Rethinking TCP Throughput and Latency Modeling

Yi Cao
Stony Brook University
yicao1@cs.stonybrook.edu

Aruna Balasubramanian
Stony Brook University
arunab@cs.stonybrook.edu

Anshul Gandhi
Stony Brook University
anshul@cs.stonybrook.edu

ABSTRACT

TCP throughput and latency models are useful tools to characterize the TCP performance. The canonical throughput model [2], while useful, has some limitations since it does not consider how packet loss rate changes over time. This approach leads to poor predictions for short flows. We present a new modeling approach that characterizes the throughput and latency models by: (i) discovering the relationship between the packet loss rate and the congestion window size, and (ii) incorporating the starting congestion window and the number of parallel connections. Experimental results show that our models significantly improve modeling accuracy.

CCS CONCEPTS

• **Networks** → **Transport protocols**; **Network performance modeling**;

KEYWORDS

Network measurements, transport layer protocols

ACM Reference format:
Yi Cao, Aruna Balasubramanian, and Anshul Gandhi. 2017. Rethinking TCP Throughput and Latency Modeling. In *Proceedings of SIGCOMM Posters and Demos '17, Los Angeles, CA, USA, August 22–24, 2017,* 3 pages.
https://doi.org/10.1145/3123878.3131995

1 INTRODUCTION

Over the past few years, various TCP protocols have been proposed to meet specific needs. Subsequently, researchers have been interested in building models to analyze their performance. We observe via experiments that existing TCP throughput and latency models do not work well for short flows (see Table 1). Given that TCP models are important in many applications, for example, in predicting the page load time, we analyze TCP performance and propose a better performance model to address the shortcomings of existing models.

2 DRAWBACKS OF EXISTING MODELS

There are **three problems** with existing TCP throughput and latency models:

1) Existing models [1–3] assume that the packet loss rate (p) is independent of the congestion window size ($cwnd$). However, this property does not always hold in practice. Figure 1 shows the empirical $p - cwnd$ function plots in our experiments (see §3.1 for details). Here, each data point represents the average per-connection packet loss rate given a congestion window and number of parallel connections. Clearly, p is not independent of $cwnd$. The observed $p - cwnd$ relationship can be explained as follows: when $cwnd$ is low, the network condition is usually poor (leading to low $cwnd$), so p tends to be high; when $cwnd$ is high, the client is sending packets at a high rate, resulting in a congested network and high p; between these extremes (moderate $cwnd$), the network usage is fair, so p is relatively low.

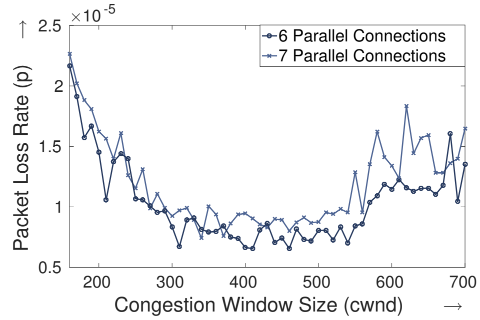

Figure 1: Empirical *p-cwnd* plot.

2) Existing models do not take the number of parallel connections (npc) into account. We observe in experiments that p is related to the number of parallel connections. In Figure 1, we see that the more parallel connections we have, the higher is the packet loss rate.

3) Existing models ignore the initial $cwnd$ ($scwnd$) when the transfer starts. While $scwnd$ does not significantly affect performance for long flows, it is of critical importance for short flows since it effectively determines the range of $cwnd$s encountered during the transfer.

3 OUR MODELING APPROACH

We design new TCP throughput and latency models that address the above-mentioned problems.

3.1 Experimental Setup

We use iPerf to transfer packets with *TCP Reno* from the sender in New York to the receiver in Seattle for 5 hours. We vary the number of parallel TCP connections between the sender and the receiver between 1 and 10, and run 5 sets of 1-hour experiments for each setting. In each experiment, we use the linux *tcpprobe* module to record the $cwnd$ values when sending packets, and use this logged information to determine the losses and the packet loss rate, p. Using this data, we obtain, for each setting of number of parallel connections (npc), the empirical relationship between packet loss

SIGCOMM Posters and Demos '17, August 22–24, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5057-0/17/08.
https://doi.org/10.1145/3123878.3131995

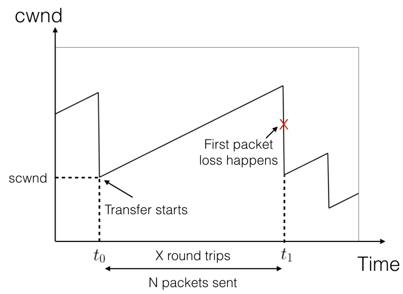

Figure 2: Network in congestion avoidance state.

rate and congestion window size. For convenience, we denote this empirical relationship as $p(cwnd, npc)$.

3.2 The Throughput Model

Let k be the number of parallel TCP connections. Without loss of generality, we consider the i_{th} connection ($i \in \{1, \cdots, k\}$) and analyze its throughput. We restrict our analysis to modeling the TCP congestion avoidance phase, similar to prior works [2].

