A Steady State Throughput Model of TCP SACK

Zhiming Wang Xiaoping Zeng Xiaofan Jia Liu xue Jie Huang

College of Communication Engineering, Chong Qing University

Chongqing 400040, China

{jamew, zxp, godisjxf, maojiu2005, hj_1988}@cqu.edu.cn

Received 10 October 2014; Revised 06 November 2014 ; Accepted 29 December 2014

Abstract. This paper firstly models fast retransmission and recovery in TCP selective acknowledgement (SACK). The model of fast retransmission and recovery in TCP SACK is represented by the expected number of packets transmitted during fast retransmission and recovery in TCP SACK and the corresponding expected duration. Then, based on Gilbert model and the model of fast retransmission and recovery, a steady state throughput model (full model) for TCP SACK is modeled and expressed as a function of burst event rate, burst loss rate and average round trip time. Finally, the full model of TCP SACK is validated under one bottleneck scenarios and multiple bottlenecks scenarios with HTTP, FTP and UDP traffics. The results show that the full model estimates the throughput of TCP SACK flow more accurate than TST model under high bandwidth delay product (BDP) networks. Furthermore, the results also indirectly validate the correctness of the fast retransmission and recovery model in TCP SACK and reveal that the smaller the BDP, the less bursty the losses.

Keywords: Congestion control, TCP SACK, Throughput model, Fast recovery.

1 Introduction

The weekly summary reports and daily reports of Internet2[1] show that around 90% of Internet traffic is carried by the Transmission Control Protocol (TCP), which provides reliable end-to-end transfer over Internet. In Internet Official Protocol Standards (STD 1)[2], a selective acknowledgement (SACK) mechanism[3], combined with a selective retransmission policy is recommended to be implemented. SACK mechanism consists of two options: (1) “SACK-permitted” option. It is sent in a SYN packet to indicate that SACK option can be used once the connection is established. (2) SACK option. It contains at most 4 blocks in each packet to inform the sender of non-continuous data blocks that have been received and queued. The selective retransmission policy refers to the SACK based loss recovery algorithm[4] which is used in fast retransmission and fast recovery[5]. In practice, employment of SACK options include the use of SACK options and a SACK-based recovery algorithm. TCP SACK or other TCP variants employing SACK options enable TCP sender to recover more effectively when multiple losses occurs in a single window.

Most models on the steady state throughput of TCP are based on assumption that packets within a round are sent back-to-back: in other words, once a packet is lost, all packets sent after the lost packet in that round are also lost[6]. Here a “round” starts when a packet in a window is transmitted and ends when the corresponding ACK is received. As the deployment of high speed links and active queue management policies [7], recent Internet loss studies[8, 9] show that some flows suffer from higher loss rate or lower loss rate while very few flows were observed a loss rate similar to the average loss rate. The studies also show the length of consecutive packet loss is small. Furthermore, Internet health report[10] show that packet loss rate during small time scare is very high, packet loss rates are very high during loss periods. Therefore, when congestion or loss occurs in a large window, which is usual in today’s high bandwidth-delay product (BDP) networks[11], the packet loss of a flow is assumed to be burst losses with a high loss rate. In this paper, this is modeled by a Gilbert model[12].

An analytic throughput model of TCP Reno was proposed in [6], which is a function of loss rate and round trip time. Parvez et al. followed this method and modeled the throughput of TCP NewReno under a two parameter loss model[13]. Dunaytsev assumed that there would be only one packet over the wireless channel in [14]. Most existing models of TCP SACK are still based on partial ACK recovery mechanism [15, 16]: upon receipt of a duplicate ACK during fast recovery, *pipe* is decreased by 1; upon receipt of a partial ACK, *pipe* is decreased by 2. Here *pipe* is a variable records the current number packets inflight upon receipt of a SACK. However, in the SACK-based loss recovery algorithm under Standards Track[4], *pipe* is defined to estimate the number of packets outstanding in the networks, which counts the number of packets inflight while removing the packet has been determined lost. A packet is determined lost whether either *DupThresh* (3) discontinuous SACK sequences have arrived above the given sequence number *SeqNum* or *DupThresh**SMSS bytes with sequence numbers greater than *SeqNum* have been SACKed[4]. Therefore, this paper models fast retransmission and recovery in TCP SACK based on [3] and [4]. Based on this, a full throughput of TCP SACK is proposed under burst losses. In addition, SACK option and SACK-based fast retransmission and recovery is steadily deployment in the Internet as a result of new Windows TCP stacks and most of Linux TCP stacks use it by default. The proposed model of fast retransmission and recovery in TCP SACK would make it easy to model TCP variants employing SACK options.

The rest of this paper is organized as follows. Section 2 presents the model of fast retransmission in TCP SACK. The full throughput model of TCP SACK is modeled in section 3. Section 4 validates the full throughput model. Finally, Section concludes the paper.

