

Modeling of SACK-based Fast Retransmission and Recovery

Zhiming Wang, Xiaoping Zeng, Xiaofan Jia, Xue Liu and Li Chen

*College of Communication Engineering, Chongqing University, Chongqing
400030, China.*

E-mail:jamewzm@163.com

Abstract

Selective acknowledgment (SACK)-based retransmission and recovery mechanism has been implemented in most of current transmission control protocol (TCP) stacks. A new SACK-based loss recovery algorithm is defined in RFC6675 under standard track. Existing models in literatures on modeling TCP SACK, which usually use SACK-based retransmission and recovery mechanism, is no longer accurate for two reasons: 1) they ignore transmission of new packets before retransmission of the last loss; 2) they are based on the strategy that pipe is decreased by 2 upon receipt of a partial ACK, where pipe is the estimated number of outstanding packets. This paper develops a new method to derive the expected number of new packets transmitted between retransmission of the first loss and the last, proposes a model that estimates the expected number of packets transmitted during fast retransmission and recovery phase (FRR) and the corresponding expected duration based on RFC6675. The numerical results show that the proposed model can accurately estimate the expected number of packets transmitted during FRR and the corresponding expected duration based on RFC6675.

Keywords: *Congestion Control; TCP SACK; Throughput Modeling; Fast Recovery*

1. Introduction

Transmission Control Protocol (TCP) provides reliable host-to-host data delivery on the Internet. It is reported that around 90% of Internet traffic is carried by TCP. Most of web servers implement TCP congestion control protocol with selective acknowledgement (SACK) options [1, 2]. In 2012, Internet Engineering Task Force introduced a new SACK-based loss recovery mechanism RFC6675 [3] in Internet official protocol standard (RFC3700) to improve the TCP performance under burst loss networks. The SACK options combined with a SACK-based loss recovery mechanism, known as SACK-based fast retransmission and recovery mechanism, can significantly improve the performance of TCP in burst loss rate networks [2-4]. Therefore, modeling SACK-based fast retransmission and recovery is significant for accurately modeling and analyzing the performance of TCP that employs SACK options.

Most existing models of TCP SACK, which uses SACK-based retransmission and recovery mechanism, are based on partial ACK recovery mechanism [14, 15]: 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 to record the current number packets inflight upon receipt of a SACK. However, in RFC6675, *pipe*, which counts the number of packets inflight while removing the packet has been determined lost, estimates the number of packets outstanding in the networks. A packet is marked lost when either certain number (duplicate threshold) of discontinuous SACKed (selective acknowledged) sequences higher than its sequence have arrived or certain size of data with sequence numbers greater than its

have been SACKed. This leads to that the existing model cannot accurately describe the fast retransmission and recovery process based on RFC6675.

Existing models in [14, 15] also ignores the transmission of new packets before retransmission of the last loss during SACK-based fast retransmission and recovery phase. In fact, new packets can be transmitted before retransmission of the last loss. An example of the packet transmission process using SACK-based fast retransmission and recovery with RFC6675 is illustrated in Fig. 1. In Fig. 1 each column represents a “round” [5]. A round starts when a packet (at the bottom) is transmitted and ends when the corresponding ACK (the box on the right-hand) is received. The top packet in each column is the one more packet transmitted in each congestion avoidance round. There are three packets (A, B and C) lost in R10. Packet A was detected lost by the third duplicate ACK received in R11 and retransmitted immediately (labeled by A(FRR)). At the same time, the size of congestion window was halved. Thus the size of congestion window during FRR was set to 7 and *pipe* was set to 11. Then the sender entered fast recovery. Receipt of the following 5 duplicate ACKs in R11 decreased *pipe* to 6 ($pipe < 7$), so the sender transmitted a new packet. No ACK had been received for lost packets (packet B and C). The ninth and the tenth duplicate ACK (SACK) in R11 acknowledged the twelfth and thirteenth packets in R10 which indicates that packet B is lost. A new packet was transmitted after the retransmission of packet B because receipt of the tenth duplicated ACK also decreased *pipe* by one. Then a similar process was carried out for Packet C. The receipt of partial and duplicate ACKs in R12 allowed the sender to transmit 5 new packets. Finally, the receipt of the full ACK (the dark blue box) terminated the fast recovery phase and started a new congestion avoidance phase. In the above example, there are 3 new packets transmitted before retransmission of the last loss C.

