Performance Analysis of TCP Connections for Finite Data Transfer

Rossano Gaeta, Marco Gribaudo, Matteo Sereno
Dipartimento di Informatica, Universit di Torino,
Corso Svizzera, 185, 10149 Torino
{rossano,marcog,matteo}@di.unito.it

Abstract

This report describes our work in progress on defining an approach to investigate IP networks loaded with traffic resulting from a set of ON-OFF finite TCP-Reno connections. The approach is based on separate model descriptions of the TCP connection latency to transfer a finite number of packets and the IP network links; the two models are iteratively solved using a fixed point method. The overall model parameters are the primitive network characteristics; the model solution yields an estimation of the packet loss probability and the average time required to transfer a finite number of packets over a TCP connection. The validation of the proposed approach and the future extensions to this work will be discussed.

1 Introduction

The performance in the Internet network is dominated by a few crucial parameters that can be grouped into two main categories. In the first group we have the physical network parameters, the IP routing strategies, and the incoming traffic parameters. In the second category we can put all the aspects related with the used protocols. Although the role of the protocol parameters could be less evident, the Quality of Service (QoS) perceived by end users in their access to Internet services is often driven by TCP, the connection oriented transport protocol, whose congestion control algorithms dictate the
latency of information transfer. The literature on TCP performance is vast, with studies based on simulations, measurements, and analytical models. In recent years, several research groups aimed at modeling the TCP behavior with analytical paradigms, so as to obtain more general and parametric results and achieving a better understanding of the TCP behavior under different operating conditions. The development of accurate models of TCP is difficult, because of the intrinsic complexity of the protocol algorithms, and because of the complex interactions between TCP and the underlying IP network. This makes the development of accurate model for TCP an interesting problem for people involved in performance evaluation researches. In this report we address the problem of analyzing the average completion time and packet loss probability of a set of homogeneous TCP connections that behave as ON-OFF data sources; the activity begins when the connection starts send a finite amount of data. Each activity period begins after an exponentially distributed silence interval. We base this analysis on separate model descriptions for the latency of TCP connections and for the IP network that are iteratively solved using a fixed point method.

1.1 Related works
The literature on analytical models of TCP is quite vast, so that it is quite difficult to provide here a comprehensive summary of previous contributions. In this section we just mention some of the approaches that have been successfully used so far in this field and that are most related to our work. The TCP models that have appeared so far can be grouped in two classes:

- models that assume the round trip time and the loss characteristics of the IP network are known, and try to derive from them the throughput of TCP connections;
- models that assume only network parameters (such as network topology, number of users, data rates, propagation delays, buffer sizes, etc.) are known, and try to derive from them the throughput (and in some cases the delay) of TCP connections, the round trip time, and the loss characteristics of the IP network.

In both classes, models can either ideally assume a greedy behavior of TCP connections, this assumptions considers situations representing the transfer
of extremely large files, or more realistically consider the case of finite and short-lived TCP connections, as typical HTTP transactions.

The analysis of the behavior of TCP presented in [16] and [15] is based on measurements. In [16] the modeling technique is empirical, while in [15] the model is based on the analysis of TCP transmission cycles. The packet loss probability and the connection round trip time are needed as inputs to allow the model to compute the throughput and the average window size of the TCP connections. The first paper assumes that losses are not correlated, while the second paper takes into account the correlation in packet losses. Extensions to this latter paper that allow computing the latency of short file transfers are presented in [6] [17]. These four works can be considered representative of the first category defined above. Other works discussed here, can be included in the second category, though the modeling approaches may differ significantly.

Modeling techniques based on differential equations are used in [5], [13] [14]. In [5] a fluid model is used to compare the performance of TCP-Reno and TCP-Vegas. The authors of [13] and [14] used stochastic differential equations to model the behavior of TCP windows size (in [13]) and to combine it with the network behavior (in [14]). The model resulting from the last paper can cope with multiple interacting connections, defining a closed-loop system that can be studied with powerful control-theoretic approaches.

The use of Markovian modeling is exploited in [11]. This paper presents Markov reward models of several TCP versions on lossy links. Also the work in [7] [3] is based on a Markovian approach. The peculiarity in [7] [3] is the consideration of connections that exhibit a on-off behavior, following a Markov model. The approach used in [12] is based on the use of queueing network models for modeling the behavior of the window protocol. This approach has been extended in [8] [4] [9]. These proposals are based on the description of the protocol through a queueing network model that allows the estimation of the load offered to the IP network (that is also represented by means of a queueing network).

