AN OVERVIEW AND COMPARISON OF ANALYTICAL TCP MODELS

Inas Khalifa and Ljiljana Trajković

School of Engineering Science, Simon Fraser University
Vancouver, British Columbia, Canada
{ikhalifa, ljilja}@cs.sfu.ca

ABSTRACT

Modeling TCP performance is an important issue that attracted research attention over the past decade. In this paper, we present an overview of models used to capture the TCP behavior. We compare several existing analytical models with respect to modeled attributes, modeling assumptions, and validation techniques. We also identify features that new TCP models should possess. Finally, we address the importance of devising common validation techniques and performance evaluation metrics for TCP models.

1. INTRODUCTION

Recent traffic measurements in several large campus networks indicated that over 90% of network traffic is transported by the Transmission Control Protocol (TCP) [3]. This ubiquity has encouraged research on modeling TCP behavior. Several analytical TCP models have recently been proposed [2], [3], [5], [6], [7], [9]. TCP models may be used to improve the existing and evaluate new congestion control algorithms and TCP implementations. For example, it has been demonstrated that TCP SACK outperforms TCP Tahoe when segment losses are independent, while Tahoe outperforms SACK when segment losses are correlated [9]. They may also be used to define TCP-friendly behavior or investigate the interaction between TCP and queue management algorithms [4].

In this paper, we examine TCP models and present a framework for model comparison with respect to the modeling assumptions and validation techniques. We also address the lack of common validation approaches and evaluation metrics. In Section 2, we present an overview of TCP. Model classification is discussed in Section 3. Section 4 describes existing analytical TCP models, while Section 5 compares the modeling assumptions and model validation techniques. Conclusions are given in Section 6.

2. OVERVIEW OF TCP

TCP provides a reliable connection-oriented data service in packet-switched networks. This is achieved by employing acknowledgments (ACKs), sequence numbers, and timers. TCP employs window-based congestion control mechanisms to adjust its congestion window size ($cwnd$). $cwnd$ is the maximum amount of data that the sender can transmit before receiving an ACK. The receiver also advertises a limit ($rwnd$) on the amount of outstanding data. Data transmission is always governed by the window size $W_m = \min(cwnd, rwnd)$.

A client initiates a TCP connection by sending a SYN segment to the server. The connection is established using a three-way handshake. The sender then uses the *slow start* algorithm to detect the available bandwidth by incrementing $cwnd$ by one segment upon receipt of each ACK. When $cwnd$ reaches the slow start threshold ($ssthresh$), the sender enters *congestion avoidance* phase and $cwnd$ is incremented by one segment per round trip time (RTT).

If a segment is not acknowledged within the retransmission timeout interval (RTO), the sender infers that the segment is lost (TO loss). It retransmits the lost segment, doubles the RTO (exponential backoff), and switches to slow start. Segment loss may also be detected using triple duplicate ACKs (TD losses). When TD losses occur, the sender uses the *fast retransmit* algorithm: it halves the $cwnd$ and retransmits the missing segment without waiting for RTO to expire. The *fast recovery* algorithm is then used to control data transmission until a non-duplicate ACK is received [1].

A TCP receiver uses the delayed acknowledgment algorithm. It may acknowledge b segments with one ACK, where $b \geq 1$. ACKs are generated: (i) for at least every second full-sized segment of new data, (ii) within 500 ms upon the arrival of an unacknowledged segment, (iii) for out-of-order segments, to trigger fast retransmit, and (iv) for segments that fill gaps in the sequence number space.

3. MODEL CLASSIFICATION

An IP-based communication network employs routers with queue management mechanisms to enable data transfers. TCP delivers byte streams between pairs of hosts. TCP models investigate the network from three different perspectives. Models in the first class consider queue management

and they characterize TCP flows by their average throughput [4]. Models in the second class consider the interaction between TCP flows and queue management mechanisms. They typically assume simple topologies, number of flows, and direction of data flow [8]. Models in the third class deal with TCP dynamics. They are the main focus of this paper. These models consider the network from the perspective of the TCP layer and characterize it by parameters such as loss, average drop probability p, and average RTT [5], [7].

Analytical TCP models can also be classified according to the length of the TCP transfer, which determines the congestion control algorithms and loss detection mechanisms that need to be incorporated into the model. In case of *short-lived* transfers, TCP performance is strongly affected by the connection establishment and slow start phases, with segment losses mostly being TO losses. Models for *long-lived* transfers capture the steady-state performance of TCP, which is dominated by the congestion avoidance phase, and losses may be TD as well as TO. There are also models for TCP transfers of *arbitrary* length.

