistically gauge the proportion of the burst events that fit an exponential distribution, 10% of the bursts were randomly selected from the dataset. Approximately 1/3 of the selected sample contained the minimum 5 observations necessary to fit a distribution. Using the Anderson-Darling test, 61% of the examined bursts fit an exponential distribution. When the bursts were further analyzed to remove large burst time outliers, which may in fact belong to the left edge of normal distribution on the right-hand side of Figure 4, 78% fit an exponential distribution using the Kolmogorov-Smirnov test. Thus, between 61% and 78% of the intraburst elapsed times fit an exponential distribution. Previous studies [MiGoTo1999] [ZhPaSh2000] have performed this analysis on a large number of collected packet losses, and determined that a significant percentage of packet losses exhibited exponential interarrival times and i.i.d. behavior.

From this analysis, it appears that the elapsed time between packet loss events exhibits two characteristics. First, within a packet burst, the losses arise from a Poisson process, due to exponentially distributed interarrival times. The conglomeration of these exponential times can be reasonably approximated as an exponential distribution with a median value derived from empirical observations. Second, the elapsed time between packet loss burst events arises from a longer time scale process approximated by a random process that can be characterized by the median derived from the observed

distribution. The independence property of the elapsed time between packet loss events was validated using the Brock-Dechert-Scheinkman (BDS) test described in [LeBaron1997] using ε of 1 standard deviation. The BDS test is designed to validate independence in time series data that may contain long-range dependence data, such as stock market prices.

5.2.2 Selecting a Packet Loss Model

There are many loss models presented in the literature. The work by Altman [AlAvBa2000] contains an excellent discussion and analysis of many of the models that have been used over the years. To select the loss model that most accurately characterizes the observed packet loss characteristics, the inherent features and assumptions of each loss model must be considered. Each loss model is based on statistical assumptions about actual loss behavior that is modeled. This section will review the loss models used in the literature, and assess the appropriateness of each model based on the loss behavior observed in the dataset.

The simplest models are the constant loss probability and the random loss models. The observed packet loss behavior exhibited burst characteristics in which a group of packet losses occurred in a short time, followed by a long period of elapsed time with no loss. This observed loss characteristic eliminated these models.

Another model used in the literature is the Poisson arrival process, in which packet losses arrive according to a Poisson arrival process with exponential interarrival times. This model is only valid if the packet loss interarrival times can be shown to be exponential, and the packet loss events can be shown to be independent events. The Poisson model mimics behavior within a packet burst, but fails to capture the long periods that elapse between packet burst events. Recent work [YaMoKu1999] found that the packet loss data collected for their study were independent and fit an exponential distribution. The aggregate packet loss data collected for this paper, however, did not reasonably fit an exponential distribution. Thus, this model was rejected.

The next loss model considered is the unconditional and conditional loss model. In this model, bursty loss behavior is modeled as two probabilities. This model is described by Bolot [Bolot1993]. The use of this loss model requires memory of the state of the previous packet (lost or not lost). Since this model only retains memory of the state of the last packet, it does not contain the granularity necessary to accurately model the multiple packet losses that occur within a packet loss burst. Moreover, Jiang [JiSc2000] found that this model underestimated the probability of consecutive packet losses, and thus could not accurately model packet bursts. For these reasons, this model was rejected.

To retain a limited amount of memory of the gaps between packet loss events, a 6-state Markov chain was chosen for the simulation experiments. The sojourn times for each state and transition probabilities were selected from the observed distribution of the elapsed time between packet burst events. Based on the comparison between simulated and measured throughput (discussed in detail later), the 6-state Markov chain was sufficient. Fewer states would result in a less accurate model, and more states would provide a more accurate model.

The Markov model selected for modeling the interburst loss behavior cannot model the packet loss behavior within a packet burst event (intraburst behavior). To model the intraburst behavior, one node of the Markov chain was selected to represent the loss activity that occurs during a burst event. Since the intraburst behavior demonstrated Poisson arrival characteristics (exponential interarrival times, independent events), the loss state of the Markov chain simulated intraburst loss events by selecting random elapsed times between packet losses from an exponential distribution with a median calculated from the observed data. The sojourn time of the loss state was calculated from the 85% CDF of the intraburst loss data. This model is a Markov Modulated Poisson Process (MMPP). MMPPs are described in detail in Fischer [FiHe1992]. Altman [AlAvBA2000] found that losses on a wide area network are best modeled with a Poisson loss process based on a Markov Arrival Process, of which MMPP is a subset.

5.2.3 Implementing the MMPP model in ns2

The ns2 simulator supports a *k*-state Markov error model with the MultiState loss module. The MultiState module supports a separate loss module for each Markov state, with fixed state sojourn times. To implement a 6-state MMPP model in ns2, one state was configured to generate packet losses from an exponential distribution with a median value calculated from the trace data. To simulate the randomly distributed gaps between packet bursts, the transition probability vector for the loss state L was selected to approximate a random distribution with median and range calculated from the normal approximation to the interburst elapsed time. At the end of the sojourn time for each non-loss state, the loss state was always selected as the next state. Each successful run of a loss simulation consisted of 10 separate runs, each seeded with a different random seed selected from a random distribution.

We corrected two shortcomings in the ns2 MultiState loss module. First, the MultiState model did not sup-

port generating losses on multiple simulated network links. To correct this, the MultiState model was extended to support loss simulation on multiple links. Second, the MultiState loss module does not have the ability to attach the loss process to an individual flow. To support the simulation of losses on a single link across multiple TCP streams, the sojourn times for lossless Markov states were divided by the number of streams, to ensure that the intensity of the loss process for each stream would be consistent across all of the parallel TCP streams as the number of streams was increased.

