A STATE-BASED MODEL OF TCP

TUAN A. TRINH, SÁNDOR MOLNÁR

Department of Telecommunications and Media Informatics
Phone: +36 1 463 3889, Fax: +36 1 463 3107, Email: {tuan,molnar}@tmit.bme.hu

Abstract

In this paper, a state-based model of TCP is presented. During a connection, TCP stays in either of the following states: Slow Start, Congestion Avoidance, Fast Retransmit (Recovery) and Time Out. We propose the use discrete-time batch Markov process (D-BMAP) to model the TCP. We use simulation to examine TCP state by state. We have developed a tool to investigate the behavior of TCP in different periods. At this stage, our tool can detect the beginning and the end of each period, collects some useful statistics for our state-based model. Finally, validation for Markov assumption is provided and verified.

I. Introduction

TCP modeling can be found in two main approaches: packet level and fluid level. One of the motivations for the packet level approach is the possibility of applying existing discrete-time models [1], [2], [5], [10]. Respectively, the motivation for fluid level model is the possibility of applying existing continuous-time models [3], [6], [7], [8], [9]. In both approaches, good points have been addressed and important, subtle results have been achieved. In [6], T. Ott *et al* used stochastic differential equations to model TCP behavior and first suggested the *square-root formula*. J. Padhye *et al* in [1] extends the model in [6] to capture Time Out. This model is widely accepted as one of the most accurate models for TCP Reno (in the case of bulk data transfer). We can also mention here the chaotic nature of TCP as suggested and examined in [3]. However, as TCP modeling is application-sensitive, a general purposed TCP model that is precise, yet simple, is still unavailable. This makes the TCP modeling task still very challenging. Another issue of TCP modeling is the types of modeling: black-box modeling and white-box modeling.

Black-box modeling approaches usually start from a theoretical model while white-box modeling approaches try to mimics TCP inherent operations based on some statistics. An example of white-box modeling is the well-known ON/OFF model for voice traffic. It has two states: IDDLE and SPEAK. If the speaker speaks, then it is in SPEAK state, and it is in IDDLE state otherwise. A natural question arises then: How about TCP? Our state-based model of TCP follows white-box modeling approach. We model TCP by its states. During a connection, TCP stays in either of the following states: Slow-Start, Congestion Avoidance, Fast Recovery, Exponential Back-off. TCP can jump from one state to another state in response to external events such as packet loss or Time Out. We consider how much time TCP stays in each state and the distribution of time elapsed at each state. We then consider the jumping probability from one state to another state. From the statistics, we can build a model to estimate TCP throughput. We have developed our technique in a tool called **TCP_ASD** (**TCP A**utomatic **S**tate **D**etection) that automates state analysis of TCP connections. This paper reports some results achieved from the tool and simulations of TCP.

The remainder of the paper is organized as follows. In Session 2 we describe the simulation setup that we use throughout this paper. In Session 3 we describe our state-based model of TCP in more detail. Session 4 describes the TCP_ASD tool. Session 5 presents some results from our analysis. Session 6 concludes the paper.

II. Our proposed model of TCP

A. The Modulating Markov Chain

We consider the dynamics of TCP state by state. After the hand-shake period, TCP starts in Slow-Start. In this period, the congestion window is increased in an exponential manner. From Slow-Start, TCP can jump into either of the following states: Fast Recovery, Congestion Avoidance and Exponential Back-off, depending on the external events. If the congestion window gets the slow start threshold without packet loss, then TCP will enter Congestion Avoidance. We call the probability that TCP jumps from Slow-Start to Congestion Avoidance by p_{sc} If packet loss happens then on the arrival of the third duplicate ACK, TCP will retransmit the lost packet and enter Fast Recovery. We call the probability that TCP jumps from Slow-Start to Fast Recovery by p_{sf} . Otherwise it enters Time Out. We call the probability that TCP jumps from Slow-Start to Exponential Back-off (or Time Out) by p_s . Let's suppose now that TCP is in Fast Recovery.