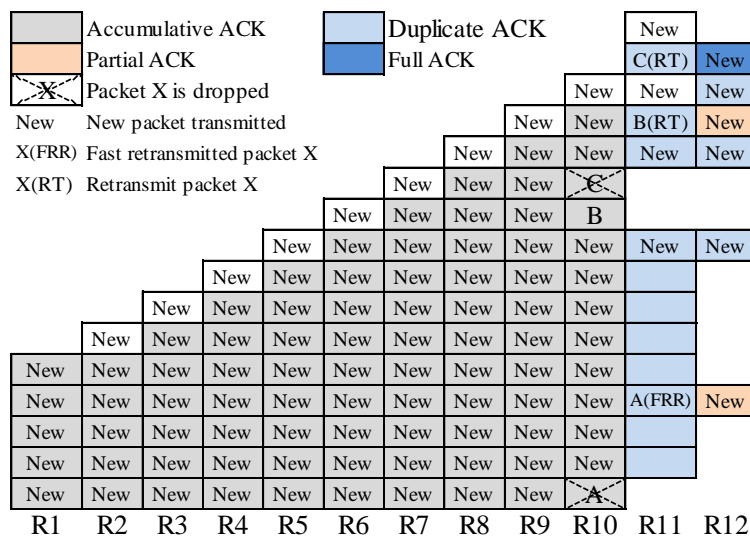


Figure 1. New Packets Transmission in Fast Retransmission and Recovery Phase

In networks where SACK-based retransmission and recovery is used, to evaluate the performance of TCP, accurate throughput model of TCP with SACK-based retransmission and recovery is needed. Up to our knowledge, there is no model about SACK-based fast retransmission and recovery with RFC6675 in terms of burst loss rate or packet loss rate. Considering there are many throughput models for TCP variants which have modeled the congestion avoidance phase, this paper focuses on modeling of SACK-based fast retransmission and recovery with RFC6675 (FRR). In addition,

without specific declaration, the loss recovery mechanism in RFC6675 is used in SACK-based fast retransmission and recovery.

Recently Internet loss studies show that the packet losses are strongly correlated[2,8], some flows suffer from higher loss rate or lower loss rate while very few flows observe a loss rate similar to the average loss rate [9, 10]. Burst loss model has been considered in modeling throughput of TCP flows. Parvez et al used a two-parameter loss model to model throughput of TCP NewReno [7]. ZENG Bin [12], et al introduced a special Gilbert loss model, which models the end-to-end burst loss characteristics, in modeling throughput of TCP. Furthermore, the real-time Internet healthy report [11] shows that some links suffer from much higher loss rate than other links and their short time loss rate is higher than 5%. All these studies indicate that the end-to-end flows suffer from burst losses. Therefore, Gilbert loss model [13] is assumed to be used to model the burst loss characteristics of the end-to-end flows in terms of burst loss rate, burst event rate and burst length. The burst loss rate denotes the packet loss rate within a burst. The burst length denotes the length of the burst. The burst event rate denotes the frequency of burst event. When burst length decreases to one, Gilbert loss model becomes a random loss model.

The remainder of this paper is organized as follows. Section 2 introduces the SACK-based loss recovery in RFC6675. The detailed modeling of SACK-based fast retransmission of recovery is presented in section 3. The model is validated using simulation in section 4. Section 5 concludes the paper.

2. SACK-based Loss Recovery

In the current SACK implementation, according to RFC 6675 [3], upon the receipt of the first and second duplicate ACKs, a sender does not transmit any packet. When the sender receives the third (*DupThresh*) duplicate ACK, it records the highest sequence number transmitted (*RecoveryPoint*) and the highest ACK number cumulatively ACKed, then starts a fast retransmission and recovery phase.

During the fast retransmission and recovery phase, the sender maintains a variable *pipe* to record the estimated number of packets outstanding in the network, which counts the number of packets inflight while removing the packet has been determined lost. Upon receipt of a duplicate ACK with SACK option, the sender marks the packets cumulatively ACKed or SACKed and determines packet losses using the rule whether either three (*DupThresh*) discontinuous SACK sequences have arrived above the given sequence number (*SeqNum*) or (*DupThresh *SMSS*) bytes with sequence numbers greater than *SeqNum* have been SACKed. Then it updates *pipe*.