Since SACK TCP is designed to handle multiple packet losses within a window, simulations that rely on a constant loss model will not accurately represent observed packet loss behaviors. To verify this, a subset of the simulations presented in this paper was run with a constant packet loss model. The results from a constant loss model were different from the results using the MMPP packet loss model presented in this paper.

To validate the accuracy of the MMPP loss model, a series of single TCP SACK streams were simulated, and compared with throughput observed from the Iperf measurements. The medians of the observed and simulated throughput distributions were within the 90% confidence interval of each other, and thus were statistically similar. All of the simulations presented in this paper are based on SACK TCP.

6 Effectiveness, Fairness, and Efficiency

This section will raise and answer the questions of effectiveness, fairness, and efficiency when parallel TCP connections are used to increase aggregate throughput. To answer these questions, results from simulation experiments will be presented and discussed.

6.1 Effectiveness

Striping data transfers across parallel TCP sockets is beneficial only if overall throughput is increased. To determine the effectiveness of striping data transfers, a series of simulation experiments were conducted for each of the three approaches (unmodified, fractional and combination, described earlier). Each experiment consisted of a 500 second simulated FTP transfer across the simulated network from a single node with no other traffic. The number of parallel TCP connections used for each experiment ranged from 1 to 12. For a given parallel factor, 10 experiments were conducted with the random seed for the loss model selected from a predetermined set of random values. Figures 6 and 7 show the simulation results for the systemic and congestive loss simulations. Results from the simulations indicate that an increase in aggregate throughput can be expected when data is striped across parallel TCP connections, regardless of the approach used.

The difference in effectiveness between the different approaches is due to the difference in recovery rate of each individual flow from losses. With the unmodified approach, each flow of the parallel flow recovers at the rate of 1 packet for every ACK received. With the other approaches, the recovery rate per stream is much less. For the fractional approach, the recovery rate is only $1/s$ for each stream. For the combined approach, the recovery rate is $1/10s$ for the systemic loss simulation and $1/2s$ for the congestive simulation for a parallel TCP flow set of size s. Recall that the value of j selected for the recovery rate is equivalent to increasing the round trip time of a flow. Thus, the selection of j is a trade-off between effectiveness and fairness.

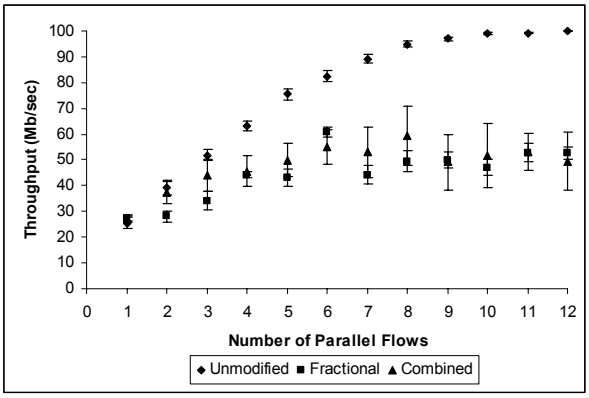

Figure 6. Aggregate Throughput vs. Number of Parallel TCP in Systemic Loss Simulation.

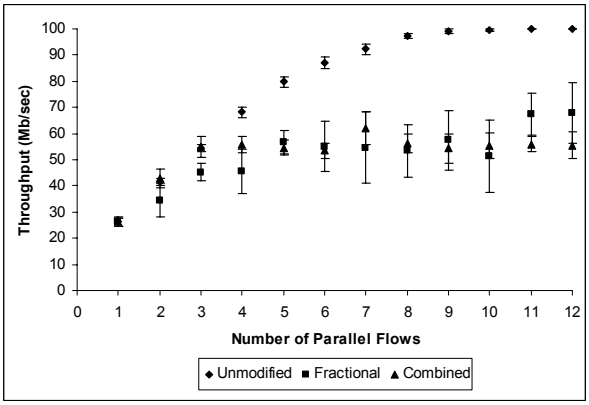

Figure 7. Aggregate Throughput vs. Number of Parallel TCP Streams in Congestive Simulation.

Since overall throughput for each approach increases with the number of streams, the use of parallel TCP connections to improve aggregate throughput is effective. The relative effectiveness of each approach depends on the rate of recovery of each flow from loss.

6.2 Fairness

Fairness is defined as proportional sharing of the bottleneck by flows. Specifically, a parallel stream should not steal bandwidth from competing single stream flows. To address this question, a series of simulation experiments were performed for each of the three approaches that varied the number of parallel TCP streams from 1 to 12 on the parallel TCP node, with 1 to 5 competing single TCP flows from cross traffic nodes. All of the traffic source nodes in the simulation share a 100 Mb/sec bottleneck link.

Figures 8.1 and 8.2 shows the effects of increasing the number of cross traffic flows on aggregate TCP throughput from the parallel TCP node using a set of single stream flows with the unmodified approach and the systemic loss simulation. Each figure is a stacked area graph that shows the total cross traffic throughput on the bottom and the aggregate parallel TCP throughput on top. The quantities are stacked in the graph, thus the height of each stack is the total throughput on the network bottleneck for all traffic.