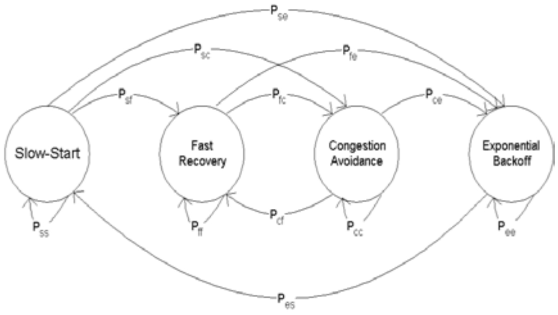

Figure 1: The modulating Markov chain

Similarly, from Fast Recovery, TCP jumps to Congestion Avoidance with probability p_{fc} and to Exponential Back-off with probability p_{fe} . From Congestion Avoidance, TCP jumps to Exponential Back-off with probability p_{ce} and to Fast Recovery with probability p_{cf} . From Exponential Back-off, the only possibility for TCP is to jump to Slow-Start (with p_{es}). We also consider the distribution of time elapsed in each state. If the time spent at each state is geometrically distributed, then we have a discrete-time Markov chain. The states of the Markov chain are the states of TCP itself.

B. A D-BMAP model for TCP

We propose a discrete-time model for TCP. The states of the background process (modulating process) is the states of TCP itself (i.e. Slow Start, Congestion Avoidance, Fast Recovery and Time Out)

First, let's consider a general model of discrete MAP

- The process is time-slotted: the slot length is a round-trip time (RTT)
- The probability of a transition from state *i* to state *j* for is denoted by p_{ij} and the transition probability of the modulating Markov-chain is $\mathbf{P} = \{p_{ij}\}$
- When the chain is in state *l*, the source transmits a random number of packets with probability generating function (p.g.f) $\mathbf{B}_l(z) = \sum_i b_i^{(l)} z^i$, where $b_i^{(l)}$ denotes the probability of *i* arrivals in a slot when the Markov chain is in state *l*.

Let us define **B**(z) matrix as follow:

$$\mathbf{B}(z) = \begin{bmatrix} p_{00}B_0(z) & p_{10}B_0(z) & \dots & p_{N0}B_0(z) \\ p_{01}B_1(z) & p_{11}B_1(z) & \dots & p_{N1}B_1(z) \\ . & . & . & . \\ p_{0N}B_N(z) & p_{1N}B_N(z) & \dots & p_{NN}B_N(z) \end{bmatrix}$$

Let Π denotes the stationary distribution of the modulating Markov chain. Then we can estimate the long term average throughput of a TCP connection as follow:

$$\overline{BW} = \Pi\left(B'(1)\right)^T \bar{e} \text{ [packet/RTT]}$$

where $\bar{e}=[1,1,\dots,1]^T$.

III. A tool for validation

In order to validate the model, we need to build a tool to collect the statistics that we use for our validation process. Our tool has two major parts. The first part is responsible for running simulation and preprocessing the data. This part also collects useful information from the simulations of TCP connections such as the congestion window and slow start threshold (*cwnd_* and *ssthresh_*). The second part is responsible for actually producing the results. We have implemented (in C++) a number of our algorithms [4] to illustrate a number of TCP metrics such as number of out-going packets, number of forward-going packets. The TCP_ASD tool uses the dynamics of the congestion window to detect state changes of a TCP connection, collects some statistics such as sojourn time at states, bytes sent at states for our state-based model. We also examine different TCP versions such as TCP Tahoe, Reno, New-Reno, and SACK. Finally the MATLAB scripts part is responsible for illustrating the dynamics of the mentioned processes.