4. ANALYTICAL TCP MODELS

We illustrate the evolution of modeling techniques by surveying six analytical TCP models. We define three TCP performance attributes: *throughput* T as the total amount of data sent by a TCP source per unit time (including retransmissions), *goodput* G as the total amount of data correctly received per unit time, and *latency* L as the time required to successfully complete a TCP transfer.

4.1. Models for Long-Lived Transfers

Mathis, et al., [5] developed a model that predicts the steady-state throughput of long-lived TCP transfers in the presence of light to moderate segment losses. The model considers the congestion avoidance phase and assumes no TO losses. The segment loss process is periodic, with a constant probability p. In other words, every segment loss is followed by the successful delivery of $\frac{1}{p}$ segments. Therefore, the evolution of $cwnd$ follows a periodic sawtooth pattern during equilibrium. The number of segments transmitted during each period is $\frac{1}{p}$. Given a maximum window size of W_m, the minimum value of $cwnd$ is $\frac{W_m}{2}$. Moreover, if the receiver acknowledges every segment, then $cwnd$ is incremented by one segment every RTT. Hence, the duration of each period is $\frac{W_m}{2} \times RTT$. The area under the sawtooth, $(\frac{W_m}{2})^2 + \frac{1}{2}(\frac{W_m}{2})^2$, is also equal to $\frac{1}{p}$. Thus, the throughput T is:

$$T = \frac{MSS}{RTT}\frac{K}{\sqrt{p}}, \tag{1}$$

where MSS is the maximum segment size and K is a constant that depends on the loss model and the acknowledgment strategy. For example, $K = \sqrt{3/2}$ when loss is periodic and delayed acknowledgment is not employed.

Padhye, et al., [7] developed a stochastic model for the steady-state throughput of long-lived bulk transfer TCP flows. The behavior of TCP congestion control is modeled in terms of *rounds*. A round starts with the transmission of $cwnd$ segments and ends when the first ACK is received. Hence, the duration of a round is equal to RTT. It is assumed to be independent of $cwnd$. The model considers both TO and TD losses and $cwnd$ limitation by W_m. Moreover, it assumes a correlated (*bursty*) loss model: if a segment is lost, all remaining segments in the same round are also lost. Stochastic systems techniques are used to determine the expected values of the number of segments transmitted in a round and the duration of the round in terms of the loss probability p. The throughput is approximated as:

$$T = \min\left(\frac{W_m}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + RTO_0 \min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}\right), \tag{2}$$

where RTO_0 is the initial value of RTO. The model can be extended to capture the goodput G of the transfer by computing the number of received, instead of transmitted, segments [7].

Altman, et al., [2] proposed a model for the throughput of long-lived TCP transfers, subjected to a stationary ergodic loss process. The dynamics of TCP are modeled by observing the instantaneous transmission rate $X(t)$, defined as the number of packets in the network (window size) divided by RTT at time t. The instants Y_n when loss events occur are modeled by a stationary ergodic point process $\{Y_n\}_{n=-\infty}^{+\infty}$. The inter-loss duration S_n is equal to $(Y_{n+1} - Y_n)$ when loss is TD, and $(Y_{n+1} - Y_n - E[t_{TO}])$ when loss is TO, where $E[t_{TO}]$ is the average duration of the timeout period. The throughput T, computed as the time average of the process $X(t)$, is:

$$T = \frac{1}{RTT\sqrt{bp}}(1 - \lambda_{TO}E[t_{TO}]) \times \sqrt{\frac{1+\nu}{2(1-\nu)} + \frac{1}{2}\hat{C}(0) + \sum_{k=0}^{\infty}\frac{1}{2^k}\hat{C}(k)}\,, \tag{3}$$

where $\hat{C}(k) = (E[S_nS_{n+k}] - E[S_n]^2)/E[S_n]^2$ is the normalized covariance, ν is the factor used to reduce $cwnd$ when loss occurs, and λ_{TO} is the number of TO losses per unit time. When all losses are TD and random, (3) becomes identical to (1). When $W_m \neq \infty$, an explicit expression for T cannot not be obtained. Expressions for the lower and upper bounds on T are derived, instead.