The simulation results showed that when the available bandwidth was more than approximately 10 Mb/sec (less than 90% utilization), parallel flows consumed available bandwidth without stealing bandwidth from competing single stream flows. This is apparent in Figure 8.1. When the available bandwidth was less than 10 Mb/sec, the only way that the parallel TCP flow could increase throughput is to steal the bandwidth from competing single stream flows. This is apparent in both Figures 8.1 and 8.2.

Unmodified TCP fairness in the congestive simulation is shown in Figures 9.1 and 9.2. In the congestive simulation, the combination of systemic and queuing packet loss is large enough to prevent the full utilization of the bottleneck bandwidth. Additionally, the set of parallel flows dominate the queue and lock out packets from the cross stream flow, causing packet loss due to queuing drops and packet timeout on the cross stream nodes.

Figure 10 shows the median amount of bandwidth stolen from each single stream in the systemic loss simulation when the network was over 90% utilized.

From the simulations, it is clear that when systemic loss is the only source of loss, unmodified TCP parallel flows can utilize unused bandwidth until about 90% of the network bottleneck is utilized. In the presence of both systemic and congestion loss, unmodified TCP flows steal a significant amount of bandwidth from single stream flows.

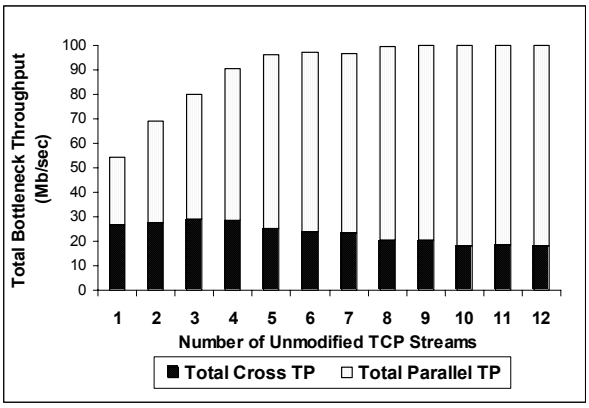

Figure 8.1 Single Cross TCP Stream with Unmodified Approach in Systemic Loss Simulation

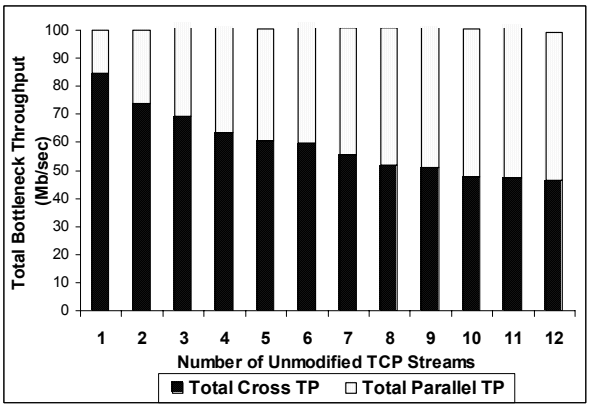

Figure 8.2 Five Cross TCP Streams with Unmodified Approach in Systemic Loss Simulation

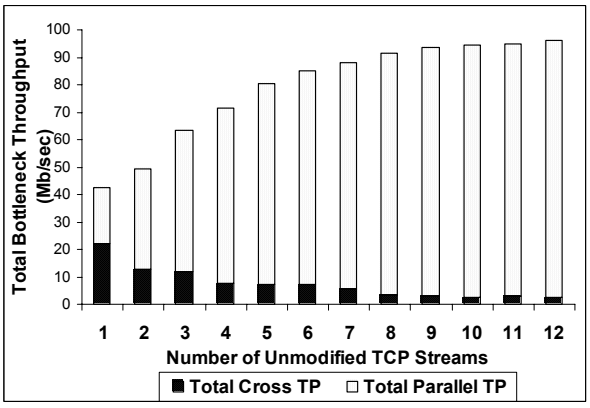

Figure 9.1 Single Cross TCP Stream with Unmodified Approach in Congestive Simulation

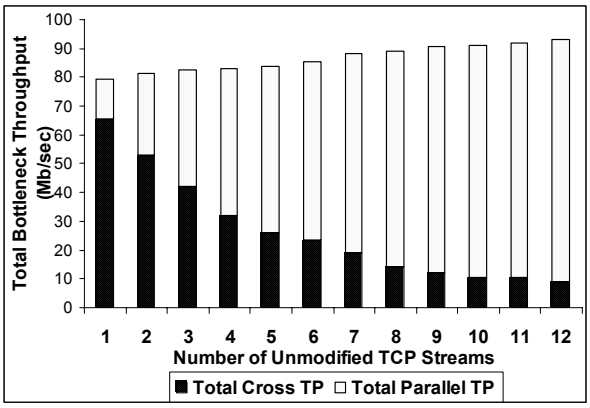

Figure 9.2 Five Cross TCP Streams with Unmodified Approach in Congestive Simulation