The description of the protocol behavior is decoupled from the description of the network behavior and the interaction between the two submodels is handled by iterating the solution of the submodels until the complete model solution converges according a fixed point algorithm.

The balance of this report is outlined as follows: Section 2 describes the modeling assumptions and the proposed iterative approach, Section 3 describes the TCP connection latency model, Section 4 presents the model
we developed to describe the IP network link, and Section 5 discusses our current experiments with the technique as well as some future development.

2 The modeling approach and assumptions

In this study we consider a set of $K$ homogeneous TCP Reno connections characterized by:

- an ON-OFF behavior where the OFF periods are exponentially distributed with average $T_{silence}$;
- a packet emission distribution $g = \{g_i\}$ where
  
  \[ g_i = Prob\{\# \text{ Packets to transfer} = i\}, \text{ and } i = N_{\min}, N_{\min} + 1, \ldots, N_{\max} + 1 \]

- a packet loss distribution $pl = \{pl_{ij}\}$, where
  
  \[ pl_{ij} = Prob\{j-th \text{ packet is lost} \mid \text{ given that } i \text{ packets must be transferred}\}. \]

  A strong assumption is that a TCP connection may experience at most one packet loss;

- Same routing in the network

- Similar round trip times (RTT) and packet loss probabilities

- Same maximum value for the congestion window ($W_{max}$)

The topology we consider is a simple bottleneck link connecting $K$ TCP senders to their receivers.

2.1 The modeling approach

Following an approach already employed for studying the behavior of TCP connections, the proposed approach is based on separate descriptions of the TCP connection latency to transfer a finite amount of packets and the IP network links. The TCP latency submodel describes the behavior of one of the set of $K$ connections sharing homogeneous characteristics.

The TCP connection latency submodel is derived from that presented in [17] where an analytical characterization of the average time required to transfer $N$ packets over a route experiencing a given RTT and packet loss probability $p$ is proposed. This model will be described in Section 3.
The IP network link submodel is based on a Generalized Stochastic Petri Net (GSPN) [1] [2] description of a finite buffer queue with batch arrivals modulated by an ON-OFF process. This model will be described in Section 4. A high-level description of the proposed modeling approach is depicted in Figure 1.

![Diagram of TCP connection and IP network link submodels with arrows indicating packet loss probability and round trip time]

Figure 1: High level description of the proposed approach

The TCP connection latency submodel receives as inputs from the IP network link submodel the following parameters:

- estimate of the packet loss probability;
- estimate of the RTT summing up the average queuing delay for accepted packets, packet transmission time, and constant propagation delays.

The TCP connection latency model provides an estimate of the average completion time, the average number of batches required to complete the packets transfer, the batch size distribution. The IP network link model receives as inputs:

- estimate of the average completion time for the modulating batch arrival process;
- average number of batches required to complete the packets transfer to be used to set the arrival rate of batches at the link queue;
- batch size distribution.

The estimates produced by the IP network link model are fed back to the TCP connection latency model in an iterative process until convergence is reached.
3 The TCP connection latency model

To characterize the latency of a TCP connection to transfer $i$ packets, given that the connection experiences a given RTT and packet loss probability $p(i)$, we used an analytical model derived from that proposed in [17]. We briefly describe the paper assumptions and main results; we refer the reader to the original paper for the full details.

3.1 The model description

The TCP connections are assumed to use the Reno congestion control algorithm; delays arising at the end points from factors such as buffer limitations are not considered for the latency computation. The sender sends full-sized segments as fast as its congestion window allows and the receiver advertises a consistent flow control window. The latency is modeled by considering the concept of ”rounds” as already introduced in [15] [6]. A round begins with the transmission of a window of packets and end when the sender receives an acknowledgement for one or more of these packets. Losses in one round are assumed to be independent of losses in other rounds; losses in one round are assumed to be independent. The time to transmit all packets in a round is assumed to be smaller than the duration of a round and that the duration of a round is independent of the window size. We modified the original model:

- not considering the delayed acknowledgement scheme for the receivers;
- neglecting the contribution of the connection establishment phase to the total connection latency;
- assuming that a connection may experience at most one packet loss.