2 Loss Model

Gilbert model[12], showed in Fig.1, has two states(Good and Bad) and three parameters. Within Good State packets are never lost. Within Bad state, packet is droped with a probability q. Good state transfers to bad (burst) state with a probability p, at the same time, a packet is droped. The probability p is also called burst event rate. The transition probability from busrt state to good state is represented by l. Burst length B is often used instead of l, where B=1/l. Gilbert model is capable to model burst losses, or losses in high BDP networks.

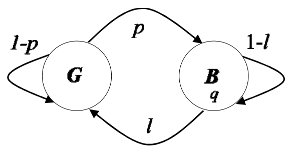


Fig. 1. Gilbert model

3 Model of Fast Retransmission and Recovery in TCP SACK

Packets transmission process with multiple losses is showed in Fig. 2. The notations used in this this paper are listed in Table 1. A sender transfers from congest avoidance phase to fast retransmission when 3 duplicate packets is received. After fast retransmission, it transfers to fast recovery phase. As a result, *pipe* is set to W-3. When a duplicate SACK or partial SACK is received, the sender updates *pipe*. Then it retransmits lost packets or sends new packets when *pipe* is less than the current window W/2. Following assumptions are used throughout this paper:

(1) Information contained in the 4 blocks of the SACK option is enough to inform the sender all the blocks that have been queued.
(2) Burst length B is less than W. Due to the limitation of the window in congestion avoidance phase, the burst length B is less than W.
(3) The model focus on the steady-state throughput of TCP bulk transfer.
(4) There is no ACK lost.

Note: a loss round starts when a packet lost and ends when the first duplicate SACK is received.

In fact, modeling fast retransmission and recovery in TCP SACK is to model the expected number of packets transmitted during fast retransmission and recovery (FRR) and the corresponding expected duration. Packets transmitted during FRR can be divided into three parts: retransmitted packets, new packets transmitted between the first retransmission and the last retransmission, new packets transmitted after the last retransmission. Therefore, the expected number of packets transmitted during FRR can be obtained as

$$S_{FRR} = N_R + N_t + N_a \ . \quad (1)$$

where S_{FRR} denotes the expected number of packets transmitted during FRR. N_R denotes the expected number of packets retransmitted. N_t denotes the expected number of new packets transmitted between the first retransmission and the last retransmission. N_a denotes the expected number of new packets transmitted after the last retransmission.

Since loss of retransmitted packet is considered as a new congestion indication[5], the number of retransmission during FRR equals the number of losses in W. Hence, N_R equals the expected number of losses under expected maximum window $E[W]$. so we can obtain

$$N_R = E[m] = \sum_{j=2}^{E[W]-1} (j-1)\cdot C_{E[W]-1}^{j-1} q^{j-1} (1-q)^{E[W]-j} + 1 \approx 1 + (E[W]-1)q \ . \quad \textbf{(2)}$$

Table 1. Model notations

Name	Definition	Name	Definition
W	size of congestion window when loss occurs	m	Number of packet losses in W
$E[W]$	Expected size of maximum window	p	Burst event rate
q	Burst loss rate	R	Average round trip time

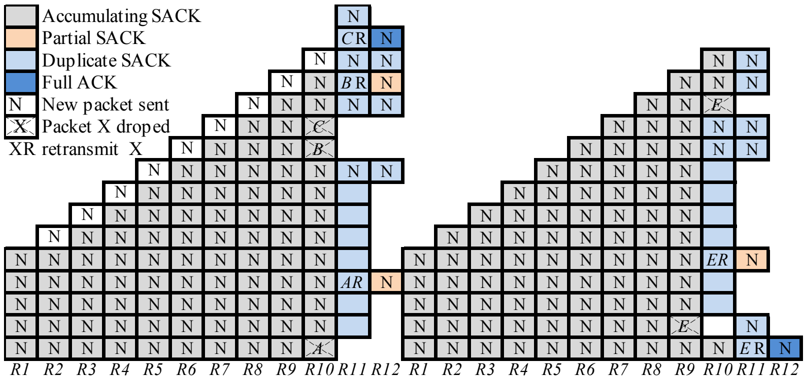


Fig. 2 Packets transmission with multiple losses and evolution of window size

On the other hand, after retransmission of the last loss in W, the sender deceases *pipe* upon receipt of a duplicate or partial SACK. A new packet would be allowed to transmit since the window stays at W/2 in FRR. When the full ACK is received, there would be W/2-1 new packets transmitted after retransmission of the last loss. Therefore, N_a can be obtained as

$$N_a = E[W]/2-1 \ . \quad \textbf{(3)}$$

In order to calculate N_t, let's firstly consider when m losses occur among the first (W/2+m-2) packets in W. Since fast retransmission is triggered by a loss and *pipe* is set to W-3, There are at most W/2 duplicate ACKs before identification and retransmission of the last loss. As a result, *pipe* would not be less than W/2. Therefore, the sender can not transmit any new packet before retransmission of the last loss. Use the Binomial distribution, we can get the probability of this case as

