ISSN:0975-9646
Komal Tahiliani et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1315-1317

A Survey on Performance Analysis of TCP Variants in IEEE 802.11 Based Ad-Hoc Networks

Komal Tahiliani#, Anju singh*, Dr R.C.jain#
#*Computer science department, Barkatulla University Bhopal,India*
**SATI, Vidisha,India*

***Abstract*— In order to allow programmers to follow conventional techniques while creating applications that uses an internet, software in the internet must provide the same semantics as a conventional computer system, it means it must guarantee reliable communication. Transport protocol provides reliability, which is fundamental for all the applications. The transmission Control protocol (TCP) is the transport level protocol that provides a completely reliable connection –oriented, full duplex stream transport service that allows two application programs to form a connection, send data in either direction, and then terminate the connection. This paper reviews existing work done in this area**

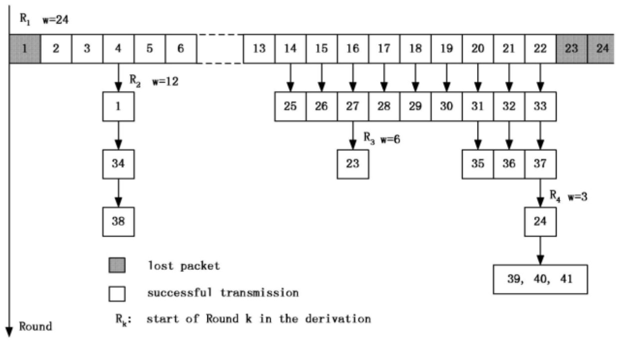
***Keywords*—** congestion, interference Range, flow control, error recovery.

I. INTRODUCTION

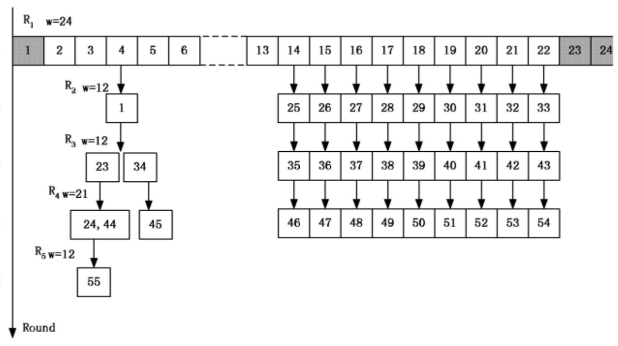
It is noted that TCP uses packet loss as an indicator of congestion. In wired networks, this works well because the transmission channel is so reliable and the topology is stable that the only statistically relevant cause of loss is congestion[2]. However, in mobile networks packets can be lost by a number of different causes, such as transmission errors, link failures, and topology changes, for which TCP's response of reducing its transmission rate is inappropriate. The result is less than ideal performance Fundamental Principles of TCP the Transmission Control Protocol (TCP) is a widely used transport protocol in wired and wireless communications, layered on top of IP networks to provide reliable end-to-end congestion control. TCP sends data in segments which do not exceed a maximum segment size as negotiated via a three-way handshake between the communicating agents during an initial connection establishment phase. Each byte (octet) of data has a sequence number assigned to it. When the receiver receives a segment, it notes the bytes of data (or sequence number range) of the segment and responds by sending back a cumulative acknowledgement (ACK) which confirms that all bytes up to the given sequence number have successfully arrived. The TCP sender also maintains a retransmission timeout (RTO) timer, which on expiration indicates that a segment has been lost and is to be retransmitted. The functionality offered by cumulative ACKs, the RTO timer as well as a checksum on the segment header and data ensures reliability on top of IP.[4,5] Another important functionality of TCP is flow and congestion control through the use of Recent traffic monitoring over the Internet has confirmed the popularity of the Reno and New Reno TCP variants as well as the increasing adoption of the TCP selective acknowledgements (SACK) modification.[20].

II. EXISTING SYSTEM

We consider TCP in terms of rounds where a round starts when the sender begins the transmission of a window of packets and ends when the sender receives an ACK for one or more of these packets. the several rounds of fast recovery process for Reno and New Reno are best explained via the examples shown in the Fig Suppose that Packets 1, 23 and 24 are lost When the window reaches w = 24 and that packets 2 through 22 are successful . this window is called the loss window[2] . In this case, Reno and New Reno have the same behavior in the second round.[6,7] The source first receives 21 duplicate ACK's , each with a sequence number requesting Packet 1. The first three ACK's trigger the fast retransmit of Packet 1 and cause the window to drop to 12. Then, this window is temporarily inflated by the number of duplicate ACK's . Once the duplicate ACK triggered by Packet 13 is received, the window is inflated to 12 + 12 = 24. The number of Packets in the pipe known by the source is still 24, since each ACK carries the sequence number requesting for packet 1. [4,5] During this period , transmission of new packets is not permitted packets beyond the allowed window . for every subsequent duplicate ACK, TCP continues to inflate its window and transmits one new packet , up transmitting nine new packets , with the largest sequence number being 24+9 =33[12].