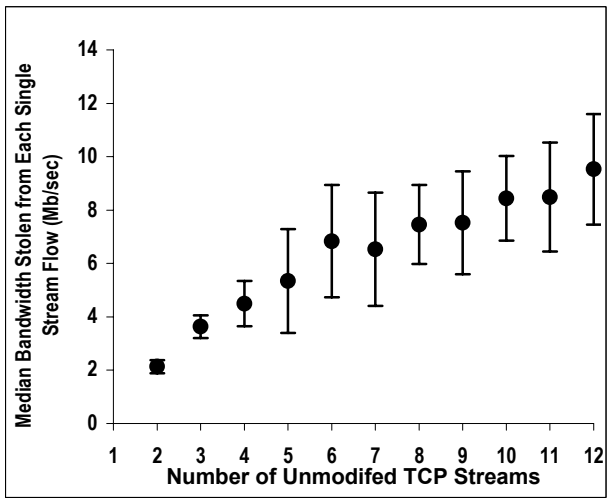

Figure 10. Median Bandwidth Stolen

6.2.1 Improving Fairness

In large research networks such as the Abilene network [Ab2001], there is currently considerable unused bandwidth available, and router drops due to congestion are rarely (if ever) encountered. On this class of shared networks with available unused bandwidth, the use of parallel TCP flows is unlikely to steal bandwidth from other users of the backbone network. On the commercial Internet, however, there may not be available unused bandwidth. In this situation, the use of parallel TCP connections will steal bandwidth from other flows.

To assess the effect of the two modified approaches described in Section 3.2 on fairness, the simulations were run for the fractional and combined approaches. Figures 11.1 through 11.2 show the results of the fractional approach on parallel and cross traffic throughput for 1 and 12 cross streams in the systemic loss simulation.

Comparing the graphs in Figures 8 and 11, it is apparent that the fractional approach resulted in less aggressiveness than the normal congestion avoidance algorithm. However, it still competes unfairly. Figure 13 shows the median amount of bandwidth stolen from each competing single cross stream as the number of fractional parallel TCP flows increase when the network is over 90% utilized in the systemic loss simulation.

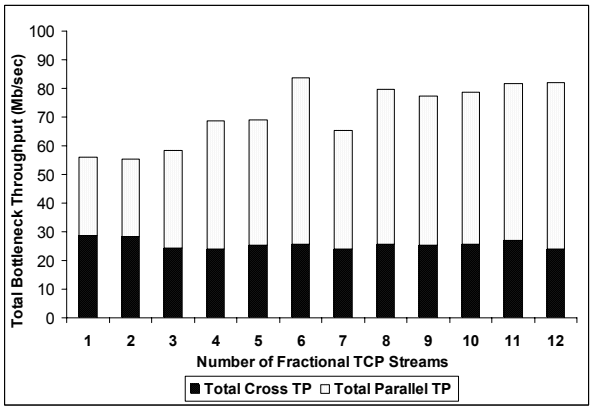

Figure 11.1 Fractional Parallel Flows with 1 Cross Stream in Systemic Loss Simulation

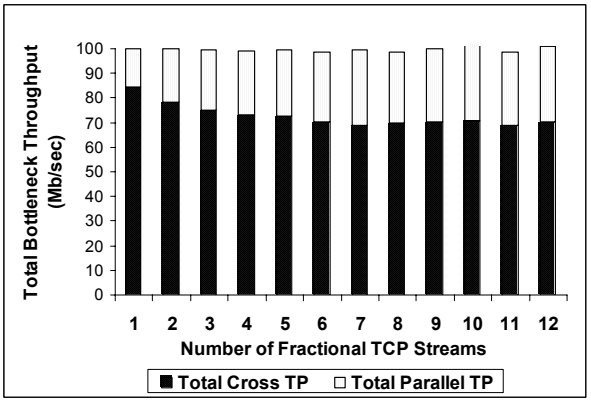

Figure 11.2 Fractional Parallel Flows with 5 Cross Streams in Systemic Loss Simulation

In Figures 12.1 and 12.2, the fractional approach resulted in less stealing of bandwidth from cross stream traffic than unmodified TCP (shown in Figure 9) in the presence of systemic and queuing packet loss. Since each fractional flow requires more packets to increase the flow's congestion window, it has a higher "virtual" round trip time compared to the actual round trip time of the cross stream flows. With a higher round trip time, the fractional flows are unable to compete with cross stream flows for bandwidth.

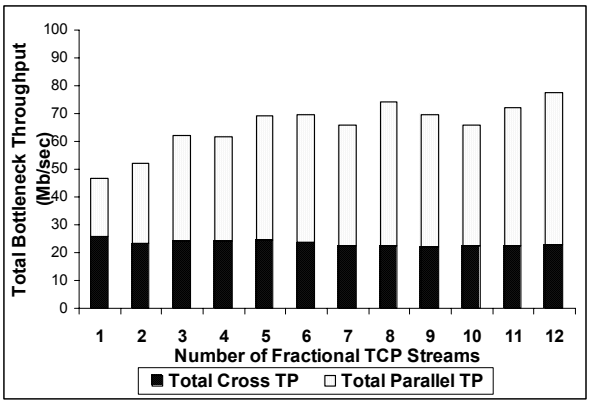

Figure 12.1 Fractional Parallel Flows with 1 Cross Stream in Congestion Simulation

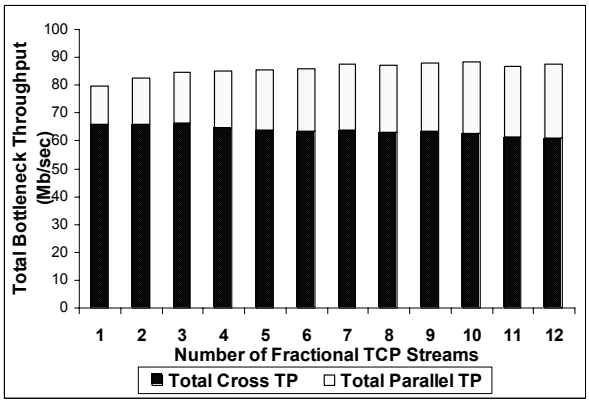

Figure 12.2 Fractional Parallel Flows with 5 Cross Streams in Congestion Simulation