To compute the total transfer time for a connection with $i$ packets to send and a packet loss probability $p(i)$, two terms have to be computed: the transfer time where no losses occur (denoted by $T_{no-loss}(i)$) and the average transfer time when one loss occurs (denoted by $E\{T_{1-loss}(i)\}$). The term $T_{no-loss}(i)$ is obtained by computing the number of rounds necessary to transfer $i$ packets: the congestion window increases exponentially in slow start until it reaches $W_{max}$ and then retains this value. The number of rounds is then multiplied by RTT to obtain the connection latency expressed in time units.
The computation of the term $E\{T_{1-loss}(i)\}$ is more complex and two cases are considered: if the lost packet (denoted as $j$) is among the first three packets then the loss will lead to a timeout. The window will reduce to 1, the slow start phase after the timeout lasts until the window will be equal to 2 and then the congestion avoidance phase will be entered. The number of rounds required to transmit $a$ packets in congestion avoidance starting with a window equal to $b$ can be easily computed and is denoted as $T_{linear}(a, b)$. The average duration of the timeout period is denoted as $E[TO]$. In this case:

$$T_{1-loss}(i, j) = (T_{no-loss}(i) + E[TO] + T_{linear}(i - k - 1, 2) - 1) * RTT,$$

where $k$ can be easily computed and represents the number of packets sent until the sender experiences the timeout. If the lost packet does not lead to a timeout then once the sender receives three duplicate acknowledgements it infers a loss and retransmits the lost packet entering the fast recovery phase before entering the congestion avoidance phase. There are two cases: if the lost packet $j$ is not among the last three packets of a window then

$$T_{1-loss}(i, j) = (T_{no-loss}(i) + 1) * RTT + T_{linear}(i - k, n) * RTT,$$

otherwise

$$T_{1-loss}(i, j) = (T_{no-loss}(i) + 1) * RTT + (T_{linear}(i - k, n) + 1) * RTT,$$

where $k$ and $n$ can be computed and represents the number of packets sent until the sender enters the congestion avoidance phase and window value when entering the congestion avoidance, respectively.

We can now define the average latency when a single packet loss occurs

$$E\{T_{1-loss}(i)\} = \sum_{j=1}^{i} p_{lj} * T_{1-loss}(i, j)$$

and the total transfer latency for the TCP connection

$$T_{transfer}(i) = (1 - p(i)) * T_{no-loss}(i) + p(i) * E\{T_{1-loss}(i)\}.$$

### 3.2 The model exploitation

The expression for $T_{transfer}(i)$ can also be used to compute the latency of the TCP transfer expressed in rounds unit by simply dividing $T_{no-loss}(i)$
and \( T_{1-\text{loss}}(i,j) \) by RTT. We denote this value as \( T_{\text{rounds}}(i) \). Furthermore, the analytical expressions for \( T_{\text{no-loss}}(i) \) and \( T_{1-\text{loss}}(i,j) \) are based on the computation of the window size at each round; therefore, we use the same expressions to define the value of \( W_{\max} \) functions \( \{w_h\} \), where

\[
w_h(i,j) = \text{the number of times the congestion window size is equal to } h \text{ during the transfer of } i \text{ packets assuming the } j\text{-th packet is lost.}
\]

We also define the value of \( w_h(i,j) \) in the case of \( j = 0 \) to represent the case when no packet is lost. The average value \( w_h(i,j) \) is defined as

\[
W_h(i) = (1 - p(i)) \cdot w_h(i,0) + p(i) \cdot \sum_{j=1}^{i} p_{ij} \cdot w_h(i,j),
\]

and the probability that the congestion window has assumed the value \( h \) during the transfer of \( i \) packets is defined as

\[
B_{h,i} = \text{Prob}\{\text{congestion window has assumed the value } h\} = \frac{W_h(i)}{\sum W_h(i)},
\]

where \( b \in \{1,2,\ldots,W_{\max}\} \). Starting from these definitions we compute the following results:

- \( T_{\text{transfer}} = E\{T_{\text{transfer}}(i)\} = \sum_{i=N_{\min}}^{N_{\max}} g_i \cdot T_{\text{transfer}}(i) \);

- \( T_{\text{rounds}} = E\{T_{\text{rounds}}(i)\} = \sum_{i=N_{\min}}^{N_{\max}} g_i \cdot T_{\text{rounds}}(i) \);

- \( B_h = \text{Prob}\{\text{congestion window has assumed the value } h\} = \sum_{i=N_{\min}}^{N_{\max}} g_i \cdot B_{i,h} \).

