STOCHASTIC MARKOV MODEL FOR TCP THROUGHPUT

Asmir Gogić [1], Nermin Suljanović [1], Amer Hasanović [1], Aljo Mujčić [1], Matej Zajc [2]

[1]Faculty of Electrical Engineering, University of Tuzla
Franjevačka 2, Tuzla 75000, Bosnia and Herzegovina
[2]University of Ljubljana, Faculty of Electrical Engineering
1000 Ljubljana, Tržaška 25, Slovenia
asmir.gogic@untz.ba(Asmir Gogić)

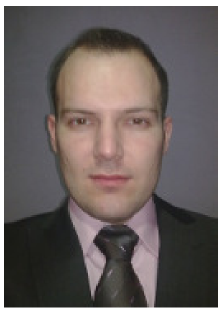
Abstract

The paper presents a stochastic model that predicts the throughput of the short and long lived Transmission Control Protocol (TCP) flows. Analytical model for each TCP stage are derived and combined with discrete time Markov chain, with round trip time granularity, in order to describe TCP throughput behavior during the TCP connection life time. Model accuracy is improved by the detailed analysis whether the packets are lost or received after the first packet loss.
While deriving the model described in this paper, assumption that the packet loss function is known was made. The model accuracy has been tested on the simple network topology created in the packet level simulator ns-2. The proposed model can be used for TCP connection performance analysis, reducing computation complexity compared to the packet-level simulators.

Keywords: TCP modeling, Stochastic models, Markov chain.

Presenting Author's Biography

Asmir Gogić is teaching assistant on the Faculty of Electrical Engineering at the University of Tuzla, where he received his B.Sc degree in 2007 and M.Sc. degree in 2009. His research interests are in modeling and simulation of communication networks and protocols.



1 Introduction

TCP protocol is the main transport protocol used by Internet applications [1]. Measurement of the TCP connection performance is of great importance for estimation of a network performance. TCP traffic in a given network is usually analyzed using three different methodologies: measurements, simulation and mathematical models. Measurement is usually impractical and expensive due to growth and complexity of the Internet [2]. On the other side, packet simulations are effective tools for studying network and protocols behavior [3] since they employ full TCP stack implementation. Furthermore, computer efficient analytical models can be used for improvement of existing and evaluation of new congestion control algorithms [4] and queuing mechanisms [5], link design [6] and planning IP (Internet Protocol) networks.

There are two approaches for deriving of analytical models. One way is to use differential equations do describe relevant parameter [7, 8]. Second approach, is based on the probability analysis [9, 10, 11, 12]. In this approach, statistical formulas are used to describe TCP behavior in different stages. By aggregating models for all TCP stages, the full TCP behavior have can be obtained.

In this paper, a stochastic Markov model for throughput of the TCP Reno is derived. The model is validated by comparing the results with the packet-level simulation tool ns-2 [13].

2 TCP basic principals

TCP is a reliable connection-oriented transport protocol for packet switched networks. Reliability is achieved by employing acknowledgments (ACKs) [14]. Using ACKs and sequence numbers, a transmitter tracks packets that are successfully delivered to a receiver. TCP operates as state machine and for the TCP Reno, the states are: slow start, congestion avoidance, fast retransmit, fast recovery and timeout. Transition between states is determined by a packet loss or acknowledgment of predefined number of packets. During each stage, window size w_i is incremented or decremented by a different rule. The window (buffer) represents maximum number of packets that can be sent without receiving any ACKs. Time interval, from departure of the first packet to the departure of the last packet in the window, represents a round. In the slow start state, window size is incremented with each received ACK, but in congestion avoidance state, window size is incremented by the small amount of the window size $1/w_i$. The window size varies with the rate of the packet loss in the network. Hence, the packet loss probability increases with a number of sent packets due to the congestion in the network. When a packet loss occurs during the slow start or congestion avoidance stage, there are two mechanism to detect it. First mechanism detects packet loss by using timeouts (TO). The second mechanism detects packet loss upon receiving the three duplicate ACKs.

