Performance Modeling of TCP/IP in a Wide-Area Network

Eitan Altman, Jean Bolot, Philippe Nain, Mohammed Erramdani, Patrick Brown, Denis Collange, Driss Elouadghiri

INRIA

Performance Modeling of TCP/IP in a Wide-Area Network

Eitan Altman, Jean Bolot, Philippe Nain, Driss Elouadghiri, Mohammed Erramdani, Patrick Brown, Denis Collange.

N° 3142

Mars 1997

——— THÈME 1 ———

Rapport de recherche

Performance Modeling of TCP/IP in a Wide-Area Network[*]

Eitan Altman, Jean Bolot, Philippe Nain, [**] Driss Elouadghiri, [***] Mohammed Erramdani[****], Patrick Brown, Denis Collange. [*****]

Thème 1 — Réseaux et systèmes

Projet Mistral

Abstract:

We examine the problem of evaluating the performance of TCP connections over wide area networks. Our approach combines experimental and analytic methods, and proceeds in three steps. First, we have used measurements taken over Renater to provide a basis for the chosen analytic model. This model turns out to be a shared bottleneck model, in which a finite buffer queue is shared by two connections, one being the reference TCP connection, and the other representing the exogenous traffic (i.e. all the other connections). This model is unlike previous models which did not explicitly consider the impact of exogenous traffic on the reference TCP connection. Second, we use fluid modeling to analyze the behavior of the reference TCP connection. We identify two modes of operation. For each mode, we derive closed-form expressions for the average throughput and delay as a function of buffering, roundtrip delay, and characteristics of exogenous traffic. Third, we use simulation to validate the model, and we find in general good correlation with the analytic results.

Key-words: TCP/IP performance; Queueing theory; Fluid approximation; Simulation; Validation; Network measurements.

(Résumé : tsvp)

Unité de recherche INRIA Sophia Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex (France)
Téléphone : 04 93 65 77 77 – Télécopie : 04 93 65 77 65

Analyse des Performances de TCP/IP

Résumé :   Nous nous intéressons à l'évaluation des performances de connexions TCP/IP. Notre approche qui allie mesures et analyses se décompose en trois étapes. Dans un premier temps, nous avons effectué des mesures sur le réseau national Renater afin de valider le modèle mathématique choisi. Ce modèle est celui d'un goulot d'étranglement, représenté par une file d'attente à capacité finie, recevant deux types de trafics: une connexion TCP de référence et une connexion dite exogène représentant la contribution de tous les autres trafics. A la différence des modèles déjà proposés et analysés ce modèle considère l'impact du trafic exogène sur une connexion TCP de référence. Dans une deuxième étape nous développons un calcul approché des performances à l'aide de processus fluides qui nous permet de caractériser le comportement de la connexion TCP de référence. Nous identifions deux modes de fonctionnement et pour chacun d'entre eux nous obtenons des formules exactes pour le débit et le délai moyens considérés comme des fonctions du temps de transmission de bout en bout et des caractéristiques de la connexion exogène. Enfin, une campagne intensive de simulations nous permet de valider le modèle et les résultats obtenus.

Mots-clé :   Performances de TCP/IP; Théorie des files d'attente; Approximation fluide; Simulation; Validation; Mesures.

1 Introduction

The work in this paper is a response to concerns voiced by an operator about the performance of TCP (Transport Control Protocol [24]) over wide-area networks in general, and over the French research network Renater[1] in particular. Our specific goal was to obtain analytic expressions to analyze, and predict if possible, the time required to transfer large files over Renater. This analysis in turn would be used by the network operator for dimensioning purposes ("is the current bandwidth sufficient to provide reasonable end to end performance?") and for customer performance prediction purposes ("what kind of performance should a new customer expect given the current network configuration?").

Answering the questions above requires that one develop an understanding of TCP behavior as a function of network parameters, including both static parameters (such as buffer size or link bandwidth) and dynamic parameters (such as the dynamic aspects of the TCP control scheme or the characteristics of the traffic generated by other connections). While much is known about the behavior of TCP from observations of simulation and experimental results [23, 19, 21, 14], little is available in terms of closed-form expressions for the delay or throughput of a TCP connection in a wide area network. This is in large part because the complex dynamics of the TCP window flow control mechanism makes it difficult to analyze. Thus, much work so far has relied on experiments or simulation. Early analytical work examined simple models of a single connection in isolation over low and medium size/bandwidth networks [23], and was then extended to consider more complex but more accurate models [20], as well as connections with high bandwidth-delay products [15] and/or asymmetric characteristics [17]. In parallel, other work examined the impact of other connections on a reference TCP connection. This impact has been so far modeled as being that of an additional Bernoulli loss process [15].

Our work is closest in spirit and in approach to, and complements that of, reference [15]. We use a fluid modeling approach similar to that used there, however we explicitly model the impact of other connections on a reference TCP connection. Our general approach has proceeded in 3 steps. First, we have used measurements taken over Renater to provide a basis for the chosen analytic model. Our main result for the purpose of this paper is that the measurements can be interpreted using a shared bottleneck model. Not surprisingly, this result ties in well with that obtained in [4], which used measurements obtained over a variety of connections spanning the Internet.