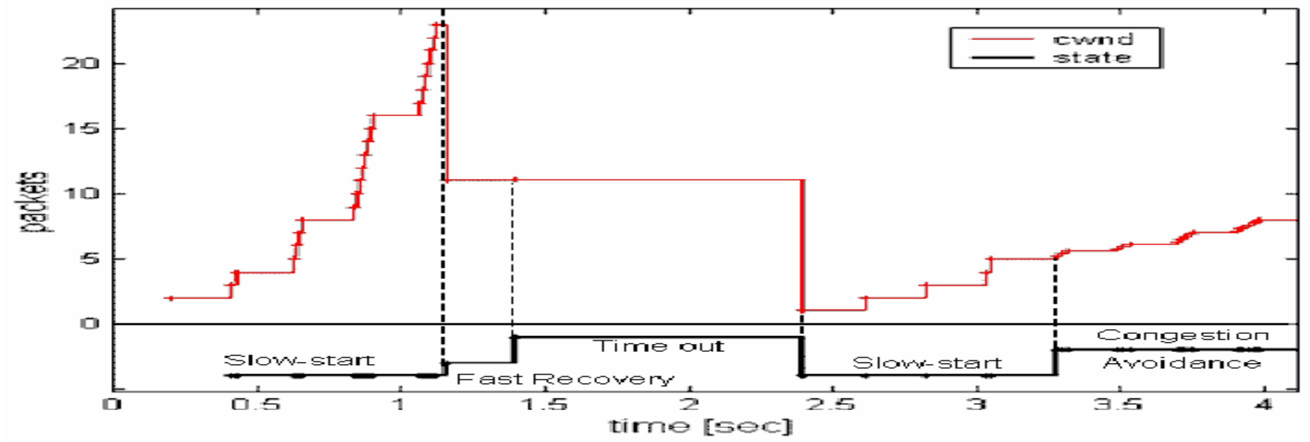

Figure 2: State detection

Figure 2 illustrates how state change is detected. The connection starts with a Slow Start. After the third duplicate ACK arrives, the congestion window is halved and TCP enters Fast Recovery. The receipt of the recovery ACK gets TCP out of Fast Recovery. At this point there are two possibilities: Congestion Avoidance or Time Out. If only one packet is lost in the window of packets, then

Congestion Avoidance follows. If multiple losses occur, then Time Out (or Exponential Back-off) follows. When the timer expires, the congestion window is set to 1 and Slow Start begins and the congestion window is increased in an exponential manner. If the congestion window gets the slow start threshold, TCP enters Congestion Avoidance.

IV. Markov chain validation

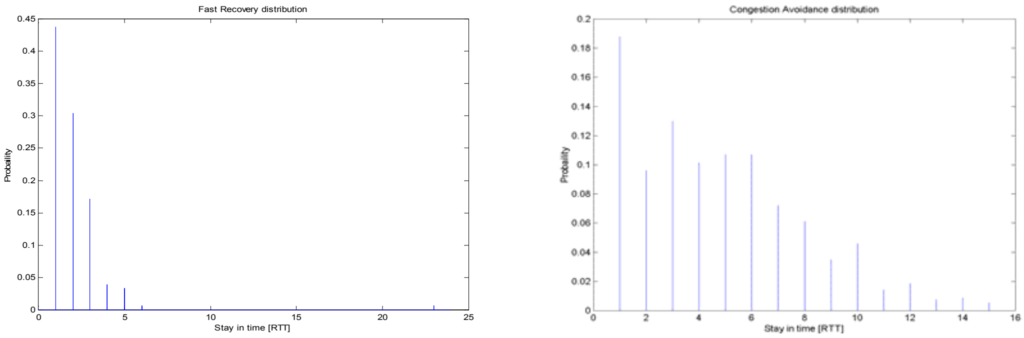

Figure 3 Sojourn time distributions

As we can see in Figure 3, the sojourn time distribution at Congestion Avoidance state (for all versions of TCP) and at Fast Recovery state (for New Reno) can be approximated by a geometric distribution. This is certainly the truncated version of the distribution. The geometrically distributed sojourn time implies Markov property for discrete time.

V. Conclusion

We have presented a state-based model of TCP through. We have developed a tool for our analysis. We used the tool to collects useful statistics for our state-based model of TCP. Analysis and validation of our state-based model is left as our future work.

References

[1] Jitendra Padhye *et al*, Modeling TCP Reno Throughput: A Simple Model and its Empirical Validation, *SIGCOMM'98*, 1998.

[2] A. Kumar, " Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link", *IEEE/ACM Transactions on Networking*, 1998.

[3] A. Veres, M. Boda, The Chaotic Nature of TCP Congestion Control, INFOCOMM 2000, Tel Aviv, Israel, 2000.

[4] T. A. Trinh, T. Éltető, On Some Metrics of TCP, *LCN 2000*, Tampa, Florida, USA, November 2000.

[5] N. Cardwell, S. Savage, T. Anderson, Modeling TCP Latency, *INFOCOMM 2000*.

[6] T. Ott, J.H.B. Kemperman, M. Mathis, The Stationary Behavior of Ideal TCP Congestion Avoidance, Bell Lab Technical Report, 1996.

[7] V. Misra, W. Gong and D. Towsley. A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, *SIGCOMM 2000.*

[8] Schwefel, Behavior of TCP-like Elastic Traffic at a Buffered Bottleneck Router, *INFOCOMM 2001.*

[9] Steven Low *et al,* Dynamics of TCP/RED and a Scalable Control, *INFOCOMM 2002.*

[10] Peerapol Tinnakornsrisuphap, Armand Makowski, Limit Behavior of ECN/RED Gateways Under a Large Number of TCP Flows, *INFOCOMM 2003*.