$$C_{W/2+m-2}^{m-1} p^{m-1}(1-p)^{W-m} \ . \quad \textbf{(4)}$$

When the m losses occur among the first (W/2+m-1) packets but not all among the first (W/2+m-2) packets, the (W/2+m-1)th packet is the last loss in W. In this case, the last loss is identified by the SACK of the (W/2+m+2)th packet transmitted in the lost round. As a result, the sender would receive (W/2+2) duplicate SACKs which decrease *pipe* to W/2-1 when the last loss is retransmitted. Therefore, the sender can transmit one new packet before retransmission of the last loss. There are m-2 free losses among (W/2+m-3) packets. The probability of this case can be obtained as

$$C_{W/2+m-3}^{m-2} p^{m-1}(1-p)^{W-m} \ . \quad \textbf{(5)}$$

Now consider a common case. When the m losses are among the first (W/2+m-3+i) ($i \ge 2$) packets but not all among the first (W/2+m-4+i) packets, the sender would receive (W/2+i) duplicate SACKs before identification and retransmission of the last loss. The same as in the former case, these SACKs would allow the sender to transmit (i-1) new packets. The probability of this is

$$C_{W/2+m+i-5}^{m-2} p^{m-1}(1-p)^{W-m}, 2 \le i \le W/2-m+3 \ . \quad \textbf{(6)}$$

When m>W/2, the sender would not transmit any new packets before retransmission of the last loss. When m=1, there is only one retransmission. Use the Binomial distribution again and substitute W with $E[W]$, N_t is obtained as

$$N_t = \sum_{m=2}^{E[W]/2}\left(\sum_{i=2}^{E[W]/2-m+\gamma}(i-\gamma+2)\cdot C_{E[W]/2+m-\gamma+i-2}^{m-2}p^{m-1}(1-p)^{E[W]-m}\right). \quad \textbf{(7)}$$

Therefore, S_{FRR} can obtained as

$$S_{FRR} = \frac{E[W]}{2} + \left(E[W]-1\right)p + N_t . \quad \textbf{(8)}$$

During FRR, all the losses can be determined at the beginning of the second round. Since the window is limited by $W/2$, the sender sends $W/2$ packets every round during FRR. On the other hand, FRR lasts at least one round even there is only one loss. Therefore, the expected duration of fast retransmission and recovery D_{FRR} can be obtained as

$$D_{FRR} = \max\left(\frac{S_{FRR}}{W/2}R, R\right) . \quad \textbf{(9)}$$

In order to model the steady state of TCP variants with SACK option using (7) and (8) directly. We try to simplify (7) with an approaximated expression. Fig. 3 shows N_t with respect to q. The results show that N_t decreases almost linearly as q increases when q>0.04. Owing the overlap of lines when N_t is plotted with respect to W, only lines of q>0.04 are showed in Fig. 4. The results show that N_t increases linearly with respect to W. As we have explained in the section 1, burst loss rate is usually much higher than average loss rate. It is reasonable to assume a high burst loss rate (q>0.04). Therefore, the expression of N_t is approximated by a function of p and W. The approximated expression of N_t in (10) is obtained when q ranges from 0.05 to 0.5 and W ranges from 50 to 1600. Fig. 5 shows the approximated results. The results show the approximation is quite satisfactory.