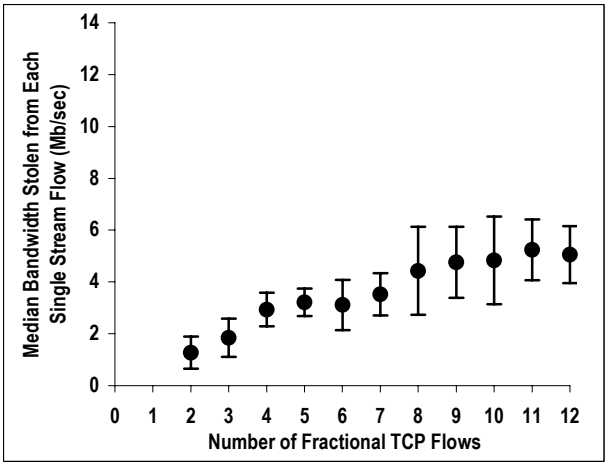

Figure 13. Median Bandwidth Stolen

Table 1 shows the percentage of loss of total cross single stream throughput for each congestion avoidance approach in the systemic loss simulation, calculated as the drop in median total cross stream throughput from 1 parallel flow (essentially a single stream flow) to 12 parallel flows. From Table 1, it is clear that the fractional approach is about half as aggressive as the set of single streams approach.

Figure 14 shows the best-case fractional single stream throughput from equation (5) with the worst observed fractional single stream throughput from the simulation. From these graphs, it is clear that as the number of streams increase, the ability of an individual fractional flow to steal bandwidth becomes progressively limited. For comparison, the throughput per stream for the unmodified congestion avoidance algorithm remains constant as the number of streams increase, up to the bottleneck limit.

Number Cross Streams	Unmodified	Fractional	Combined
1	32.58%	13.50%	11.21%
2	49.04%	16.32%	7.78 %
3	47.49%	26.13%	10.29%
4	48.43%	26.11%	5.98%
5	45.15%	18.64%	7.16%

Table 1. Percentage Decrease in Total Cross Stream Throughput

Note that the fractional approach results in progressively less aggressiveness as the number of parallel flows increase, and that there is a distinct knee in the cross stream throughput curve as the number of competing cross streams increase. Understanding the underlying mechanisms of the modified congestion algorithm that produce this effect is critical to determining how aggressiveness will scale with the number of streams.

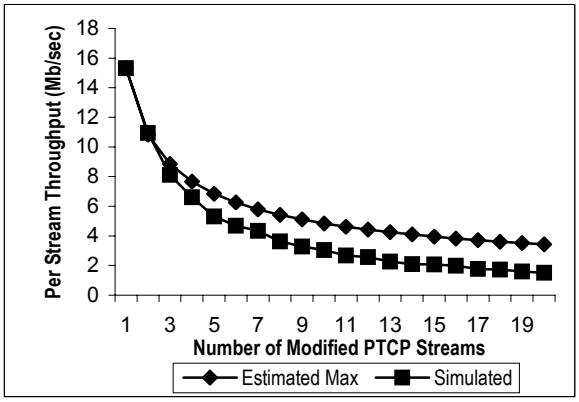

Figure 14. Best Case Fractional Throughput vs. Simulated Fractional Throughput

Finally, we ran the simulations for the combined approach. Figures 15.1 and 15.2 show the results for the systemic loss simulation.

With the combined approach, it is clear that the fractional streams are able to use unutilized bandwidth. More importantly, where there is no available bandwidth, the fractional flows lack the aggressiveness necessary to steal substantial amounts of bandwidth from competing single stream flows.

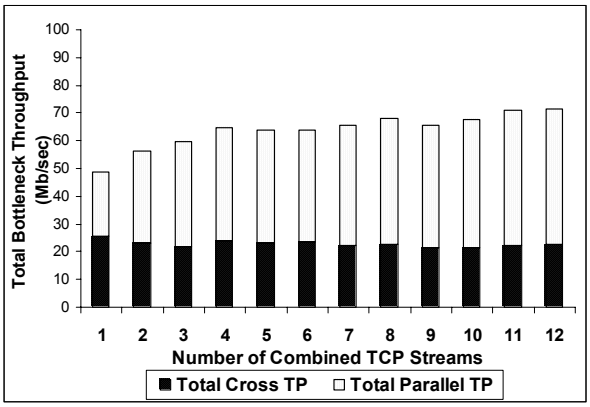

Figure 15.1 Combined Approach with 1 Cross Stream in Systemic Loss Simulation

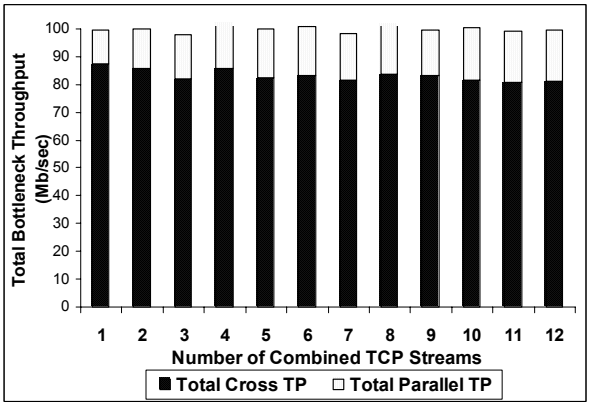

Figure 15.2 Combined Approach with 5 Cross Streams in Systemic Loss Simulation

Figure 16 shows the median amount of bandwidth stolen from competing single stream flows when the network is over 90% utilized. Comparing this figure with Figures 10 and 13, it is clear that the combined approach steals substantially less bandwidth than the other approaches. The single stream component of the combined parallel TCP flow, however, is able to secure at least a fair share (one single flow's worth) for the application. Thus, the combined approach fulfills the goal of providing effectiveness while simultaneously preserving fairness.