Upon receipt of a partial ACK, which cumulatively acknowledges the packets in the loss window, the same steps are taken for partial ACK. Then the sender sends new or retransmitted segments depending on whether the congestion window is larger than *pipe* or not.

Finally, an incoming full ACK whose sequence number is greater than the *RecoveryPoint* terminates the fast retransmission and recovery phase.

3. The Model (SFRR)

Modeling of SACK-based fast retransmission and recovery includes modeling the expected number of packets transmitted S_{FRR} and the corresponding expected duration D_{FRR} in a FRR. The notation used is listed in Table I. To derive S_{FRR} and D_{FRR} , following assumptions are made.

1) The number of losses in W , denote by m , is no more than $(W-\gamma)$. This assumption ensures that there will be enough duplicate ACKs to invoke fast retransmission;

2) There is no packet loss during fast retransmission and recovery;

3) The other packets following the first loss in a loss window is deemed to be transmitted in a burst with burst loss rate p . A loss window represents a congestion window that starts with a packet loss. It corresponds to burst length in Gilbert loss model.

Table 1. Model Notations

Name	Definition
W	The size of congestion window when loss occurs
p	Expected size of maximum window
γ	Duplicate retransmission threshold
S_{FRR}	Expected number of packets transmitted in FRR
D_{FRR}	Expected duration of time in FRR
R	Average round trip time

Since there is no packet loss during fast retransmission and recovery phase, there is no time-out. All losses can be determined at the beginning of the second recovery round, the sender will only retransmit the losses before sending any new packets from the second recovery round on. Therefore, S_{FRR} can be calculated as follows.

$$S_{FRR} = N_R + N_1 + N_2. \quad (1)$$

where N_R is the expected number of retransmitted packets, N_1 is the expected number of new packets transmitted between retransmission of the first loss and retransmission of the last loss. N_2 is the expected number of new packets transmitted after retransmission of the last loss and before receipt of its ACK.

Firstly, considering there is no time-out, N_R obviously equals to the expected number of losses. When the first loss is detected, there are still $W-\gamma$ packets inflight. Assuming there are m losses in all, so the other $(m-1)$ losses are among the $(W-1)$ inflight packets. Therefore, N_R can be deduced as

$$N_R = \sum_{m=2}^{W-\gamma} (m-1) \cdot C_{W-1}^{m-1} p^{m-1} (1-p)^{W-m} + 1 \approx 1 + (W-1)p. \quad (2)$$

Secondly, the size of congestion window during FRR stays at $W/2$. The packets transmitted before retransmission of the last loss will all be ACKed before receipt of full ACK, and the sender can send $(W/2-1)$ new packets after retransmission of the last loss no matter how many losses are in a loss window, as in Fig. 1. Therefore, N_1 can be expresses as

$$N_1 = \frac{W}{2} - 1. \quad (3)$$

In order to determine N_1 , consider there are m losses in a loss window W . Firstly, when m losses are within the first $(W/2+m-\gamma+1)$ packets, there are at most $W/2$ duplicate ACKs before detection of the last loss. The $W/2$ duplicate ACKs only decrease $pipe$ to $W/2$. Therefore, the sender will not have any chance to transmit a new packet before retransmission of the last loss. The probability of this case is as follows

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

Then when the m losses are within the first $W/2+m-\gamma+2$ packets and not all within the first $W/2+m-\gamma+1$ packets, in other words, the last loss is the $(W/2+m-\gamma+2)$ th packet in the loss window W , there would be $W/2+1$ duplicate ACKs before detection of the last loss. The $W/2+1$ duplicate ACKs would decrease $pipe$ to $W/2-1$, this would allow the sender to transmit one new packet before retransmission of the last loss. The probability of this case is