Second, we have considered a shared bottleneck model of a TCP connection. In this model, a reference TCP connection shares a single bottleneck node with other connections while non bottleneck nodes are modeled by fixed delays. The traffic generated by these connections is assumed to be independent from the behavior of the reference TCP connection. We refer to this traffic as exogenous traffic, and to these connections as non controlled connections (even though some of them might be TCP connections, and hence might be flow controlled connections). We model these connections as a single exogenous connection that competes with the reference TCP connection for the resources of the bottleneck node. In prac-

---

[1]Renater is an acronym for REseau NAtional de télécommunications pour la Technologie, l'Enseignement, et la Recherche.

tice, we model the exogenous and the reference connection using a fluid model and use it to analyze the behavior of the reference TCP connection. Fluid models have been found to be helpful in providing insights into the dynamic as well as stationary behavior of a variety of feedback control mechanisms [3, 7, 9, 26]. This is important because the dynamic behavior of the TCP flow control mechanism has an important impact on the performance of TCP connections.

Third, we have used simulation to validate the fluid approximation and the analytic results obtained in step 2.

The rest of the paper is organized as follows. In Section 2, we briefly describe the TCP flow control mechanism and our fluid model of it. This model turns out to have two modes of operation. We analyze these modes in Sections 3 and 4, respectively. In Section 5, we describe and analyze the simulation results obtained to validate the analytic results. Section 6 concludes the paper.

2 Modeling TCP flow control

The TCP flow control scheme

Flow control mechanisms regulate the flow of packets into a network to prevent network resources from becoming congested and to make sure that these resources are shared fairly between different users. This regulation is typically done using feedback information about the state (more or less congested) of the network. In TCP, the state of the network is characterized by packet losses and the regulation is done using a dynamic window scheme [11].

The specifics of the control mechanism are as follows. Packets are assigned increasing sequence numbers. The source TCP sends in each packet continuous data octets accompanied by the sequence number of the first octet. The destination TCP maintains a set of continuous sequence numbers. When it receives a packet, the destination TCP sends an acknowledgment (ack) packet indicating the value of the receive window and the next expected octet sequence number. The receive window indicates how much buffer space is available for this connection at the destination host. Its size is fixed. Data octets below the receive window have been passed on to the application. Data octets received out of sequence but within the receive window are buffered.

The source TCP maintains another window called the congestion window, equal to the maximum allowed number of unacknowledged packets. The congestion window size is adjusted dynamically in response to ack reception and packet loss. A packet loss in turn is detected either with the receipt of 3 duplicate acks (i.e. consecutive acks that have the same "next expected" sequence number), or with the expiration of a timer. The source then is allowed to send min(receive window, congestion window) packets.

The congestion window size adjustments work in cycles made up of two phases, namely the *slow-start phase* and the *congestion-avoidance phase*. The window size is initially set to 1. In the slow-start phase,

the window size is increased by one every time an ack is received. Thus, when an ack arrives at the source, two packets are generated, one for the received ack and one because the window size is increased by one. This behavior causes an exponential increase of the window size as a function of time, and the amount of data in transit over the connection increases rapidly. The slow-start phase ends when the window reaches a certain level called the *slow-start threshold*.

At this point the congestion avoidance phase starts. The purpose of this phase is to slowly increase the load over the connection so as to adapt to and probe for the available bandwidth. This is done by increasing the current size W of the window by 1/[W] whenever a packet is acknowledged[2]. Thus, the window size increases by 1 when W packets have been received, i.e. roughly every roundtrip. When a packet is lost, the slow-start threshold is set to half the size of the window, the window is then set to 1, and a new cycle begins.

The control algorithm described above is referred to in the literature as "TCP Tahoe". Other versions such as "Reno" [12] and "Vegas" [6] have been proposed recently. They differ from Tahoe by slightly different window adjustments and packet loss detection schemes, and we will not consider them in this paper.

The shared bottleneck model

We next turn to the problem of modeling and analyzing the performance of a reference Tahoe TCP connection in a wide-area network. As indicated in Section 1, we consider a shared bottleneck model. In this model, the reference TCP connection shares a finite buffer FIFO bottleneck mode with other connections. Thus, the arrival stream at the bottleneck queue is the superposition of two streams, namely the reference TCP stream and the exogenous stream, which is in turn the superposition of many streams active during the TCP connection lifetime. We assume in this section that the exogenous stream is a constant rate stream independent of the state of the network, i.e. independent of the behavior of the TCP connection under study and of the load in the bottleneck queue (we examine in Section 5 how the performance measures are impacted when the exogenous stream is a Poisson stream). We use a constant delay τ to model the fixed component of the round trip delay of the reference TCP packets. Refer to Figure 2.

The parameters of the model are

- $W(t) \triangleq$ window size at time t. We take $W(0) = 1$.

- $W_{\mathrm{th}}(t) \triangleq$ value of the slow-start threshold at time t.

- $Q(t) \triangleq$ number of packets in the queue (from the TCP and exogenous stream) at time t.

---

[2][x] denotes the integer part of x.