$$N_t = 0.9853W(0.5-p)-1.323 . \quad \textbf{(10)}$$

Therefore, equation (8) can be rewritten as

$$S_{FRR} = 0.9927E[W]+\left(0.0147E[W]-1\right)q-1.323 . \quad \textbf{(11)}$$

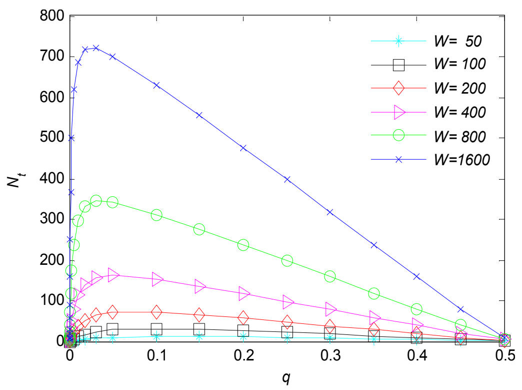


Fig. 3. N_t V.S. q

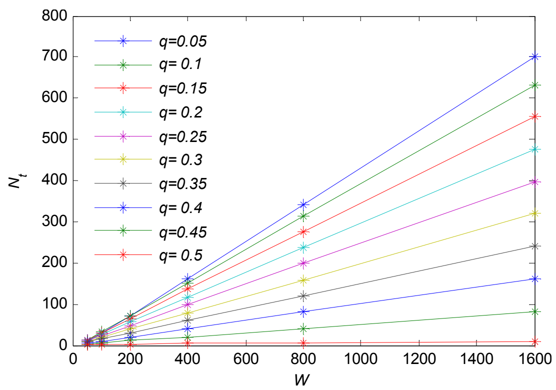


Fig. 4. N_t V.S. W

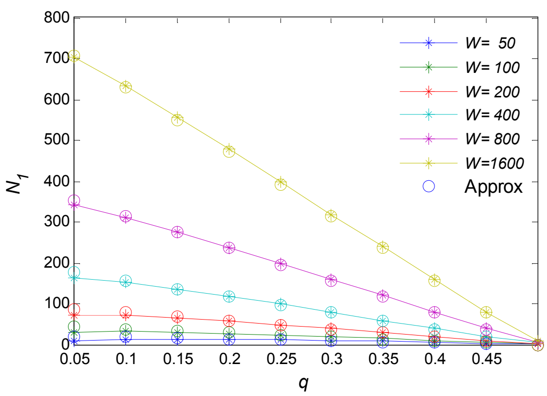


Fig. 5. Approximation result

4 Throughput Model of TCP SACK

Without timeout, the evolution of the congestion window can be viewed as the concatenation of triple duplicate periods (TDP). Each TDP consists of a congestion avoidance phase, a fast retransmission and a fast recovery. The steady state of the TCP flow is developed by concatenation of TDPs. Therefore, the throughput model of TCP with timeout is built by modeling one TDP of a TCP flow. The full model is derived by adding a slow start phase and a timeout phase to concatenation of TDPs.

4.1 None Timeout Model(NTO)

The packet transmission is illustrated in Fig. 2. Let S_{TDP} be the expected number of packets transmitted in a TDP, D_{TDP} be the expected duration of the TDP. The average throughput of a TCP flow is

$$T_{TDP} = \frac{S_{TDP}}{D_{TDP}} \ . \tag{12}$$

Let p, q and $E(B)$ be the burst event rate, burst loss rate and burst length of Gilbert model, respectively. As we have explained before, $E(B)$ is replaced by W, which represents the maximum burst length of a TCP flow. The expected number of packets transmitted between two burst events $E[\alpha]$ is obtained as

$$E[\alpha] = \sum_{k=1}^{\infty} (1-p)^{k-1} p^k = \frac{1}{p} \ . \tag{13}$$

Let $E[\delta]$ be the expected number of packets transmitted between the first loss and the last loss in a loss round and $E[m]$ be the expected number of losses in a loss round. Since each burst is started by a loss, there would be m-1 losses among the other W-1 packets. By assuming these losses are uniformly distributed, $E[m]$ and $E[\delta]$ can obtained as

$$E[m] = 1 + (E[W]-1)q \ . \tag{14}$$

$$E[\delta] = E[W] - \frac{E[W]-1}{2(E[m]-1)} \ . \tag{15}$$

Then the expected number of packets transmitted in a TDP is

$$S_{TDP} = E[\alpha] + E[\delta] = \frac{1}{p} + E[W] - \frac{E[W]-1}{2(E[W]-1)} = \frac{1}{p} + E[W] - \frac{1}{2q} \ . \tag{16}$$

One the other hand, as showed in Fig. 2, S_{TDP} and D_{TDP} can be expressed as

$$S_{TDP} = S_{LI} + S_{\beta} + S_{FRR} - E[m] \ . \tag{17}$$

$$D_{TDP} = D_{LI} + D_{\beta} + D_{FRR} \ . \tag{18}$$

where S_{LI} and D_{LI} are the expected number of packets transmitted and the duration in linear increase phase, respectively, S_{β} and D_{β} are the expected number of packets transmitted and the duration from the beginning of next round after the first packet loss to the termination of the congestion avoidance phase by triple duplicate ACKs, respectively. By referring to [6, 13], we can get

$$S_{LI} = \frac{3}{8}E[W]^2 + \frac{3}{4}E[W] \ . \tag{19}$$

$$D_{LI} = \left(\frac{E[W]}{2} + 1\right)R \ . \tag{20}$$

$$S_{\beta} = \frac{E[W]-1}{2} \ . \tag{21}$$

$$D_\beta = \frac{R}{E[W]} S_\beta = \frac{R(E[W]-1)}{2E[W]} \ . \qquad \textbf{(22)}$$

Now, by equating the right hand sides of (15) and (16), we have

$$\frac{1}{p} + E[W] - \frac{1}{2q} = S_{LI} + S_\beta + S_{FRR} - E[m] \ . \qquad \textbf{(23)}$$

Substitute all the intermediate variables into (23) and neglect the higher order terms. The expression of $E[W]$ is obtained approximately as

$$E[W] \approx \sqrt{\frac{24q - 12p + 84pq - 40pq^2}{9pq}} + \frac{4q-5}{3} \ . \qquad \textbf{(24)}$$