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

Then when the last loss is the $(W/2+m-\gamma+i)$ th ($W/2-m+\gamma \geq i \geq 2$) packet in loss window W , there are $W/2+i-1$ duplicate ACKs before detection of the last loss. The $W/2+i-1$ duplicate ACKs would decrease *pipe* to $W/2-i+1$, this would allow the sender transmit $i-1$ new packet before retransmission of the last loss. The probability of this case is

$$C_{W/2+m-\gamma+i-2}^{m-2} p^{m-1} (1-p)^{W-m}, 2 \leq i \leq W/2-m+\gamma. \quad (6)$$

Then, N_1 can be obtained as the weighted sum of all possible cases with respect to m and i . When $m > W/2$, the sender has no chance to transmit new packet in the first recovery round. As stated before, it would not transmit any new packets before retransmission of all losses from the second retransmission round on. When $m=1$, the sender retransmits the loss, which is also the last retransmission. When $m=1$ or $m > W/2$, there is no new packet transmission between retransmission of the first loss and retransmission of the last loss. Thus N_1 can be obtained as follows.

$$N_1 = \sum_{m=2}^{W/2} \left(\sum_{i=2}^{W/2-m+\gamma} (i-1) \cdot C_{W/2+m-\gamma+i-2}^{m-2} p^{m-1} (1-p)^{W-m} \right). \quad (7)$$

Therefore, the expected number of packets transmitted during SACK-based fast retransmission and recovery is obtained as follows.

$$S_{FRR} = N_R + N_1 + N_2 = (W-1)p + \frac{W}{2} + N_1. \quad (8)$$

According to the assumption, all the losses can be determined at the beginning of the second round. With the consideration that *pipe* is limited by $W/2$, during each round of FRR, the sender can transmit $W/2$ packets. On the other hand, fast retransmission and recovery lasts at least one round even there is only one loss. Therefore, D_{FRR} can be obtained as

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

Up to now, S_{FRR} and D_{FRR} are determined. To facilitate the calculation of N_1 and use (8) and (9) in modeling throughput of TCP variants with SACK-based retransmission and recovery, we will try to approximate N_1 .

In order to approximate N_1 , a three-dimensioned shaded surface of N_1 is showed in Fig. 2. The results show that N_1 increases to a vertex and then decreases to zero as p grow from 0 to 0.5 when W is fixed. N_1 are approximately zero since $N_1 > 0.5$. The results also show that N_1 increases as W increases from 50 to 1600 when $0 < p < 0.5$. To find the detailed relationship, N_1 are plotted with W and p respectively in Fig.3 and Fig.4.

From Fig.3, we can see that N_1 increases quickly when p moves away from 0 and then decreases linearly to 0 as p moves to 0.5 when W is fixed. This agrees with the real fact when p approaches 0, there is likely just one loss (N_1 equals 0). The transmission chance of new packets before last retransmission also decreases to zero when p approaches 0.5 (half of inflight packets are lost).

On the other hand, Fig.4 shows N_1 with respect to W . Due the overlap of lines when N_1 is plotted with respect to W . Thus only lines when $0.05 < p < 0.5$ are showed. The results show that N_1 increases linearly as W increases when p is fixed. This agrees with the real fact when p is fixed by increasing W . The larger the congestion window W grows, the larger the number of packets allowed inflight are. The more packets inflight, the more average number of new packets can be transmitted before retransmission of the last loss.

Since Gilbert loss model is assumed to model the burst loss characteristics, recent Internet loss studies [2, 8-10] show that the burst loss rate is usually much higher

than the overall packet loss rate. The Internet experiments in [7] showed that burst loss rate is much higher than loss event rate. Internet health report [15] shows that the temporal loss rate is higher than 5%. These studies all indicate that assuming a high burst loss rate in a loss window is reasonable. Therefore, for practical calculation, N_1 is approximated under burst loss rate $p > 0.05$. Since when $0.05 < p < 0.5$, N_1 is proportional W and p . When $0.5 < p < 1$, N_1 are approximately zero. Thus N_1 are approximated separately when $0.05 < p < 0.5$ and when $0.5 < p < 1$ as follows.

$$N_1 \approx \begin{cases} -1.323 + 0.9853(0.5 - p)W, & 0.05 < p < 0.5 \\ 0, & 0.5 < p < 1 \end{cases} \quad (10)$$

The approximated expression of N_1 ($0.05 < p < 0.5$) in (10) is obtained by regression when p ranges from 0.05 to 0.5 and W ranges from 50 to 1600. Fig. 5 compares the approximated result and the theoretical result from (7). The results show the approximation is quite satisfactory and the overall approximation error is less 1%.