The combined approach in the congestive simulation is shown in Figures 17.1 and 17.2. Comparing these figures with the fractional approach shown in Figures 12.1 and 12.2, there is only a small difference in fairness between the fractional and combined approaches. The combined approach in Figure 17.2 steals approximately 0.4 Mb/sec more per cross stream than the fractional approach in Figure 12.2.

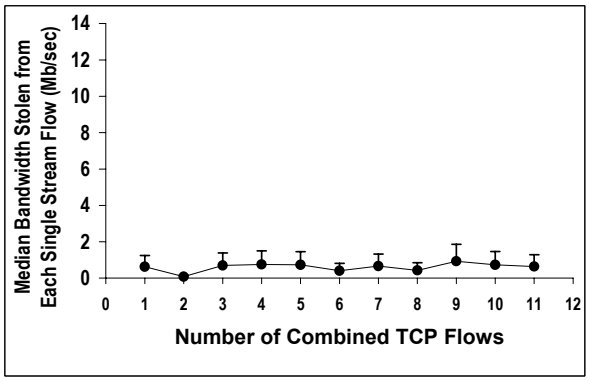

Figure 16. Median Bandwidth Stolen

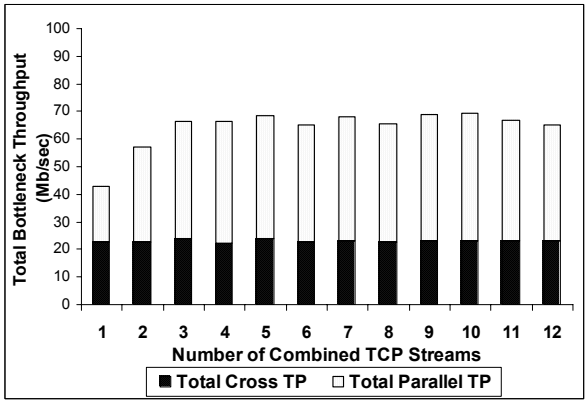

Figure 17.1 Combined Approach with 1 Cross Stream in Congestive Simulation

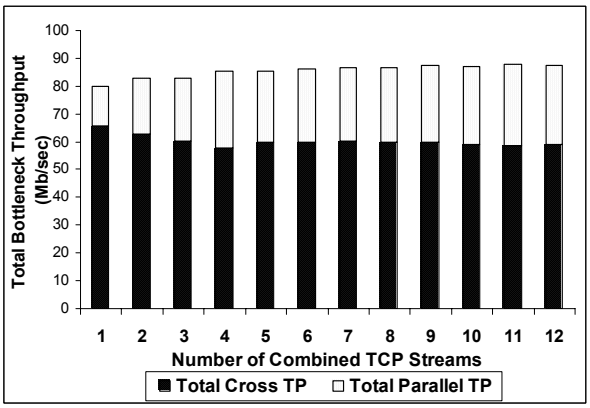

Figure 17.2 Combined Approach with 5 Cross Streams in Congestive Simulation

5.2.2 Fairness Conclusions

From the discussion in this section, we can conclude several things about fairness. First, when there is only systemic loss and sufficient unused bandwidth in the bottleneck, and the number of TCP flows does not fill the bottleneck beyond approximately 90% capacity, the use of parallel TCP flows is fair to competing streams, regardless of technique. Second, when loss arises from both systemic and queuing loss, parallel flows using unmodified TCP is unfair. Third, when the network bottleneck is near capacity, parallel TCP flows steal bandwidth from competing single stream flows. To alleviate the fairness problem at bottleneck capacity, two approaches to modifying the parallel flow response to congestion demonstrated improved fairness characteristics. The modified approach realized a partial improvement in fairness. The combined approach, with each fractional stream at 1/*js* the aggressiveness of a single stream, demonstrated greatly improved fairness characteristics, while maintaining reasonable effectiveness. The improvement in fairness is due to increasing the virtual round trip time of the fractional flows, thus decreasing the ability of the fractional flows to compete with unmodified single stream flows.

6.3 Efficiency

Chiu and Jain [ChJa89] define efficiency as the measure of the overall utilization of the network bottleneck. Obviously, higher utilization of the bottleneck is desirable, as long as congestion is avoided. The question of efficiency raised here is whether the use of parallel TCP connections, if used by all hosts, avoids congestion collapse.

To explore this, simulation experiments for the three approaches (unmodified, fractional, and combined) in which the number of parallel TCP connections was increased from 1 through 12 with the congestive simulation for all of the transmission nodes was performed. Figures 18 through 20 show the distribution of aggregate throughput for every node.

These figures show that as the number of parallel streams increases, the median aggregate throughput per node remains at approximately 14 Mb/sec. The results in Figures 18 through 20 demonstrate that fair sharing of the network bottleneck with other TCP streams takes precedence over effectiveness when every node uses parallel TCP connections. In addition, since the network bottleneck remains fully utilized as the number of parallel TCP connections used per node increases, when every node uses parallel TCP connections, efficiency is maintained. Thus, if every application uses parallel TCP flows, efficiency, fairness, and effectiveness are maintained, with priority given to fairness and efficiency over effectiveness.