4.2. Models for Transfers of Arbitrary Length

Cardwell, et al., [3] developed a model for the latency of TCP transfers of arbitrary length by extending [7] to include connection establishment and initial slow start. The

expected latency $E[L_{CE}]$ of the three-way handshake used during the connection establishment (CE) phase is:

$$E[L_{CE}] = RTT + RTO_0\left(\frac{1-p_r}{1-2p_r} + \frac{1-p_f}{1-2p_f} - 2\right), \quad (4)$$

where p_f is the segment loss rate in the forward path from the server to the client, and p_r is the loss rate in the reverse path. The data transfer latency $E[L]$ required to complete a transfer of size N segments is computed as the sum of four components:

$$E[L] = E[L_{ss}] + E[L_{loss}] + E[L_{ca}] + E[L_{delack}], \quad (5)$$

where L_{ss} is the latency of the initial slow start, E_{loss} is the expected cost of TO losses or fast recovery that occurs at the end of the initial slow start, L_{ca} is the expected time required to transfer the remaining $(N - E[d_{ss}])$ segments, and L_{delack} is the cost of the first delayed ACK if $cwnd_0$ is equal to 1. Both $E[L_{ss}]$ and $E[L_{loss}]$ are functions of RTT, b, W_m, p, N, $E[t_{TO}]$, and the initial window size $cwnd_0$. $E[L_{ca}] = (N - E[d_{ss}])/T$, where T is the throughput [7]. $E[d_{ss}]$ is the expected number of segments sent before a loss is encountered during the initial slow start. Finally, $E[L_{delack}]$ is equal to 150 ms for Windows platforms or 100 ms for BSD UNIX.

Sikdar, et al., [9] modeled the latencies of arbitrary length transfers of TCP Tahoe, Reno, and SACK by estimating the transfer time given that the transfer experiences no loss, a single loss, and multiple loss indications. The expected latency to transfer N segments is:

$$\begin{array}{l} E[L(N)] = E[L_{CE}] + E[L_{delack}] + E[L_{ml}(N)] \\ +(1-p)^N E[L_{nl}(N)] + p(1-p)^{N-1} E[L_{sl}(N)], \end{array} \quad (6)$$

where $E[L_{CE}]$ is given in (4) with $p_f = p_r = p$, and $E[L_{delack}]$ is as given in [3]. $E[L_{ml}(N)]$ is the expected latency with M loss indications. It is a function of N, RTT, W_m, $E[t_{TO}]$, $cwnd_0$, and M. $E[L_{nl}(N)]$ is the expected latency when no losses occur. It is a function of N, RTT, and W_m. $L_{sl}(N)$ is the expected latency when there is a single loss and it is a function of N, RTT, W_m, $E[t_{TO}]$, and $cwnd_0$. The last three terms in (6) are computed for TCP Tahoe, Reno, and SACK. The steady-state throughput of long-lived transfers is:

$$T = \frac{d \times MSS}{E[L_{ss}(p)]}, \quad (7)$$

where $d = \frac{1}{p}$ is the average number of segments transmitted between two loss indications during steady-state. $E[L_{ss}(p)]$ is the expected time to transfer d segments. It is calculated by averaging the expected time to transmit d segments in the presence of multiple losses over all possible values of $cwnd$ and all possible positions of loss indications within the window.

4.3. A Model for Short-Lived Transfers

Mellia, et al., [6] devised a recursive analytical model for the average latency of short-lived TCP transfers during the slow start phase by exhaustively enumerating loss scenarios and their probabilities. The connection establishment latency $E[L_{CE}]$ is calculated using (4) with $p_f = p_r = p_s$, where p_s is the SYN segment dropping probability. The latency is computed using L_m^w, defined as the average time spent to successfully send m segments with an initial $cwnd$ of w. L_n^1 is the average time required to transfer n segments with $cwnd_0 = 1$. L_n^1 is equal to $(L_1^1 + L_{n-1}^2)$. It is derived *recursively* as a function of p, RTO, RTT, and $L_{m'}^{w'}$ where $m' < m$. L_m^m is the average time required to transfer m segments that belong to the same window. For example,

$$\begin{array}{rcl} L_1^1 & = & RTT + RTO\frac{p}{1-2p} \\ L_2^2 & = & RTTq^2 + qp(RTO + RTT + L_1^1) \\ & & + pq(RTO + L_1^1) + p^2(RTO + L_2^1) \end{array} \quad (8)$$

where $q = 1 - p$, and p is uniformly distributed. L_m^w is computed recursively up to $m = 9$, where the data transfer time is $L_9^1 = L_1^1 + L_{9-1}^2$. The procedure may be repeated further. However, the complexity grows exponentially since all loss scenarios need to be considered.

5. ASSUMPTIONS AND MODEL VALIDATION

The modeling assumptions for the surveyed models are summarized in Table 1. They are classified in three categories [3].

Data Transfer length and congestion control algorithms affect TCP performance. Table 1 shows that neither slow start after TO losses nor fast retransmit are considered, although TO losses are common [7]. ISS denotes the initial slow start performed after connection establishment.