3 Modeling TCP

There are several approaches for modeling of packet switched networks. The most accurate network models are packet level models, where behavior of each packet is modeled individually. These models are implemented in network simulators, such as ns-2 [13]. They can be used for validation of analytical models. The main drawback of packet level simulators is a large computational effort required for large scale simulations. Analytical models are based on the concept to model the TCP behavior mathematically in order to reduce simulation complexity [9, 11, 12, 15]. These models do not describe TCP dynamics. They assume that round-trip time and loss probability are constant and there are no interactions between TCP flows. On the other hand, fluid models overcome network scalability problem by keeping track of average quantities for relevant network parameters [16, 17]. Additionally, they use the assumption that the bit rates are piecewise constant. Hybrid models incorporate continuous time states but also discrete-time logic [18, 19]. They introduce reduction of the computing complexity by continuous approximation of variables such as queue and window size. Hybrid simulations require significantly less computational resources than packet level simulators. However, solution of hybrid equations is still necessary in order to simulate networks. One group of analytical models are stochastic models [9, 10, 20] which introduce reduction of computation resources and simulation time by employing stochastic analysis. Main drawback of these models is that they can't describe dynamic behavior of TCP.

4 Stochastic Markov model

The TCP flow throughput is essential for the TCP performance analysis. The throughput of the TCP flow is obtained as average of the throughput in stages of the TCP state machine [9]. In the TCP state machine, transition from one state to another is characterized by appropriate probability (Fig. 1). Therefore, Markov chain can be applied for the TCP behavior modeling [20]. This approach requires mathematical formulas for the throughput and window size in each stage and assessment of transition probabilities. Our analysis is based on the following assumptions:

- Packet loss probability function is known.
- Packet losses occur only on the path from server to client.
- Packet losses in two successive rounds are not correlated.
- Packet propagation time from client to server and back is constant.

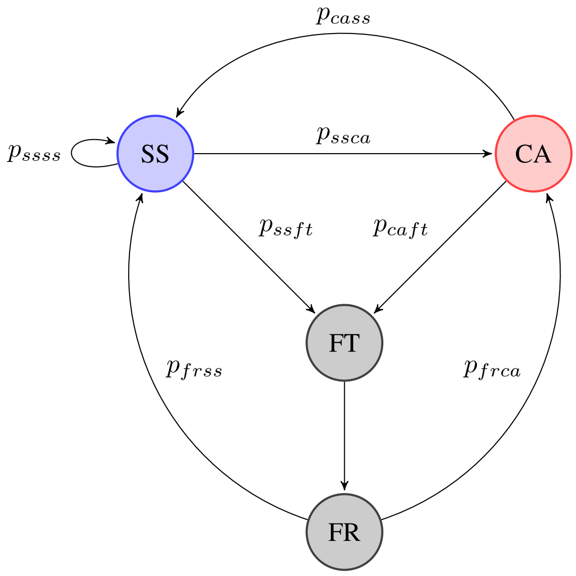


Fig. 1 TCP state machine

Transition between state x to state y (Fig 1) occurs with probability p_{xy}. After the three way handshake procedure is preformed, a TCP connections is established and TCP model is initialized in the slow start state. From slow start state (Fig 1. SS) TCP can transition to congestion avoidance (CA) state, fast retransmit (FT) state or remain in slow start by going through the timeout state. Transition from SS to CA state is determined by conditional probability p_{ssca}. In the slow start state, window size growth is limited by the value *ssthresh*=65536 bits. If the packet loss probability is p, then probability for the window size reaching the $ssthresh$ limit equals conditional probability p_{ssca}

$$p_{ssca} = p(CA|SS) = (1-p)^M \tag{1}$$

where $M = 2 \cdot ssthresh/packet\_size - 1$ in packets. TCP can return SS to SS state (Fig 1) if packet loss occurs. This transition is characterized by probability p_{ssss} which is equal to probability of the packet loss detection with TO mechanism. The value of p_{ssss} probability represents the sum of probabilities of cases whether there was less then three successfully received packets after the first packet loss in the current state. The simplified equation for evaluation of p_{ssss} is:

$$p_{ssss} = p(SS|SS) = \pi_{SS0} + \pi_{SS1} + \pi_{SS2} \tag{2}$$

where $\pi_{SS0}, \pi_{SS1}, \pi_{SS2}$ are probabilities to have 0, 1 or 2 successfully sent packets after the first packet is lost.