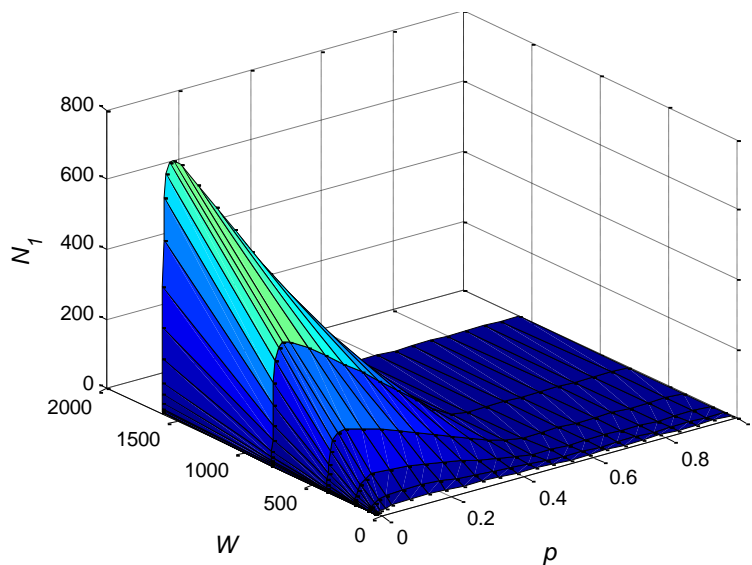


Figure 2. N_1 component in SFRR

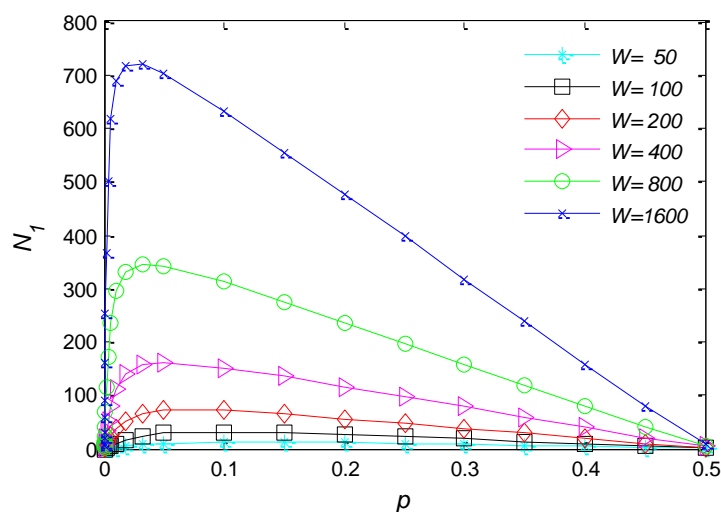


Figure 3. N_1 V.S. p

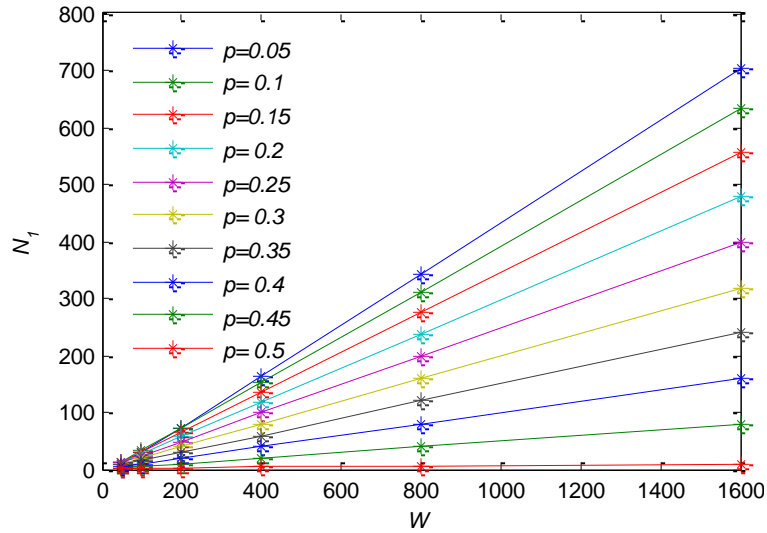


Figure 4. N_1 V.S. W

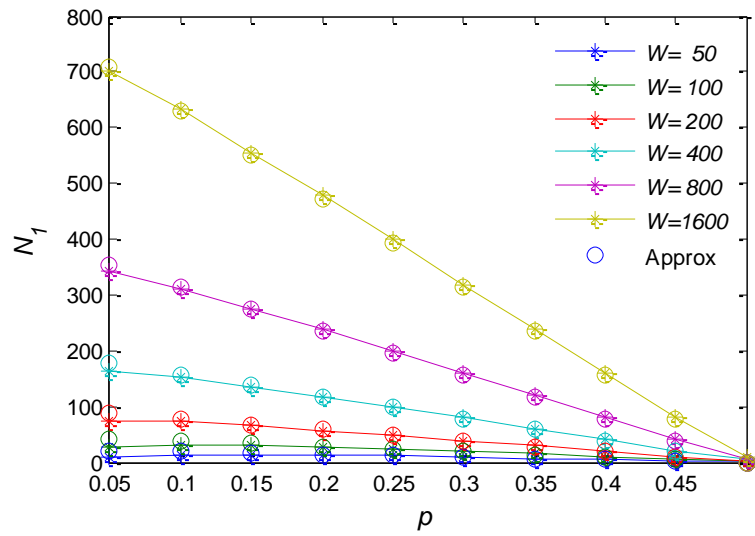


Figure 5. Approximation Result

Therefore, S_{FRR} can be obtained as

$$S_{FRR} = \begin{cases} 0.9927W + (0.0147W - 1)p + -1.323, & 0.05 < p < 0.5 \\ (W - 1)p + 0.5W, & 1 \geq p \geq 0.5 \end{cases} \quad (11)$$

4. Numerical Results

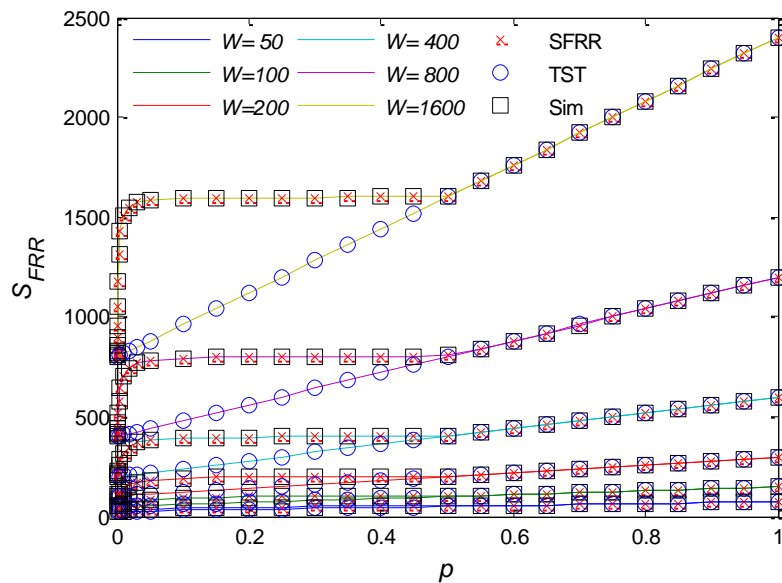


Figure 6. Expected Number of Packets Transmitted During SACK-Based Fast Retransmission and Recovery

From the derivation of S_{FRR} and D_{FRR} , we can see that S_{FRR} is only related to burst loss rate p and loss window W , which is also deemed as the maximum burst length. Therefore, in order to validate the proposed model, we first developed a program in MATLAB to emulate the selective acknowledge and retransmission mechanism according to RFC6675 and calculate the number of packets transmitted. The losses in a loss window are generated according to a uniform distribution under a given burst loss rate. Then Monte Carlo method is used to obtain the average number of packets transmitted during fast retransmission and recovery phase (simulated S_{FRR}). Finally, the simulated S_{FRR} is compared with the estimated results from the proposed model (SFRR) and TST model under given loss window W and burst loss rate p .

