# Long-Haul TCP vs. Cascaded TCP

W. Feng

## 1 Introduction

In this work, we investigate the bandwidth and transfer time of long-haul TCP versus cascaded TCP [5].

First, we discuss the models for TCP throughput. For TCP flows in support of bulk data transfer (i.e., long-lived TCP flows), the TCP throughput models have been derived [2, 3]. These models rely on the congestion-avoidance algorithm of TCP.

Though these models cannot be applied with short-lived TCP connections, our interest relative to logistical networking is in longer-lived TCP flows anyway, specifically TCP flows that spend significantly more time in the steady-state congestion-avoidance phase rather than the transient slow-start phase. However, in the case where short-lived TCP connections must be modeled, several TCP latency models have been proposed [1, 4] and based on these latency models, the throughput and transfer time of short-lived TCP connections are obtainable.

Using the above models, the transfer times for a data file of size $S$ packets can be computed for both long-haul TCP and cascaded TCP. The performance of both systems is compared via their transfer times. One system is said to be preferred if its tranfer time is lower than that of the other. Based on these performance comparisons, we develop a decision model that decides whether to use the cascaded TCP or long-haul TCP.

## 2 TCP Throughput Models

In this section, we describe TCP throughput and latency models that will be used to compute the transfer time in this paper. The discussion is separated for long-lived and short-lived TCP flows.

### 2.1 Long-Lived TCP Flows

The throughput models for long-lived TCP flows have been extensively studied [2, 3]. The two accepted models are developed by Mathis et al. [2] and Padhye et al. [3] and will be used in subsequent development of this paper. Before we summarize these models, we first start with some definitions: For a path in the network, let $D$ and $p$ be the round-trip time and packet loss probability, respectively, of the TCP connection using the path. Also, define $W_{max}$ and $T_0$ be the maximum window size and the average retransmission time out (RTO) of a TCP connection, respectively. Let $b$ denote the

number of acknowledged packets by the receiver per one ACK. For example, the original TCP implementation acknowledges every $b = 1$ packet and the newer implementation with delayed acknowledgements sends one ACK for roughly every $b = 2$ packets. These parameters are used in the following models.

- **Mathis Model** is a very simple approximation of TCP throughput which models only TCP congestion avoidance and fast retransmit. For some constant $C$, the TCP throughput is given by

$$B(D,p) = \frac{C}{D\sqrt{p}} \quad \text{packet/sec.} \tag{1}$$

  The typical value of $C$ is $\sqrt{\frac{3}{2b}}$.

- **Padhye Model** is significantly more complicated than the Mathis Model. It incorporates TCP fast retransmit and retransmission timeout into the congestion-avoidance model. The formula for TCP throughput is given by

$$B(D,p) = \min\{\frac{W_{max}}{D}, \frac{1}{D\sqrt{\frac{2bp}{3}} + T_0 \min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}\} \quad \text{packet/sec.} \tag{2}$$

Using the throughput formulas (1) and (2), the average transfer time of a long-lived TCP flow having data of size $S$ can be approximated by

$$T_L = \frac{S}{B(D,p)}. \tag{3}$$

Note that we neglect the slow-start phase because long-lived TCP flows stay most of their lifetimes in the congestion-avoidance phase. In this project, we will first use the Mathis Model for TCP thoughput in order to compare the performance of the long-haul TCP and that of the cascaded TCP.

## 2.2 Short-Lived TCP Flows

Because the throughput models for long-lived TCP flows are based on the congestion-avoidance algorithms of TCP, they cannot be applied to short-lived TCP flows which are so short that their entire lifetimes are usually within the slow-start phase of TCP. Here, we rely on the TCP latency model proposed by [1].

According to [1], the expected transfer time of TCP connections can be partitioned into three intervals, i.e.,

$$T_S = \mathbf{E}[T_{ss}] + \mathbf{E}[T_{loss}] + \mathbf{E}[T_{ca}]$$

where $T_{ss}$, $T_{loss}$ and $T_{ca}$ are time durations in slow-start mode, loss-recovery after timeout, and congestion-avoidance mode. [1] Formulas for these time durations are available in [1].

[1]Note that here we drop the delay due to the delayed acknowledgement scheme of the first segment. This delay is constant for each operating system.

By computing $T_{avg}$ for long-haul TCP and cascaded TCP, their performance can be compared.

Note that the formula for $T_S$ is based on the derivation of [3] and is very complicated to compute. If the TCP transfer size is large, then it is wiser to approximate the transfer time using the steady state (long-lived TCP) model. But how large is large?