$$\pi_{SS0} = \frac{1}{1-(1-p)^{w_i}} \sum_{k=0}^{w_i-1} (1-p)^k p^{w_i-k} \cdot \rho_{SS0} \tag{3}$$

where the ρ_{SS0} is:

$$\rho_{SS0} = p^{2k} + 2kp^{2k-1}(1-p) + p^{2k-2}(1-p)^2(2k-1)k$$

$$\pi_{SS1} = \frac{1}{1-(1-p)^{w_i}} \sum_{k=0}^{w_i-2} (1-p)^{k+1} p^{w_i-k-1} \rho_{SS1} \tag{4}$$

where the ρ_{SS1} is:

$$\rho_{SS1} = (w_i - k - 1)\left(p^{2k} + 2k(1-p)p^{2k-1}\right)$$

$$\pi_{SS2} = \frac{1}{1-(1-p)^{w_i}} \sum_{k=0}^{w_i-3} (1-p)^{k+2} p^{w_i-k-2} \rho_{SS2} \tag{5}$$

where the ρ_{SS2} is:

$$\rho_{SS2} = \frac{1}{2}(w_i - k - 1)(w_i - k - 2)p^{2k}$$

Transition from SS to FT is characterized by p_{ssft} which is equal to probability of packet loss detection with TD mechanism:

$$p_{ssft} = p(FT|SS) = 1 - p_{ssca} - p_{ssss} \tag{6}$$

Throughput for slow start state is:

$$R_{ss} = \frac{E[ssd_{ack}]}{P_s E[ssd] t_{ls} + 2(E[i]+1)t_{ld} + (1-p_{ssca})t_{ADT}} \tag{7}$$

where $E[ssd_{ack}]$ is expected value for the number of ACKed packets in slow start state, P_s packet size in bits, t_{ls} link speed, $E[ssd]$ expected value for the total amount of transmitted packets in slow start, $E[i]$ average value of round in which packet drop occurred, t_{ld} packet propagation time from transmitter to receiver, p probability of packet loss and t_{ADT} average time to detect packet loss. The average time to detect packet loss is:

$$t_{ADT} = p_{ssss} E[t_{TO}] + p_{ssca} E[t_{TD}] \tag{8}$$

where t_{TD} represents the time interval from emitting the last packet in the window until the detection of packet loss using the TD mechanism and t_{TO} represents the time interval from emitting the last packet in the window until the detection of the packet loss using the TO mechanism.
The number of ACKed packets in the slow start state can be determined by analyzing two boundary cases related to the position of the lost packet in the window. The first case corresponds to packet loss at the beginning of the round (Fig. 2). The packet loss at the end of the round (Fig. 3) is the second case.

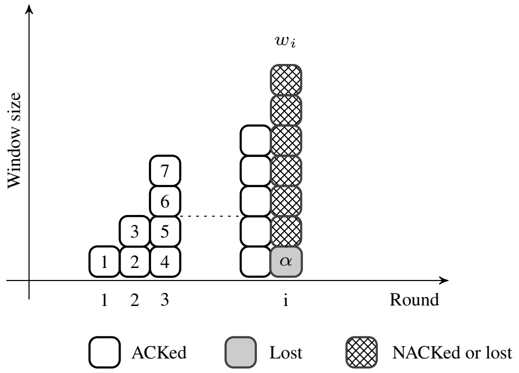


Fig. 2 Slow start state, packet drop occurred at beginning of the round *i*

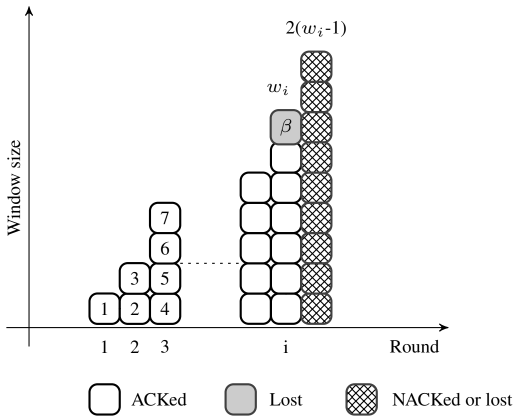


Fig. 3 Slow start state, packet drop occurred at end of the round *i*

Taking into account the window growth limit *ssthresh*, the expected value for number of ACKed packets in the slow start state is:

$$E[ssd_{ack}] = \frac{1-(1+Mp)(1-p)^M - p}{p} \tag{9}$$

Expected value of the window size at the end of the slow start state is:

$$E[W_{SS}] = \frac{3(\gamma-1)(2A+1) - 2A - 3 + 6w_0 - \gamma}{\gamma+1} \tag{10}$$

where $A = E[ssd_{ack}]$ from equation (9), γ coefficient of exponential growth of window size and w_0 initial window size. Expected value of the total number of transmitted packets in slow start is:

$$E[ssd_{ack}] = E[ssd_{ack}] + E[W_{SS}] \tag{11}$$

Calculation of the throughput for the slow start state in equation (7) requires information of average of rounds in which packet loss occurred. This value can be derived by averaging the round number with respect to the slow start exponential window growth in the following manner:

$$E[i] = \lim_{N\to\infty} \sum_{i=1}^{N} i(1-p)^{w_0 \frac{\gamma^{i-1}-1}{\gamma-1}} p \sum_{k=0}^{w_0\gamma^{i-1}-1} (1-p)^k \tag{12}$$

Value of $E[i]$ can be computed for finite value N, in such manner that the cumulative distribution function (CDF) is higher then 0.995

$$E[i] \approx \sum_{i=1}^{N} (1-p)^{w_0 \frac{\gamma^{i-1}-1}{\gamma-1}} \tag{13}$$

Throughput computation in the congestion avoidance state is similar to the above approach. Throughput for the congestion avoidance state can be expressed as:

$$R_{ca} = \frac{E[cad_{ack}]}{P_s E[cad] t_{ls} + 2(E[i]+1)t_{ld} + t_{ADT}} \tag{14}$$

where $E[cad_{ack}]$ is expected value of the number of ACKed packets and $E[cad]$ is expected value of the total amount of transmitted packets in congestion avoidance state. Now, using the same analysis depicted in (Fig. 2, Fig. 3) and rule for the window growth in congestion avoidance state, we derive equation for expected value for number of ACKed packets for congestion avoidance state:

$$E[cad_{ack}] = \frac{1-p}{p} \tag{15}$$

The expected value of the window size at the end of the congestion avoidance state is:

$$E[W_{CA}] = \sqrt{2\frac{1-p}{p} + w_{0CA}^2} - w_{0CA} \tag{16}$$

where w_{0CA} is initial window size in the congestion avoidance state

$$w_{0CA} = (1-p)^M E[W_p] + (1-(1-p)^M)\frac{E[W_p]}{2} \tag{17}$$

W_p is the window size at the end of previous state. The probability of the transition between congestion avoidance and slow start is equal to the probability of the packet loss detection with TO mechanism:

$$p_{cass} = p(SS|CA) = \pi_{CA0} + \pi_{CA1} + \pi_{CA2} \tag{18}$$

where $\pi_{CA0}, \pi_{CA1}, \pi_{CA2}$ are probabilities to have 0, 1 or 2 successfully sent packets after the first packet is lost.

$$\pi_{CA0} = p^{w_i - 1} \tag{19}$$

$$\pi_{CA1} = (1-p)p^{w_i-2}(w_i-1) \tag{20}$$

$$\pi_{CA2} = \frac{1}{2}(1-p)^2 p^{w_i-3}(w_i-1)(w_i-2) \tag{21}$$

During the fast retransmit state, the first lost packet in the previous state is retransmitted with the probability $1-p$ and unconditional transition to fast recovery occurs.
In the fast recovery stage, TCP will remain for the duration of RTT×L time interval, where L is number of the lost packets in the previous state (slow start or congestion avoidance). This value can be determined by averaging the number of lost packets in the window (Fig. 4).