Figure 2 shows how the congestion window changes in the congestion avoidance state. Here, t_0 denotes the data transfer start time and $scwnd$ denotes the starting $cwnd$ size. The $cwnd$ keeps increasing linearly (under TCP Reno) until a packet loss occurs at time t_1.

To derive expected throughput, we first derive $E[X]$, the expected number of RTTs between t_0 to t_1. Since each RTT corresponds to one $cwnd$, we have

$$E[X] = \sum_{n=1}^{\infty} n \cdot q(n, k),$$

where $q(n, k)$ is the probability of the first packet loss occurring during the n^{th} $cwnd$ after the transfer starts. To derive $q(n, k)$, we first derive the probability that there is no packet loss in a $cwnd$ of size s given k connections as $\overline{q}_{s,k} = (1 - p(s, k))^s$, where $p(s, k)$ is the empirical loss rate observed for a $cwnd$ of size s given k connections, as defined in §3.1. Thus, the probability of a loss in a $cwnd$ of size s is $q_{s,k} = 1 - \overline{q}_{s,k}$. Now, $q(n, k)$ can be expressed in terms of $q_{s,k}$ as

$$q(n, k) = q_{scwnd+n-1,k} \cdot \prod_{l=1}^{n-1} \overline{q}_{scwnd+l-1,k}.$$

Finally, $E[X]$ can be obtained as:

$$\begin{aligned} E[X] &= \sum_{n=1}^{\infty} n \cdot q(n, k) \\ &= \sum_{n=1}^{\infty} n(1 - (1 - p(scwnd + n - 1, k))^{scwnd+n-1}) \\ &\quad \times \prod_{l=1}^{n-1} (1 - p(scwnd + l - 1, k))^{scwnd+l-1} \end{aligned} \quad (1)$$

$E[N]$, the expected number of packets sent during t_0 to t_1, can be similarly derived. The expected throughput of the i_{th} connection can then be derived as:

$$B_i(scwnd, p(cwnd, k)) = E[N]/((E[X] + 1) \cdot RTT), \quad (2)$$

Finally, the total throughput can be obtained by summing the individual throughputs of the k connections:

$$B = \sum_{i=1}^{k} B_i(scwnd, p(cwnd, k)) \quad (3)$$

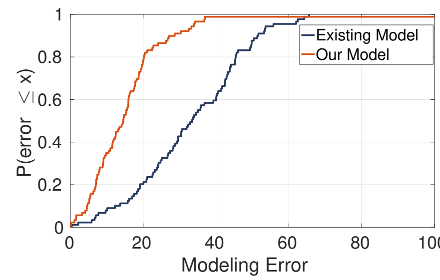

Figure 3: Error CDF of throughput prediction.

3.3 The Latency Model

Based on the TCP throughput model, we build the latency model to calculate the file (or data) transfer time. Due to space limitations, we omit details and present the latency model as:

$$L = f(npc, scwnd, p(cwnd, npc), fileSize, RTT) \quad (4)$$

Intuitively, the latency model is obtained by determining the time required to transfer the given file (with size $fileSize$) across k parallel connections; the throughput for each connection is obtained via Eq. (3) above.

4 RESULTS

We validate our model by comparing its throughput and latency predictions with experimentally observed values and reporting the accuracy. We also report the accuracy of existing models [2] for further comparison.

We transfer 100 files from the sender in New York to the receiver in Seattle. Table 1 shows our modeling results. Our modeling error for throughput and latency is about 17% and 13%, respectively. Compared to existing models, we reduce the throughput prediction error by about 50% and the latency prediction error by about 65%. Figure 3 shows the CDF of throughput modeling error for all 100 flows for high $scwnd$. We see that more than 80% of the errors for our model are below 20%. By contrast, only about 20% of the errors of the existing model are below 20%. The results for latency predictions are similar.

Model	Throughput	Latency
Existing Model [2]	33.2%	37.5%
Our Model	16.6%	13.0%

Table 1: Average throughput and latency modeling error of 100 short flows (≤ 10MB).

5 FUTURE WORK

We plan to build extended models to evaluate the performance of the HTTP/2 protocol. Specifically, our proposed models will aim to: (i) accurately predict the page load time, and (ii) find the optimal TCP flow to transfer data when multiple flows are available.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation through the grant CNS-1551909, CNS-1566260, CNS-1617046, CNS-1464151, and a Google Research Grant R2-2015-839.

REFERENCES

[1] Neal Cardwell, Stefan Savage, and Thomas Anderson. 2000. Modeling TCP latency. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 3. 1742–1751.

[2] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. 1998. Modeling TCP throughput: A simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review* 28, 4 (1998), 303–314.

[3] Nadim Parvez, Anirban Mahanti, and Carey Williamson. 2010. An analytic throughput model for TCP NewReno. *IEEE/ACM Transactions on Networking (ToN)* 18, 2 (2010), 448–461.