Finally, substitute (16) and (18) and all the intermediate variables into (12), average throughput of a TCP flow without timeout is obtained as

$$T_{NTO} \approx \frac{\left(2q-p\right)E[W] + 2E[W]^2 pq}{\left(E[W]^2 + 7E[W] + 0.04E[W]q - 4q - 10.64\right)pqR} \ . \qquad \textbf{(25)}$$

4.2 Full Throughput Model

Average throughput of TCP SACK can be expressed[13, 15] as

$$T_{Full} = \frac{\left(1-p_{TO}\right)S_{TDP} + p_{TO}\left(S_{LI} + S_{SS}\right)}{\left(1-p_{TO}\right)D_{TDP} + p_{TO}\left(D_{LI} + D_\beta + D_{TO} + D_{SS}\right)} . \qquad \textbf{(26)}$$

where p_{TO} is the probability of the timeout during a TDP, S_{SS} is the expected number of packets transmitted in slow start phase, and D_{SS} is the expected duration in the slow start phase. Here the sender ignores all the out-standing data when timeout happens. For TCP SACK, timeout occurs under two cases: 1) there are too many losses in W (m>W-3); 2) loss of any retransmitted packets. Therefore,

$$p_{TO} = \sum_{m=E[W]-2}^{E[W]} C\left(E[W]-1, m-1\right) q^{m-1} \left(1-q\right)^{E[W]-m} + \sum_{m=2}^{E[W]-3} \left(1 - \left(1-p\right)^m\right) \qquad \textbf{(27)}$$

By referring to [13, 15], we can get

$$S_{SS} = 2^{\left(1 + lb\frac{E[W]}{4}\right)} - 1 \cdot \qquad \textbf{(28)}$$

$$D_{SS} = \left(\log_2 \frac{E[W]}{4} + 1\right) R . \qquad \textbf{(29)}$$

$$D_{TO} = RTO \frac{1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6}{1-p} . \qquad \textbf{(30)}$$

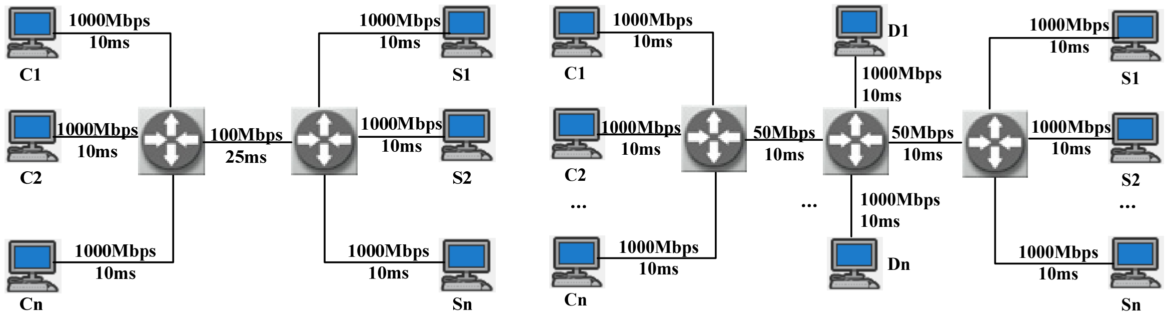
Substitute these variables into (26), full average throughput of TCP SACK can be obtained as

$$T_{Full} \approx \frac{\frac{1}{p} + E[W] - \frac{1}{2q} + \frac{p_{TO}}{1-p_{TO}}\left(2^{\left(1+\log_2\frac{E[W]}{4}\right)} + \frac{3}{8}E[W]^2 + \frac{5}{4}E[W]\right)}{D_{TDP} + \frac{p_{TO}}{1-p_{TO}}\left[\left(\log_2 \frac{E[W]}{4} + \frac{E[W]+5}{2}\right)R + \frac{1+p}{1-p}RTO\right]} . \qquad \textbf{(31)}$$

where $\mathrm{D}_{TDP} = \frac{E[W]^2 + 7E[W] + 0.04E[W]q - 10.64 - 4q}{2E[W]} R$, RTO is function of R.