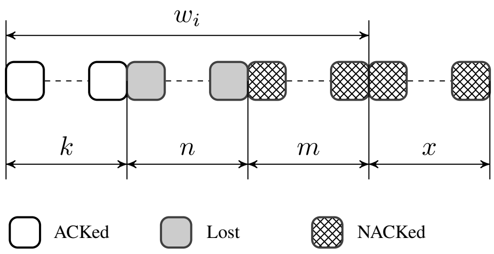


Fig. 4 Packet loss evaluation; $x = 2k$ for slow start and $x = k$ for congestion avoidance

For the slow start state, average number of the lost packets is:

$$N_{l_{SS}} = \sum_{n=1}^{w_i} \sum_{k=0}^{w_i-n} np^n(1-p)^{w_i+2k-n} \binom{w_i+k-1}{n-1} \tag{22}$$

and for congestion avoidance state:

$$N_{l_{CA}} = \sum_{n=1}^{w_i} \sum_{k=0}^{w_i-n} np^n(1-p)^{w_i-n} \binom{w_i-k-1}{n-1} \tag{23}$$

Probability of transition to the congestion avoidance is equal to the probability of having all lost packets successfully recovered:

$$p_{frca} = p(CA|FR) = (1-p)^{N_{lost}} \tag{24}$$

where $N_{lost} = N_{l_{SS}}$ when previous state was slow start and $N_{lost} = N_{l_{CA}}$ when previous state was congestion avoidance. If TCP do not recover lost packets in fast recovery state it transition to slow start. Probability for such outcome is:

$$p_{frss} = p(SS|FR) = 1 - p_{frca} \tag{25}$$

The model for the timeout mechanism has been adopted from [9], and its integrated in to slow start model for overall model simplicity.

5 Simulation results

Derived model is validated by the simulations in the packet level simulator ns-2. Since the TCP protocol is primarily built for point-to-point communications [14], this allows us to model entire network behavior and complexity (packet drop mechanism, packet propagation time, network topology) with the single probability of the packet loss p. Furthermore, test network contains only two nodes, interconnected with the full duplex link.
Packet loss distribution function is chosen to be uniform just for simulation simplicity. Model equations can be easily redefined for any certain packet loss distribution function.

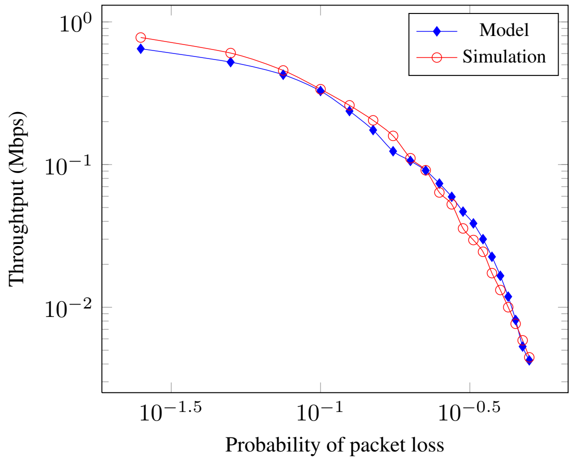


Fig. 5 Throughput for scenario link delay 12ms and link speed 1Mbps

For each scenario depicted in Fig. 5 through Fig. 10, measurements of the window size and throughput in all states were obtained through a large number of simulations so we could have more realistic picture for average value of the measured parameters. Results obtained in tests were averaged so they could be compared to the values obtained from the derived model.

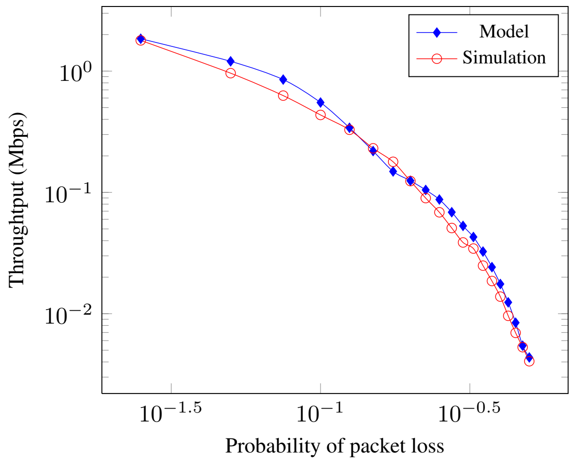


Fig. 6 Throughput for scenario link delay 10ms and link speed 10Mbps

Model accuracy has been tested on different sets of the network configurations. The model results for scenarion link delay 12ms and link speed 1Mbps is depicted in Fig. 5 and for scenarion link delay 12ms and link