Fig 6 shows S_{FRR} that are obtained from SFRR, TST and simulation (legend Sim in Fig. 6). The lines with x-mark are values obtained from the formula (8), the blue circles are results from TST model [14] and the black squares are the average values obtained from 5000 times simulations at each point. The model in [15] is not compared here because it also works on partial ACK, which would obtain the same result as TST model. The results show that S_{FRR} obtained SFRR agrees well with simulated S_{FRR} while the TST model cannot be used to track the trend of S_{FRR} using RFC6675 when burst loss rate p ranges from 0 to 0.5.

From Fig. 6, we also find that S_{FRR} increases quickly with p when p is small. This agrees with the real fact that if there is only one loss, the sender will transmit $W/2$ packets during fast recovery. S_{FRR} approaches W when p is greater than 0.1. This is because when the number of losses increases, the last loss in a loss window tends to be at the end in loss window W . By using SACK-based loss recovery, which detects packet loss according to RFC6675 and removes it from *pipe*, the sender is able to transmit almost full window of packets ($W/2$) before detection and retransmission of the last loss. After retransmission of the last loss, the sender can also transmit $W/2-1$ new packets before receipt of full ACK. Therefore, during FRR, the sender sends approaching W packets when there are multiple losses. On the other hand, TST model assumes that *pipe* is decreased by 1 upon receipt of a duplicate ACK, by 2

upon the receipt of a partial ACK and just retransmits losses before all losses retransmitted. Therefore, S_{FRR} of TST increases linearly with burst loss rate.

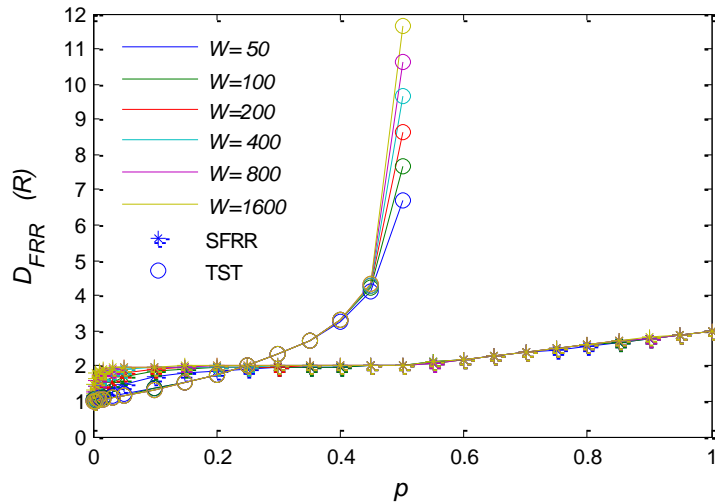


Figure 7. Expected Duration of SACK-Based Fast Retransmission and Recovery

D_{FRR} that are obtained from SFRR and TST model are showed in Fig 7. The results show that D_{FRR} of SFRR equals one average round trip time (R) when burst loss rate p is zero. It quickly approaches $2R$, stays at $2R$ as p increases within 0.5 and gradually increases to $3R$ as p increases from 0.5 to 1. According to RFC 6675 and assumption, all losses in a loss window can be determined at the beginning of the second round of the fast recovery phase. It would be reasonable that D_{FRR} is within $2R$ when the expected number of losses is less than $W/2$, or $p < 0.5$ and D_{FRR} gradually increases from $2R$ to $3R$ as p increases from 0.5 to 1.

The models in [14] and [15], assume that $pipe$ is decreased by 2 upon the receipt of a partial ACK. This assumption brings about time-out when the number of losses in a loss window W is more than $W/2+2$, which is approximately equivalent to $p > 0.5$. Thus there is no D_{FRR} of TST cannot be obtained when p is greater than 0.5. D_{FRR} of TST gradually accelerates to increase as p increases from 0 to 0.5. The results also shows that RFC6675-based loss recovery spends less time in FRR than the old recovery algorithm.

In all, the results show that the proposed model (SFRR) accurately estimates the expected number of packets transmitted during FRR using SACK-based loss recovery with RFC6675. TST model cannot be used to estimate S_{FRR} and D_{FRR} when using SACK-based loss recovery in RFC6675.