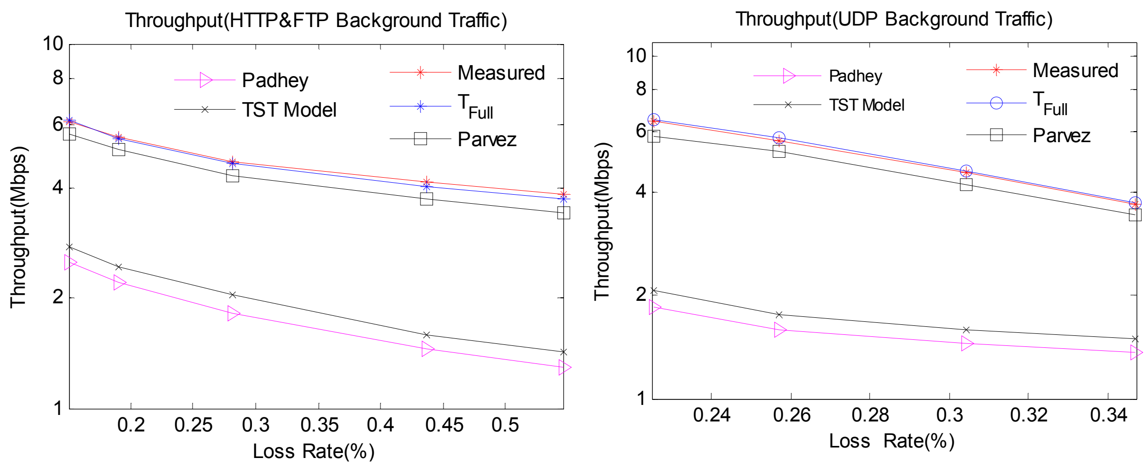
5 Validation

A dumbbell topology and a multiple bottlenecks topology are used to validate the full model, as shown in Fig. 6. The network parameters in Fig. 6 are set to simulate high bandwidth-delay product (BDP) networks[11]. HTTP, FTP traffics and UDP [19] (CBR, 1Mbps) traffics are used. The inter-request time of HTTP traffics obeys exponential distribution (mean, 0.5s) with web page size is generated by a Pareto distribution (mean size, 48 kB, shape, 1.2). The "Drop tail" policy is adopted by routers in the bottleneck queue management. The traced flow uses TCP SACK and other flows use either TCP Reno or TCP SACK randomly. The simulation lasts 200s and the results reported are measured from 50s to 200s. Burst event rate, burst loss rate, average loss rate are obtained from the trace files in all simulations. The full model are compared with Parvez's model[13], Padhye's model[6] and the TST model[15]. All the figures are plotted with respect to loss rate (overall loss rate).



(a) (b)

Fig. 6. (a) Dumbbell topology. (b) Multiple bottlenecks topology.



(a) (b)

Fig. 7. HTTP/FTP and UDP traffics. (a) Varied HTTP/FTP background flows. (b) Varied UDP background flows.

5.1 Dumbbell Topology Validation

To test the model under real traffics environment, 40 HTTP flows, 10 FTP flows and 10 UDP basic background flows are created between the clients and the servers in all scenarios. The measured results are the average results obtained by running each scenario 10 times.

Fig.7(a) shows the throughputs of the simulations and the models when the number of HTTP/FTP flows are varied from 50 to 90. Fig.7(b) shows the results when the number of UDP background flows varies from 10 to 40. Results show that the full model estimates the throughputs of TCP SACK more accurate. Parvez's model predicts the throughputs with an offset even it uses a simple Gilbert loss model. This can be attributed to the different recovery algorithms used by TCP NewReno and TCP SACK. The TST model and the Padhye's model fail to predict the throughputs of TCP SACK under these scenarios. By examining the measured parameters, we find that burst event rates is much less than average loss rates. The higher burst loss rates also verify this. This is in accordance with the expectation under high BDP networks that the losses are closely correlated[17, 18]. The results also reveals that burst event rate or average loss rate, in other words, the congestion indication rates contributed more to the throughput of a TCP flow than the burst loss rate within a burst event. Since when the number of flows increases, burst length decreases but the decrease in burst loss rates is not significant. The large differences between the full model and TST model also reveal that uniform losses (often used when generate random losses) is not suitable to simulate losses in high BDP networks. On the other hand, the results verifies the use of Gilbert model in modeling burst losses. Moreover, the results show that the larger the bandwidth-flows ratio, the more bursty the losses.

5.2 Multiple Bottlenecks Topology Validation

To provide a more real and comprehensive validation, the multiple bottlenecks topology is used as shown in Fig. 6(b). The background traffic consists of 40 HTTP flows, 10 FTP flows and 5 UDP flows between pairs of client *Ci* and server *Di* or *Si*. Fig. 8(a) shows the average throughputs of the models when the number of HTTP/FTP background flows between *Ci* and *Si* are varied from 50 to 90. The results show that all the throughputs decreases as the number of HTTP/FTP flows increases since there are more flows participating in sharing the bandwidth. The results also show that the full model predicts the throughputs of TCP SACK flows more accurate. Parvez's model estimates the throughput with an offset again. However, the prediction errors increases a little as the loss rate increases. This can also be attributed to differences in recovery algorithms of the two protocols. As the number of flows increases, the congestion period becomes short or the burst event rate increases. This highlights the differences or predicted errors. TST model and Padhye's model fail to trace the throughputs of TCP SACK again. However, the prediction errors decrease in multiple bottlenecks scenarios as the number of flows increases. This is due largely to the increase of burst event rates.