# 3 No Pipelining

The derivations in the next two sections rely on the Mathis TCP model (1). Suppose that the long-haul TCP has round-trip time $D$ and packet-loss probability $p$. From (1), the bandwidth of this connection is approximately $B(D,p) = \frac{C}{D\sqrt{p}}$ and the transfer time for data of size $S$ is simply

$$T_{lh} = \frac{S}{B(D,p)} = \frac{SD\sqrt{p}}{C}. \tag{4}$$

Now, if this data is transmitted using another path and for this path the TCP connection is broken into $N$ cascaded TCP connections where each TCP connection is denoted by TCP $i$, $i = 1, \ldots, N$, then for each $i = 2, \ldots, N$, TCP $i$ can start its transmission if TCP $i-1$ has finished all data transfer. The transmission for the cascaded TCP is said to be complete when the last TCP connection has finished its transfer. Using this idea, the transfer time of data of size $S$ is

$$T_c = \sum_{i=1}^{N} T_i$$

where for each $i = 1, \ldots, N$, $T_i$ is the transfer time of TCP $i$. For each $i = 1, \ldots, N$, let $D_i$ and $p_i$ denote the round-trip time and the packet-loss probability for TCP $i$, respectively. By applying Mathis' formula (1) to each TCP $i$, then

$$T_c = \sum_{i=1}^{N} \frac{SD_i\sqrt{p_i}}{C}. \tag{5}$$

From the above computation, we prefer the cascaded TCP when the transfer time $T_c$ is smaller than the original transfer time of long-haul TCP. Thus, from (4) and (5), we choose the cascaded TCP when

$$\sum_{i=1}^{N} \frac{SD_i\sqrt{p_i}}{C} < \frac{D\sqrt{p}}{C},$$

or equivalently,

$$\sum_{i=1}^{N} D_i\sqrt{p_i} < D\sqrt{p}.$$

Note that the decision does not depend on the transfer size $S$ but only on the statistics of the paths, namely the round-trip delay and the loss probability.

# 4 Pipelining

In this section, we relax the assumption on the transmission of the cascaded TCP by allowing pipelining. Pipelining means that the intermediate node can transmit the packet whenever it has the packet in its queue. So it does not have to wait until all the packets have arrived in order to start the transmission. However, in this transmission process, the incoming data will be buffered in the sending queue. To avoid packet drought in the buffer, the TCP connection may wait for $R$ transmission windows before starting its transmission.

For this case, the bottleneck link is simply the link with lowest TCP bandwidth. Using the Mathis Model (1), the bottleneck link is simply the link

$$k = \arg\min\{i = 1, \ldots, N : D_i\sqrt{p_i}\}.$$

Therefore, the total transfer time of the pipelining cascaded TCP can be approximated by

$$T_{pc} = \frac{S}{B(D_k, p_k)} + R\sum_{i=1}^{N} D_i \tag{6}$$

where the first term is the bottleneck link transfer time and the second term comes from the first $R$ end-to-end round-trip times that each connection waits before starting its transmission. By comparing (4) and (6), the pipelining cascaded TCP is preferred if $T_{pc} < T_{lc}$, or equivalently,

$$D_k\sqrt{p_k} + \frac{CR}{S}\sum_{i=1}^{N} D_i. < D\sqrt{p}$$

Here, the transfer size $S$ is involved in the decision. But it will has small effect if $CR\sum_{i=1}^{N} D_i \ll S$ and the decision in this case is roughly

$$D_k\sqrt{p_k} < D\sqrt{p}.$$

# 5 Note

From the operating standpoint, the cascaded TCP requires more overhead than the original TCP since it needs to establish $N$ TCP connections instead of only 1 TCP connection. In computing the decision, one might need to add the processing time $T_{proc}$ for cascaded TCP which depends on the number of TCP connections $N$ and (maybe) on the transfer size $S$.

Short-lived TCP still needs to be considered.

# References

[1] N. Cardwell, S. Savage and T. Anderson, "Modeling TCP latency," in *Proceeding of IEEE INFOCOM 2000*, March 2000, Tel Aviv, Israel.

[2] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review* **27**, 1997.

[3] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in Proceedings of ACM SIGCOMM 1998.

[4] B. Sikdar, S. Kalyanaraman and K.S. Vastola, "An integrated model for the latency and steady-state throughput of TCP connections," *Performance Evaluation* **46**, September 2001, pp. 139-154.

[5] M. Swany and R. Wolski, "Improving throughput with cascaded TCP connections: the logistical session layer," *Technical Report 2002-24*, University of California, Santa Barbara.