speed 20Mbps is depicted in Fig. 10. We observe from this figures that model accuracy is slightly sensitive on the links speed variations. Packet departure time becomes irelevant to throughput in cases of higher links speeds. This is depicted in Fig. 6 and Fig. 7. Model time granularity limits model accuracy up to one round trip time. In other words, model accuracy improves in cases of higher round trip time as it can be seen in Fig. 8 and Fig. 9.
Degradation in model arises from the fact that we didn't take in to account departure time of ACK's and how many there are of them. Since the model design is based on the Markov chain, throughput computation of the TCP flow requires acknowledgments of all packets. However, throughput calculation in each state is instant and thus our model reduces simulation time compared to real TCP implementation.

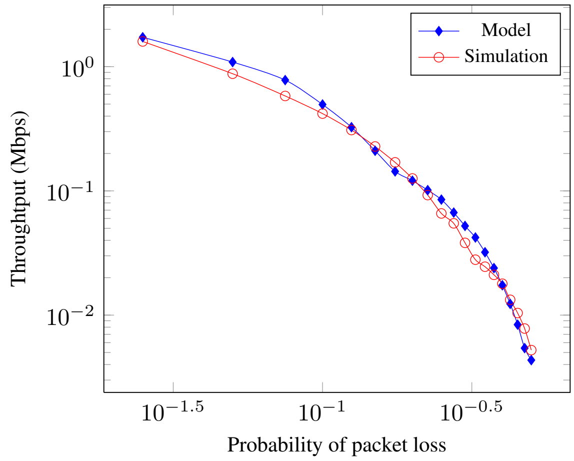


Fig. 8 Throughput for scenario link delay 12ms and link speed 20Mbps

6 Conclusion

In this paper we presented an stochastic Markov model for the TCP Reno protocol. Improvements in accuracy is achieved by detailed analysis of packets outcome after the packet loss. In other words we didn't assume that all packets after first packet loss are lost, which lead to introduction of probabilities given by equations (2) and (18). Second contributor to the accuracy comes due to integration of models for all states of the TCP Reno in to the stochastic Markov model.
The proposed model was validated by comparing the results from the model with the averaged values obtained from the packet-level simulator ns-2. Our analysis was confirmed by a large number of simulations. We observe that the model gives good description of throughput for the TCP Reno. The derived model is applicable only in the cases when a packet loss distribution function is known.