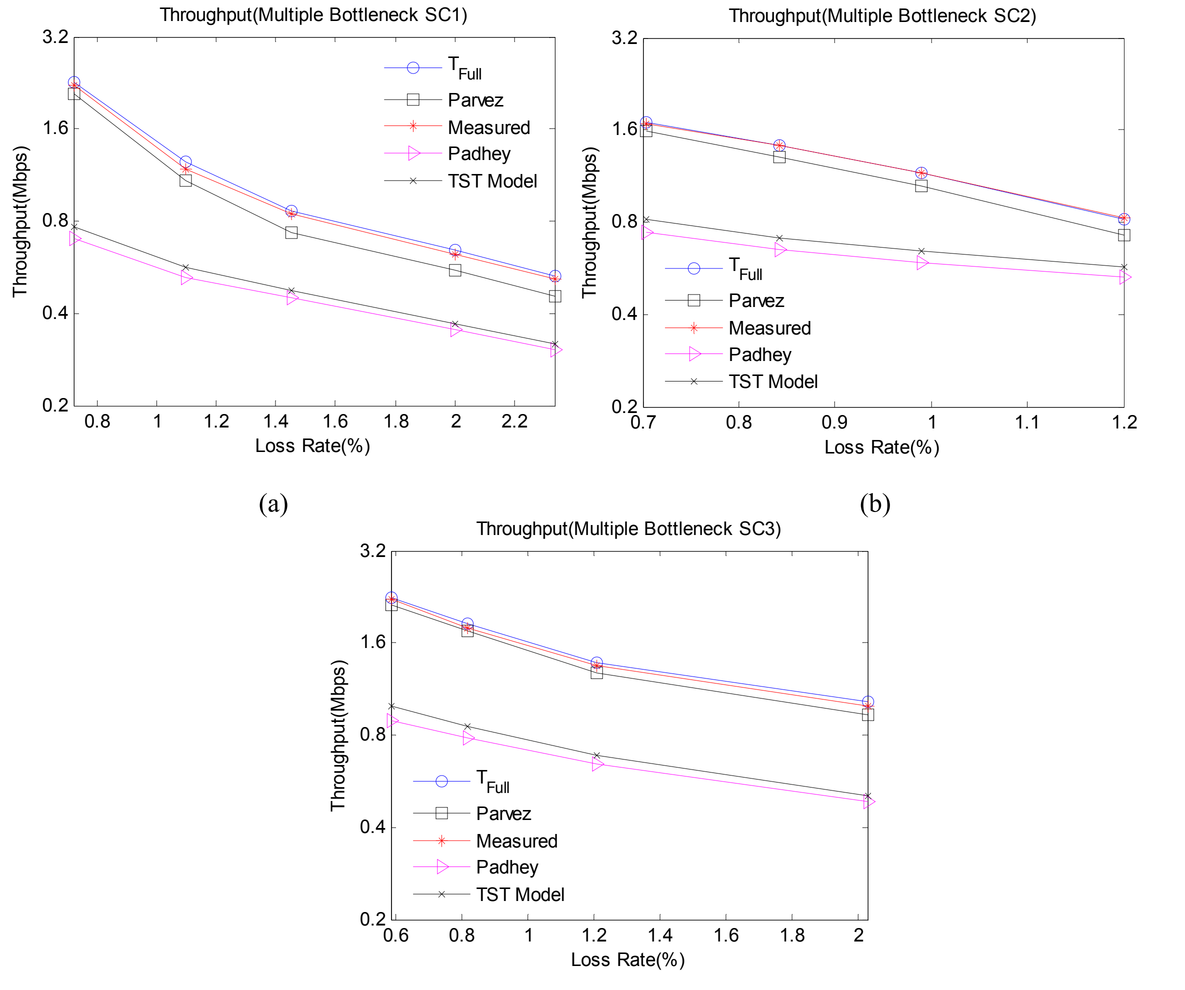


Fig. 8. Multiple Bottlenecks Simulation. (a) HTTP/FTP flows between *Ci* and *Si*. (b) UDP Flows between *Di* and *Si*. (c) HTTP/FTP Flows between *Ci* and *Di*

Fig. 8(b) shows results of the models and the simulations when the number of UDP background flows varies from 10 to 40 between *Di* and *Si*. Fig. 8(c) shows results when the number of HTTP/FTP background flows between *Ci* and *Di* varies from 10 to 40. The results in Fig. 8(b) and (c) show the same tendency with Fig. 8(a) that the more flows, the lower average throughput of the measured flow. By comparing the results of multiple bottlenecks scenarios, we find that the more flows sharing the bottleneck bandwidth, the higher the burst event rate. From Fig. 8, we can also find that the less average bandwidth the measured flow gets, the lower the loss rate of the flow achieves. On the other hand, the burstiness of losses (burst length or burst loss rate) decreases in multiple bottlenecks scenarios. It can also be inferred that if more number of flows is added or the less bandwidth is taken by the measured flow, less predicted error of Padhye's model and TST model are achieved.