\( T_{\text{transfer}} \) will be used to compute the average activity duration of a TCP connection, \( T_{\text{rounds}} \) will be the inverse of the batch arrival process rate, and \( B_h \) values will be the batch size distribution to be fed to the IP network link model.
4 The IP network link model

The model used to represent an IP network link traversed by the \( K \) ON-OFF TCP connections is a Generalized Stochastic Petri Net (GSPN) representation of a finite capacity, single server queue with exponentially distributed service times whose average is equal to the packet transmission time. We refer the reader for a complete presentation of the GSPN formalism to \([1]\) \([2]\).

The arrival process is the superposition of \( K \) homogeneous two state Markov Modulated Poisson process with batch arrivals; the mean time spent in the OFF state is one of the network primitives: the OFF periods are exponentially distributed with average \( T_{\text{silence}} \) and transition \( T_{\text{on-off}} \) models the awakening of a traffic source. The mean time spent in the ON state is input from the TCP latency model and is given by \( T_{\text{trans}} \): transition \( T_{\text{on-off}} \) models the connection termination. The mean time between batch arrivals is given by \( T_{\text{round}} \): transition \texttt{batch}_\text{arrival} models the batch production. The batch size distribution is given by \( B_h \): the set of immediate transitions \texttt{batch}_h whose weights are equal to \( B_h \) models the probabilistic choice of a batch size.

Batch arrivals have been introduced in the IP network link model to consider the traffic burstiness but packets in the same window do not enter the link queue at the same time; instead, they are spaced in time by at least a transmission time. We devised an expedient to cope with this feature: we introduced a delay in the model using an exponential transition named \texttt{deliver}_\text{packets} whose firing delay is equal to \( T_{\text{transmit}} \). Transition \texttt{batch}_\text{arrival} is enabled to fire only when all packets in a batch have entered the queue. The queue is modeled by places \texttt{QUEUE} and \texttt{CAPACITY} that represent the queue space and the queue capacity, respectively. Immediate transitions \texttt{admit} and \texttt{loss} model the admission and the loss of a packet, respectively. Finally, transition \texttt{transmit}_\text{packet} models the packet transmission along the link. An example of the resulting GSPN model where only batches of size \{1, 2, 3, 4\} are modeled, is depicted in Figure 2.

The GSPN model is solved computing the steady state solution of the associated CTMC yielding two performance indexes: the packet loss probability defined as

\[
\frac{X(\text{loss})}{X(\text{loss}) + X(\text{admit})},
\]

9
and the average queuing delay for transmitted packets defined as
\[ \frac{\text{QUEUE}}{X(\text{admit})} \]

where QUEUE denotes the mean number of tokens in place QUEUE and \( X(\text{loss}) \) and \( X(\text{admit}) \) denote the throughput of transitions loss and admit, respectively. These two computed values are then fed back to the TCP connection latency model until convergence.

5 Discussion and future directions

The modeling approach has not been thoroughly validated against a wide range of system parameters. We are currently experimenting with a scenario composed of 40 Reno TCP connections with \( W_{\text{max}} = 21 \) that cross a single bottleneck link with capacity = 445 Mb/s, propagation delay = 10 ms, and queue capacity = 96. We are currently investigating the effectiveness of the modeling technique by considering several factors that may have an impact, among them:

- we are testing the technique against different packet emission probabilities \( g = \{g_i\} \);

- we are considering different packet loss probability distributions \( pl = \{pl_{ij}\} \); given a total packet loss probability equal to \( p \) that may be decomposed as \( p(i) = p \times i / \sum d \), where \( d, i \in \{N_{\text{min}}, N_{\text{min}} + 1, \ldots, N_{\text{max}}\} \) and \( g_i > 0 \), we are investigating ways of differently weight long connections with respect to short ones.
Furthermore, we are already defining extensions to the present work:

- remove the restrictive hypothesis of single packet loss;
- employ an alternative strategy for the IP network link model based on the Fluid Stochastic Petri Net (FSPN) [10] that would hopefully allow to consider much larger systems either with larger buffer size or with larger number of TCP connections;
- characterize the output process of the IP network link model to consider multibottleneck scenarios.

References