Figures 19 and 20 show the aggregate throughput per node when the fractional and combined approaches are used for all parallel TCP streams. The median throughput for all fractional and combined categories is approximately 16 and 15 Mb/sec respectively.

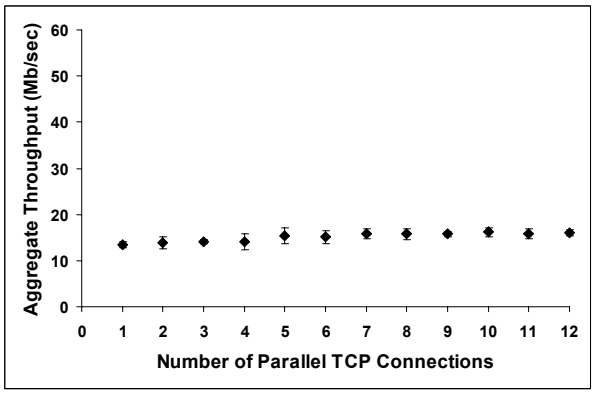

Figure 18. Aggregate Throughput on All Nodes with Unmodified TCP in Congestive Simulation

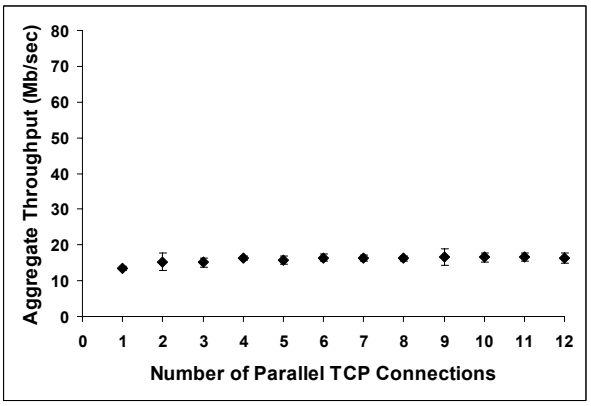

Figure 19. Aggregate Throughput on All Nodes with Fractional Approach in Congestive Simulation

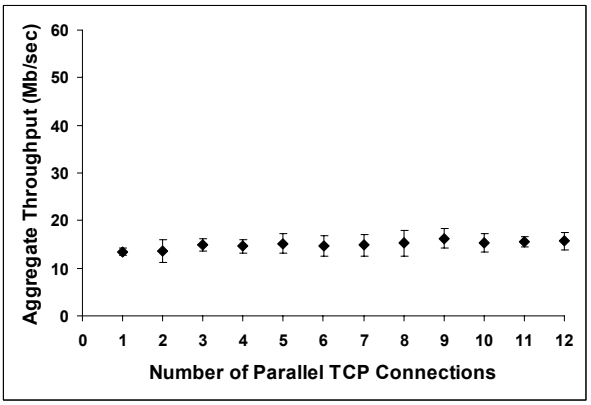

Figure 20. Aggregate Throughput on All Nodes with Combined Approach in Congestive Simulation

These results demonstrate that if all applications sharing a network bottleneck use fractional TCP streams with the same number of flows, efficiency and fairness characteristics are maintained, and congestion collapse will be avoided if all nodes utilized parallel TCP flows.

7 Conclusion

This paper addressed the questions of effectiveness, fairness, and efficiency when applications use parallel TCP connections to increase throughput. A series of simulations were performed to answer each of the questions. The results of the simulations demonstrated that parallel TCP connections are effective and efficient on networks in which drops are due to systemic packet loss or the combination of systemic and congestive loss. If the network bottleneck between the sender and receiver is underutilized, and only systemic loss is experienced, parallel TCP flows are fair. If, however, the network bottleneck is fully utilized, parallel TCP flows steal bandwidth from competing flows. A simple change to the aggressiveness of parallel flows was evaluated and found to substantially reduce aggressiveness of parallel flows while maintaining effectiveness.

The TCP congestion avoidance algorithm is a simple control system that is designed to ensure effective, fair, and efficient allocation of network bandwidth between competing TCP flows. Future work on congestion avoidance should take into account the existence of systemic packet loss to ensure that the goals of effective, fair, and efficient allocation of bandwidth are realized.

Acknowledgements

We would like to thank Dr. Atul Prakash of the Department of Electrical Engineering & Computer Science Department in the College of Engineering at the University of Michigan for suggesting the combined approach. This work described in this paper utilized the supercomputing resources of the Michigan Center for Biological Information and the Center for Advanced Computing at the University of Michigan. We would also like to thank the National Institutes of Health Visible Human Project (Grant NO1-LM-0-3511) for their continued support and encouragement.

REFERENCES

[Ab2001] Abilene Network Weather Map. http://hydra.uits.iu.edu/~abilene/traffic/

[AlAvBa2000] E. Altman, K. Avrachenkov, and C. Barakat, “A Stochastic Model of TCP/IP with Stationary Random Losses”, Proceedings of ACM SIGCOMM, August 2000, Stockholm, Sweden.

[AlBe2001] W. Allcock, J. Bester, A. Bresnahan, A. Chervenak, L. Liming, S. Tuecke, “GridFTP: Protocol Extensions to FTP for the Grid”, Global Grid Forum Draft.