Overall, the full model estimates the throughputs of TCP SACK under high BDP networks more accurate than other models. This indirectly validates the correctness of the fast retransmission and recovery model in TCP SACK. Parvez's model estimates the throughputs with an offset due to the different recovery algorithm. TST model and Padhye's model fails to estimate the throughputs accurate owing largely to the differences between burst losses and random(uniform) losses. The less the bandwidth of a flow, the less bursty the losses of the flow. In short, the model of FRR and the full throughput model is more accurate than TST model in high BDP networks. The smaller the BDP, the less bursty the losses.

6 Conclusion

This paper models fast retransmission and recovery in TCP SACK, derives the expected number of packets transmitted during fast retransmission and recovery and the corresponding expected duration. In fact, the fast retransmission and recovery model in TCP SACK also represents models of fast retransmission and recovery in all TCP stacks employing SACK options. The fast retransmission and recovery model would make it easy to model TCP variants employing SACK options. Based the fast retransmission and recovery model and Gilbert model, a steady state throughput model (full model) of TCP SACK is developed as a function of burst event rate, burst loss rate and round trip time. Finally, the full model of TCP SACK is validated under one bottleneck scenarios and multiple bottlenecks scenarios with HTTP, FTP and UDP traffics. The results show that the full model estimates the throughputs of TCP SACK more accurate in high BDP networks than TST model. Furthermore, the results also indirectly validate the correctness of the fast retransmission and recovery model in TCP SACK and reveal that the smaller the bandwidth delay product, the less the bursty the losses.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under grant 61171089.

References

[1] S. Stanislav and T. Benjamin, "Internet2 NetFlow: weekly reports," Available from: http://netflow.internet2.edu/weekly/.

[2] S. Ginoza and J. Reynolds, *Internet Official Protocol Standards*, RFC 3700, 2004.

[3] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, *TCP Selective Acknowledgement Options*, RFC 2018, 1996.

[4] E. Blanton, M. Allman, L. Wang, I. Jarvinen, M. Kojo and Y. Nishida, *A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP*, RFC 6675, 2012.

[5] M. Allman, V. Paxson and E. Blanton, *TCP congestion Control*, RFC 5681, 2009.

[6] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, 2000, pp. 133-145.

[7] V.P. Rao, M.P. Tahiliani and U.K.K. Shenoy, "Analysis of sfqCoDel for Active Queue Management," in *Proc. Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*, 2014, pp. 262-267.

[8] M.A. Mehmood, N. Sarrar, S. Uhlig and A. Feldmann, "Understanding flow performance in the wild," in *Proc. Global Communications Conference (GLOBECOM), 2013 IEEE*, 2013, pp. 1410-1415.

[9] M.A. Mehmood, N. Sarrar, S. Uhlig and A. Feldmann, "Impact of access bandwidth on packet loss: A flow-level analysis," in *Proc. Communications (MICC), 2013 IEEE Malaysia International Conference on*, 2013, pp. 259-264.

[10] Keynote, "Internet health report," Available from: http://www.internetpulse.net/.

[11] L. Sunghee and C. Kwangsue, "Congestion Control Scheme for Multimedia Streaming Service over High Bandwidth Delay Product Networks," in *Proc. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, 2014, pp. 276-281.

[12] S. Salsano, F. Ludovici, A. Ordine and D. Giannuzzi, *Definition of a general and intuitive loss model for packet networks and its implementation in the netem module in the linux kernel,* Technical report, University of Rome Tor Vergata, 2012.

[13] N. Parvez, A. Mahanti and C. Williamson, "An Analytic Throughput Model for TCP NewReno," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 2, 2010, pp. 448-461.

[14] R. Dunaytsev, D. Moltchanov, Y. Koucheryavy and J. Harju, "Modeling TCP SACK performance over wireless channels with completely reliable ARQ/FEC," *International Journal of Communication Systems*, vol. 24, no. 12, 2011, pp. 1533-1564.

[15] X. Wei, X. Yinlong, W. Xiaohu and O. Kaiqian, "Modeling TCP Sack steady state performance in lossy networks," in *Proc. Information Networking (ICOIN), 2011 International Conference on*, 2011, pp. 278-283.

[16] T.L. Sheu and L.W. Wu, "An analytical model of fast retransmission and recovery in TCP-SACK," *Performance Evaluation*, vol. 64, no. 6, 2007, pp. 524-546.

[17] C. Jin, C. Aiyou and P.P.C. Lee, "Modeling time correlation in passive network loss tomography," in *Proc. Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*, 2011, pp. 550-561.

[18] H.X. Nguyen and M. Roughan, "On the correlation of internet packet losses," in *Proc. Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian, IEEE,* 2008, pp. 22-27.

[19] Shih, C. H., Ke, C. H., & Xu, Y. Y. Efficient Packetization for DCCP Flows over ARQ-based Wireless Networks. Journal of Computers (Taiwan), vol. 24, no. 1, 2013, pp. 1-13.