End Points assumptions deal with the TCP implementation, the number of segments b acknowledged by a single ACK, and whether or not the window limitation W_m is taken into account. They also include the loss detection mechanism and the duration of RTO. Note that b alone is not sufficient to describe the complex delayed ACK mechanism. Other common assumptions are the use of full-sized segments, *greedy* sources that generate segments as fast as $cwnd$ allows, and that both Nagle algorithm and silly window syndrome are neglected.

Network assumptions deal with the loss process seen by data and ACK segments. No model assumes a specific topology, a data flow direction, or a queue management algorithm. All models neglect transmission and processing delays and assume that RTT is independent of $cwnd$.

Validation is the process of evaluating how accurately a model reflects the real-world phenomena that it tries to capture. Models can be validated by comparing their performance with results from: (i) simulations, (ii) controlled

Table 1. Summary of the modeling assumptions. The TCP algorithms are Connection Establishment (CE), Initial Slow Start (ISS), Congestion Avoidance (CA), and Fast Recovery (FRC). "Exp." indicates that exponential backoff is employed.

Model	Length	TCP algorithms	b	W_m	Loss detection	RTO	Data/ACK loss
Mathis [5]	Long-lived	CA	1, 2	Yes	TD	None	Periodic/None
Padhye [7]	Long-lived	CA	Any	Yes	TO, TD	Exp.	Bursty/None
Altman [2]	Long-lived	CA	Any	No	TO, TD	Exp.	Stationary/None
Cardwell [3]	Arbitrary	CE, ISS, CA	2	Yes	TO, TD	Exp.	Bursty/SYN only
Sikdar [9]	Arbitrary	CE, ISS, CA, FRC	2	Yes	TO, TD	Exp.	Bursty/SYN only
Mellia [6]	Short-lived	CE, ISS, FRC	1	No	TO, TD	Exp.	Uniform/SYN only

Table 2. Model validation techniques. The subscripts m and t denote model prediction and trace measurement, respectively.

Model	Simulations	Controlled meas.	Live meas.	Comparison	Evaluation Metric
Mathis [5]	Yes	Yes	No	None	Least mean squared fit error
Padhye [7]	Yes	Yes	No	None	$(T_m - T_t)/T_t$
Altman [2]	No	Yes	No	[7]	None
Cardwell [3]	Yes	Yes	Yes	[5], [7]	$(L_m - L_t)/L_t$, $(L_m - L_t)/RTT$
Sikdar [9]	Yes	No	Yes	[3], [7]	$(L_m - L_t)/L_t$
Mellia [6]	Yes	No	No	None	None

measurements under conditions that reflect the modeling assumptions, and (iii) live measurements from the Internet.

Table 2 shows that most models were neither validated using live measurements, nor compared to other models. Several models use evaluation metrics to measure how they deviate from the results used for validation. These metrics only represent the average relative error between the model prediction and the traffic measurement. With such a diversity of validation methods, traffic measurements, and evaluation metrics, it is rather difficult to compare the models.

6. CONCLUSIONS

In this paper, we presented an overview of TCP models. We surveyed a number of analytical TCP models and showed that in all models, T is inversely and L is directly proportional to $RTT, p, RTO, E[t_{TO}]$. We compared models with respect to the modeling approaches, assumptions, and validation techniques. The comparison indicated that new models should include: slow start after TO losses, fast recovery, and more accurate capture of delayed acknowledgment. In order to demonstrate their merits, new models need to be compared to existing models based on a common set of traffic measurements. Such quantitative comparisons could be even more valuable if a comprehensive set of evaluation metrics were identified. These metrics need to reflect model predictions over a range of network variables and parameters, transfer lengths, and geographical network span.

7. REFERENCES

[1] M. Allman, V. Paxson, and W. Steven, "TCP congestion control," RFC 2581, Apr. 1999.

[2] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," *ACM Computer Communication Review*, vol. 30, no. 4, pp. 231–242, Oct. 2000.

[3] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proc. INFOCOM*, Mar. 2000, pp. 1742–1751.

[4] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proc. INFOCOM*, Mar. 2000, pp. 1435–1444.

[5] M. Mathis, J. Semke, Jamshid Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, Jul. 1997.

[6] M. Mellia, I. Stoica, and H. Zhang, "TCP model for short lived flows," *IEEE Communications Letters*, vol. 6, no. 2, pp. 85–87, Feb. 2002.

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP reno performance: A simple model and its imperical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–145, Apr. 2000.

[8] R. Roy, R. C. Mudumbai, and S. S. Panwar, "Analysis of TCP congestion control using a fluid model," in *Proc. ICC*, Mar. 2001, pp. 2396–2403.

[9] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models and comparative study of the latency and steady-state throughput of TCP Tahoe, Reno and SACK," in *Proc. GLOBECOM*, Nov. 2001, pp. 1781–1787.