[BaMo1998] C. Baru, R. Moore, A. Rajasekar, and M. Wan. “The SDSC Storage Resource Broker”, Proceedings of CASCON'98, Toronto, Canada, 1998.

[BaRaSe99] H. Balakrishnan, H. Rahul, S. Seshan, "An Integrated Congestion Management Architecture for Internet Hosts", Proceedings of ACM SIGCOMM, September 1999, Cambridge, MA, USA.

[BoGrHe99] J. Bolliger, T. Gross, and U. Hengartner, “Bandwidth Modelling for Network-aware Applications”, Proceedings of ACM INFOCOM '99, March 1999, New York.

[Bolot1993] J. Bolot. “End-to-End Packet Delay and Loss Behavior in the Internet”, Proceedings of ACM SIGCOMM 1993.

[BoSw1998] M. S. Borella, D. Swider, S. Uludag, G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS," Proceedings, International Conference on Parallel Processing, August 1998.

[ChJa89] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," Computer Networks and ISDN Systems, vol. 17, pp. 1--14, 1989.

[FerCa02] Personal Communication, January 2002. Tiziana Ferrari and Salavatore Cammarata, INFN, Bologna, Italy.

[FiHe1992] W. Fischer, K. Meier-Hellstern, “The Markov-modulated Poisson Process (MMPP) Cookbook”, Performance Evaluation, Vol 18, 1992.

[Fl2002] S. Floyd, “HighSpeed TCP for Large Congestion Windows,” IETF Internet Draft draft-floyd-tcp-highspeed-00.txt

[FlRa2002] S. Floyd, S. Ratnasamy, S. Shenker, “Modifying TCP’s Congestion Control for High Speeds”, Prepublication draft at http://www.icir.org/papers/hstcp.pdf.

[FoKe1997] I. Foster, C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit”, International Journal of Supercomputing Applications, 11(2): 115-128, 1997. http://www.globus.org.

[GaWa2000] M. Gates and A.Warshavsky, Iperf version 1.1.1, Bandwidth Testing

[HaAt2001] T. Hacker, B. Athey, B. Noble, "The End-to-End Performance Effects of Parallel TCP Sockets on a Lossy Wide-Area Network", Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS), April, 2002. Ft. Lauderdale, FL.

[HaTr2001] A. Hanushevsky, A. Trunov, L. Cottell, "Peer-to-Peer Computing for Secure High Performance Data Copying", Proceedings of Computing in High Energy and Nuclear Physics. Beijing, China, September, 2001.

[InTr2001] Internet Traffic Report. http://www.internettrafficreport.com/

[Jac88] V. Jacobson, "Congestion Avoidance and Control", In Proceedings of the ACM SIGCOMM '88 Conference, pages 314--329, August 1988, Stanford, CA

[JiSc2000] W. Jiang, H. Schulzrinne, "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality", 10th International Workshop on Network and Operating System Support for Digital Audio and Video. June, 2000, Chapel Hill, NC.

[Kos2001] David Koski, University of Michigan ITCS, Personal Communication.

[LeBaron1997] B. LeBaron. "A Fast Algorithm for the BDS Statistic", Studies in Nonlinear Dynamics and Econometrics, Volume 2, Number 2, 1997. MIT Press.

[Mah] B. Mah, "pchar: A tool for measuring internet path characteristics," http://www.employees.org/bmah/Software/pchar/.

[MaSe1997] M. Mathis, J. Semke, J. Mahdavi, T. Ott. "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, Volume 27, Number 3, July 1997.

[MiGoTo1999] V. Misra, W. Gong, D. Towsley., "Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior", In Proceedings of PERFORMANCE99, Istanbul, Turkey, 1999

[NiTier99] R. Nitzan and B. Tierney. "Experiences with TCP/IP over an ATM OC12 WAN", LBNL Report, Lawrence Berkeley National Laboratory, April 1999.

[NS2] VINT project U.C. Berkeley/LBNL, NS2 Network Simulator. http://www-mash.cs.berkeley.edu/ns/.

[PaFi1998] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and Its Empirical Validation", Proceedings of ACM SIGCOMM, September 1998, Vancouver, BC, Canada

[QiZhKe2001] L. Qiu, Y. Zhang, S. Keshav, "Understanding the Performance of Many TCP Flows", Computer Networks, 37(3-4), pp. 277-306, November 2001.

[RoAb2001] Personal Communication with Internet2 Abilene Network Operations Center and Bert Rossi, Merit Networks Senior Engineer.

[Slac2001] Internet End-to-End Performance Monitoring. http://www-iepm.slac.stanford.edu/

[StPa2000] S. Jonathan Stone, C. Partridge. "When The CRC and TCP Checksum Disagree", In Proceedings of ACM SIGCOMM, August 2000, Stockholm, Sweden.

[Tos2001] "Description and Resolution to Reported Packet Loss Problem with Toshiba PCX1100 Cable Modems"
http://www.cablemodemhelp.com/pcx1000.htm

[YaMoKu1999] M. Yajnik, S. Moon, J. Kurose, D. Towsley, "Measurement and Modelling of the Temporal Dependence in Packet Loss", In Proceedings of IEEE INFOCOM '99, pages 345-52, March 1999.

[ZhPaSh2000] Y. Zhang, V. Paxson, S. Shenker, and L. Breslau, "The Stationarity of Internet Path Properties: Routing, Loss, and Throughput", ACIRI Technical Report