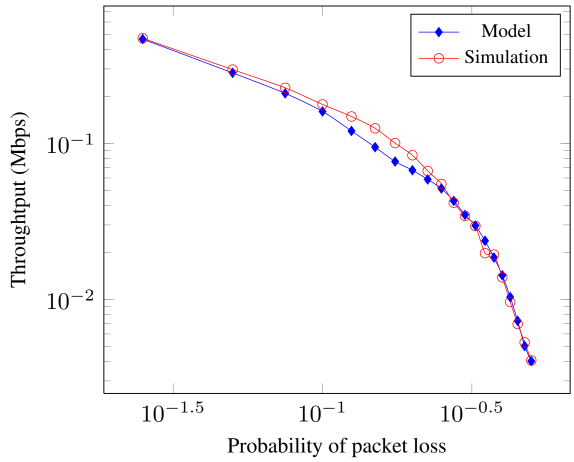


Fig. 9 Throughput for scenario link delay 50ms and link speed 20Mbps

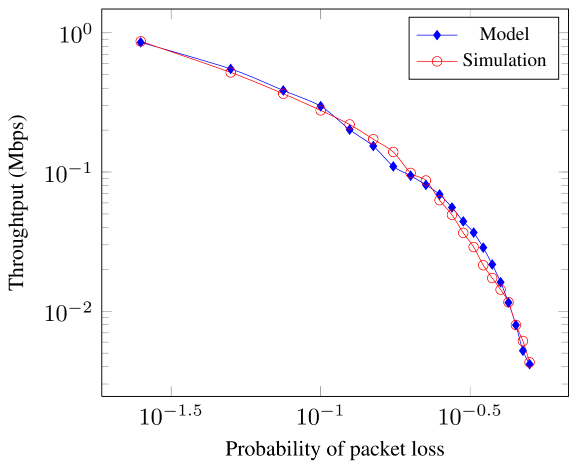


Fig. 7 Throughput for scenario link delay 25ms and link speed 10Mbps

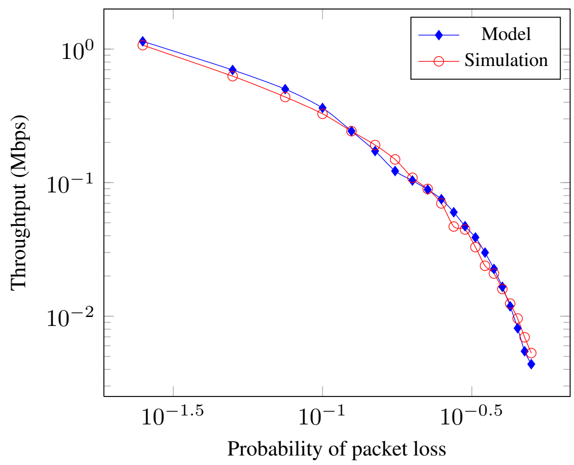


Fig. 10 Throughput for scenario link delay 20ms and link speed 100Mbps

7 References

[1] K. Thompson, G. Miller, and R. Wilder. Wide area internet traffic patterns and characteristics. 11:10–23, 1997.

[2] C.Williamson. Internet traffic measurement. 5:70–74, 2001.

[3] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. *Computer Communication Review*, 26:5–21, 1996.

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proc. ACM SIGCOMM'00*, pages 43–56, 2000.

[5] M. May, T. Bonald, and J.C. Bolot. Analytic evaluation of RED performance. In *Proc. IEEE INFOCOM*, pages 1415–1424, 2000.

[6] L. Bodrog, G. Horvth, and C. Vulkn. Analytical tcp throughput model for HSDPA. In *24th UK Performance Engineering Workshop UKPEW 2008*, pages 89–106, July 2008.

[7] A. Budhiraja, F. Hernndez-Campos, V. G. Kulkarni, and F. D. Smith. Stochastic differential equation for TCP window size: Analysis and experimental validation. *Probability in the Engineering and Informational Sciences*, 18:111–140, 2004.

[8] M.A. Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello. Using partial differential equations to model TCP mice and elephants in large IP networks. *IEEE/ACM Transactions on Networking*, 13:1289–1301, 2005.

[9] J. Padhye, D. Towsley, and J. Kurose. Modelling TCP throughput: A simple model and its empirical validation. In *Proc. ACM SIGCOMM*, pages 303–314, 1998.

[10] E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. 13:356–369, 2005.

[11] N. Cardwell, S. Savage, and T. Anderson. Modelling TCP latency. In *Proc. IEEE INFOCOM*, pages 1742–1751, 2000.

[12] B. Sikdar, S. Kalyanaraman, and K. S. Vastola. Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK. 11:959–971, 2003.

[13] VINT. *The VINT Project, a collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC*, 2000.

[14] J. Postel. Transmission control protocol. RFC 793, September 1981.

[15] K. Zhou, K. L. Yeung, and Victor O.K. Li. Throughput modeling of TCP with slow-start and fast recovery. In *Proc. GLOBECOM '05*, pages 261–265, 2005.

[16] Y. Liu, L.F. Presti, V. Misra, and D. Towsley. Fluid models and solutions for large-scale ip networks. In *ACM SIGMETRICS*, pages 91–101, 2003.

[17] V. Misra, W. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP ows with an application to RED. In *Proc. ACM SIGCOMM*, pages 151–160, 200.

[18] J. Lee, S. Bohacek, J.P. Hespanha, and K. Obraczka. Modeling communication networks with hybrid systems. 15:630–643, 2007.

[19] A. Kavimandan, W. Thottan Lee, A. M. Gokhale, and R. Viswanathan. Network simulation via hybrid system modeling: a time-stepped approach. In *Proc. ICCCN '05*, pages 531–536, 2005.

[20] J. Padhye, V. Firoiu, and D. Towsley. Stochastic Model of TCP Reno Congestion Avoidence and Control. Technical Report UM-CS-1999-002, University of Massachusetts, MA, USA, January 1999.