(a)



(b)

RENO

In Reno as shown in the figure (a), This inflation is removed, and the window is cut back to 12 when the sequence number carried by ACK advances, that is , when the ACK for the retransmission comes back with a sequence number requesting Packet 23 . At that point only one new packet 34, since the outstanding packet is 33 – 22 = 11. In the third round, the lost packet 23 is retransmitted with the arrival of four duplicate ACK's requesting packets 23 [7]. The window is further decreased in half and followed by four new packet transmissions due to the window inflation. Following this new packet can be sent. The fifth round starts with the exit of the first recovery process when packet 24 is successfully retransmitted.[22].

NEW RENO

In New Reno, as shown in Fig. 1b, with the successful retransmission of packet 1, one partial ACK requesting packet 23 arrives at the TCP source. Packet23 is immediately retransmitted without waiting for enough duplicate ACKs [5]. The congestion window is deflated by the amount of new data acknowledged minus one segment and is 33 _ 22 þ 1 ¼ 12. One more packet 34 is allowed to be transmitted, since the outstanding packet is 11. For each additional duplicate ACK received, the congestion window is incremented, which allows more new packets to be sent, as shown in the figure. Following this logic, the congestion window is artificially inflated to 12 þ 9 ¼ 21when an ACK requesting packet 24 arrives. It is not deflated, since only one new packet is acknowledged. With the successful retransmission of packet 24, the window is cut back to 12 again[6,7].

SACK

TCP with selective acknowledgements is an extension of TCP RENO and it works around the problem faced by TCP RENO and TCP NEW RENO namely detection of multiple loss packets .SACK retains the slow start and Fast Retransmits part of RENO.SACK TCP requires that segments not to be acknowledged cumulatively but to be acknowledged selectively. If there are no such segments outstanding then it sends a new packet. Thus more than one lost segment can be sent in one RTT. The biggest problem with SACK is that currently selective acknowledgements are not provided by the receiver to implement SACK which is very difficult task.

TAHOE

Tahoe refers to TCP congestion control protocol which was discussed by van Jacobson in his paper [23]. TCP is based On principle of " conservation of packets". TCP implements this principle by using the acknowledgements to Clock outgoing packets because an acknowledgement means that packet was taken of the wire by the receiver. It also maintains a congestion window CWD to reflect the network capacity. Tahoe suggests the whenever a TCP Connection starts or Restarts after a packet loss, it should go through a procedure called slow start. The reason for the procedure is that the initial burst might overwhelm the network and connection might never get started. The congestion window size gets double for each transmission until it counters congestion. Slow start suggests that sender should set the CWD at 1 and for each ACK received it should increase the window by 1.so in first RTT we send 1 packet ,in second RTT we send 2 packets and in third we send 4 packets. It goes on increasing until we encounter a loss packet which is a sign of congestion. When we encounter congestion than we decrease the sending rate and reduces the window to 1.Then we will start over again. The important thing is that Tahoe detects packet losses by timeouts.

III. CONCLUSION

Reliable data delivery is the most important aspect of any network , wired networks uses TCP protocol for the purpose , but when we talk about wireless network the performance of TCP degrades due to the unstable and mobile nature of the network .We have analyze the performance of TCP Reno ,New Reno, Tahoe and SACK. we hope this Analysis will be of some use in future work in this area while meeting the needs for today's high demanding world.

REFERENCES

[1] A.A. Abouzeid, S. Roy, and M. Azizoglu, "Stochastic Modeling of TCP over Lossy Links," Proc. IEEE INFOCOM, pp. 1724-1733, 2000.

[2] M. Allman, V. Paxson, and W. Stevens, TCP Congestion Control, IETF RFC 2581, 1999.

[3] K. Chen, Y. Xue, S.H. Shah, and K. Nahrstedt, "Understand Bandwidth-Delay Product in Mobile Ad Hoc Networks," Elsevier Computer Comm. J., special issue on protocol engineering for wired and wireless networks, vol. 27, no. 10, pp. 923-934, 2004.