5. Conclusion

In this paper, we presented a model that estimates the expected number of packets transmitted during fast retransmission and recovery S_{FRR} and the corresponding expected duration D_{FRR} based on RFC6675. S_{FRR} and D_{FRR} are expressed as a function of loss window W , burst loss rate p and average round trip time R . An approximated expression of S_{FRR} is also provided. The numerical results show that the proposed model S_{FRR} can accurately estimates the expected number of packets transmitted during fast retransmission and recovery using SACK-based loss recovery with RFC6675. However, the model ignored the time-out situation due to retransmission loss or too much losses. The time-out situation should be considered when using this model in modeling throughput of TCP flows. Furthermore, the

approximation of S_{FRR} assumes high burst loss rate. It may not be accurate when burst loss rate is very low. Testing of the proposed model under low burst loss rate and modeling throughput of TCP variants using SACK-based recovery in RFC6675 would be addressed in our future work.

Acknowledgements

This work is sponsored by the National Natural Science Foundation of China under grant No. 61171089, the Training Program of the Major Research Plan of the National Natural Science Foundation of China under grant No. 91438104, Pre-research Project of Civil Aircraft Key Technologies under grant MIIT No. Equipment [2010]307, and the 2013 Innovative Team Construction Project of Chongqing Universities.

References

- [1] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "RFC 2018: TCP selective acknowledgement options", (1996).
- [2] H. X. Nguyen and M. Roughan, "on the correlation of internet packet losses", Telecommunication Networks and Applications Conference, Australasian, Adelaide, SA, ATNAC, (2008), pp. 22-27.
- [3] N. Ekiz, A. H. Rahman and P. D. Amer, "Misbehaviors in TCP SACK generation", ACM SIGCOMM Computer Communication. Rev, vol. 41, no. 2, (2011), pp.16-23.
- [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] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", IEEE/ACM Trans. Networking, vol. 8, no. 2, (2000), pp. 133-145.
- [6] V. P. Rao, M. P. Tahiliani and U. K. K. Shenoy, "Analysis of sfqCoDel for Active Queue Management", Proc. Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), (2014), pp. 262-267.
- [7] N. Parvez, A. Mahanti and C. Williamson, "An analytic throughput model for TCP NewReno", IEEE/ACM Trans. Netw., vol. 18, no. 2 (2010), pp. 448-461.
- [8] J. Dyck, C. Gutwin and D. Makaroff, "Adaptive forward error correction for real-time groupware", Proc. the 17th ACM international conference on Supporting group work, ACM, (2012), pp. 121-130.
- [9] W. Xu, Y. Xu, X. Wu and K. Ou, "Modeling TCP SACK steady state performance in lossy networks", Proc. IEEE ICOIN, (2011), pp.278-283.
- [10] T. L. Sheu and L. W. Wu, "An analytical model of fast retransmission and recovery in TCP-SACK", Perform. Eval., vol. 64, no. 6, (2007), pp. 524-546.
- [11] 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.
- [12] M. A. Mehmood, N. Sarrar, S. Uhlig and A. Feldmann, "Understanding flow performance in the wild", Proceedings of Global Communications Conference (GLOBECOM), IEEE, (2013), pp. 1410-1415.
- [13] M. A. Mehmood, N. Sarrar, S. Uhlig and A. Feldmann, "Impact of access bandwidth on packet loss: A flow-level analysis", Proceedings of Communications (MICC), IEEE Malaysia International Conference, (2013), pp. 259-264.
- [14] . Yu, J. W. Modestino and X. Tian, "The accuracy of Gilbert models in predicting packet-loss statistics for a single-multiplexer network model", Proceedings INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, IEEE, (2005), pp. 2602-2612.
- [15] Keynote, Internet health report, Available from: <http://www.internetpulse.net/>.
- [16] B. Kim, D. Kim and J. Lee, "Lost Retransmission Detection for TCP SACK", IEEE communications letters, vol. 8, no. 9, (2004), pp. 600-602.

Authors



Zhiming Wang, he was born in 1986. He received the B.E degree in Electronic Information Engineering from the Chongqing University, Chongqing, China, in 2009. He is a Ph.D. degree candidate with the College of Communication Engineering, Chongqing University. His research interests include TCP modeling, congestion control, and aeronautical Ad hoc networks.