[4] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks," J. Wireless Comm. and Mobile Computing, vol. 2, no. 1, pp. 85-100, 2002.

[5] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput andLoss," Proc. IEEE INFOCOM, vol. 3, pp. 1744-1753, 2003.

[6] S. Floyd, T. Henderson, and A. Gurtov, The NewReno Modification to TCP's Fast Recovery Algorithm, IETF RFC 3782, 2004.

[7] S. Floyd and T. Henderson, The NewReno Modification to TCP's Fast Recovery Algorithm, IETF RFC 2582, 1999.

[8] V. Jacobson, Modified TCP Congestion Avoidance Algorithm, message to end2end-interest mailing list, ftp://ftp.ee.lbl.gov/email/vanj. 90apr30.txt., Apr. 1990.

[9] A. Kumar, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link," IEEE/ACM Trans. Networking, vol. 6, pp. 485-498, 1998.

[10] A. Kumar and J. Holtzman, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Mobile Radio Link," Sadhana: Indian Academy of Sciences Proc. in Eng. Sciences, vol. 23, pp. 113-129, 1998.

[11] B. Kim and J. Lee, "A Simple Model for TCP Loss Recovery Performance over Wireless Networks,". Comm. and Networks, vol. 6, no. 3, pp. 235-244, 2004.

[12] X. Li, P.Y. Kong, and K.C. Chua, "Analysis of TCP Throughput in IEEE-802.11-Based Multi-Hop Ad Hoc Networks," Proc. 14th IEEE Int'l Conf. Computer Comm. and Networks (ICCCN '05), pp. 297-302,2005.

[13] J. Li, C. Blake, D.S.J. Couto, H.I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," Proc. ACM MobiCom, pp. 61-69, 2001.

[14] T. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," IEEE/ACM Trans. Networking, vol. 5, no. 3, pp. 336-350, 1997.

[15] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," ACMComputer Comm. Rev., vol. 27, no. 3, pp. 67-82, July 1997

[16] V. Misra, W. Gong, and D. Towsley, "Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior," Proc. Int'l Federation for Information Processing WG 7.3 Conf. (Performance '99), 1999.

[17] A. Medina, M. Allman, and S. Floyd, “Measuring the Evolution of Transport Protocols in the Internet,” Computer Comm. Rev., vol. 35, no. 2, 2005.

[18] J. Padhye, V. Firoiu, D.F. Towsley, and J.F. Kurose, “Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation,” ACM Computer Comm. Rev., vol. 28, pp. 303-314, 1998.

[19] Scalable Mobile Network Simulator, http://pcl.cs.ucla.edu/projects/glomosim/, Nov. 2006.

[20] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, IETF RFC 2001, 1997.

[21] S. Xu and T. Saadawi, “Revealing the Problems with 802.11 Medium Access Control Protocol in Multi-Hop Wireless Ad Hoc Networks,” Computer Networks, vol. 38, no. 4, pp. 531-548, 2002.

[22] S. Xu and T. Saadwi, “Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireles ADHoc Networks?” IEEE Comm. Magazine, vol. 39, no. 6, pp. 130-137, 2001.

[23] Laxmi subedi, Muhammadreza Naiji minaini, ”Performance evaluation of TCP Tahoe and Reno with openet Modeler”june 2008.

[24] A. Gurtov and S. Floyd, “Modeling wireless links for transport protocols,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 85–96, Apr. 2004.

[25] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *Proc. ACM/IEEE Int. Conf. on Mobile Computing*, Seattle, WA, USA, Sept. 1999, pp. 219–230.

[26] Y. Shang and M. Hadjitheodosiou, “TCP splitting protocol for broadband and aeronautical satellite network,” in *Proc.23rd IEEE Digital Avionics Syst. Conf.*, Salt Lake City, UT, Oct. 2004, vol. 2, pp. 11.C.3-1–11.C.3-9.

[27] J. Zhu, S. Roy, and J. H. Kim, “Performance modeling of TCP enhancements in terrestrial-satellite hybrid networks,”*IEEE/ACM Trans. Netw.*, vol. 14, no. 4, pp. 753–766, Aug. 2006.

[28] C.-H. Ng, J. Chow, and Lj. Trajkovic, “Performance evaluation of the TCP over WLAN 802.11 with the Snoop performance enhancing proxy,” OPNETWORK 2002, Washington, DC, Aug. 2002.