ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (74-83), Month: July - September 2014, Available at: www.researchpublish.com

A Literature Survey on Packet control for wireless network

Sonia[1], Rupali Malhotra[2]

[1]Maharashi Dayanand University, Satpriya Group of Engineering & Technology
Rohtak, Haryana (India)
[2]Asst.Prof, Satpriya Group of Engineering & Technology
Rohtak, Haryana (India)

Abstract: **In this paper, I propose packet control algorithms to be deployed in intermediate network routers. They improve TCP performance in wireless networks with packet delay variations and long sudden packet delays. The ns-2 simulation results show that the proposed algorithms reduce the adverse effect of spurious fast retransmits and timeouts and greatly improve the goodput compared to the performance of TCP Reno. The TCP goodput was improved by ~30% in wireless networks with 1% packet loss. TCP performance was also improved in cases of long sudden delays. These improvements highly depend on the wireless link characteristics.**

***Keywords**:* **TCP, packet control, wireless networks, packet delay, packet delay variation.**

1. INTRODUCTION

The performance of Transmission Control Protocol (TCP) has greatly improved since 1988, when the congestion avoidance and control algorithms were first introduced. TCP is currently the most widely used Internet transport protocol. In 2002, TCP traffic accounted for 95% of the IP network traffic. This was due to a variety of popular Internet applications and protocols. Web (HTTP), file transfer (FTP), and e-mail (SMTP) rely on TCP as the underlying transport protocol. Internet applications that rely on TCP today are likely to do so in the future. With a growing deployment of wireless networks, it is important to support these applications in both wireline and wireless environments. Hence, wireless networks will also require good TCP performance.

Wireless networks have different characteristics compared to wireline networks. TCP, which was carefully designed and tuned to perform well in wireline networks, suffers performance degradation when deployed in wireless networks.

2. TRANSMISSION CONTROL PROTOCOL

TCP is a connection-oriented transport layer protocol. It provides reliable byte stream services for data applications. Its key features include reliability, flow control, connection management, and congestion control. Major TCP versions are Tahoe, Reno, and New Reno. They differ mainly in their congestion control algorithms. Tahoe, the original version of TCP, employs three congestion control algorithms: slow start, congestion avoidance, and fast retransmit. TCP Reno extends Tahoe with a fast recovery mechanism. New Reno, the latest major version of TCP, modifies TCP Reno's fast recovery algorithm and addresses the issue of partial acknowledgements (ACKs).

Differences between the characteristics of wireline and wireless networks have significant impact on TCP performance. TCP was designed and optimized to perform well in wireline networks. Wireless links, with considerable packet losses due to link errors, delay variations, and long sudden delays, violate TCP's essential design assumptions. Improving TCP performance in wireless networks has been an ongoing research activity since the mid 90's. Most improvements dealt with TCP's reaction to high bit error rate (BER) and TCP performance degradation due to delay and delay variation in wireless links. Performance of TCP's congestion control algorithms particularly deteriorates when TCP is deployed in mixed wireline/wireless networks. We describe here TCP's timer and window management, congestion control algorithms, and round-trip time (RTT) estimation.

3. PROPOSED TCP WITH PACKET CONTROL

We propose a set of packet control algorithms designed to avoid the adverse effect of long delays and delay variations on TCP performance in wireless networks. We describe the algorithms, their implementation, and evaluate their performance using the ns-2 simulator.

A. Network Architecture

Network architecture, shown in Fig. 1, represents a cellular network or a wireless LAN (WLAN). A mobile host (MH) initiates a TCP connection with a fixed host (FH) through a base station (BS), which is an edge node in the wireless network. TCP packets are sent from the FH to the MH through the BS and MH acknowledges every data packet received. TCP data may be either a long lived FTP connection with a large volume of data traffic or a short lived HTTP connection with a typically smaller volume of data traffic. We assume that the condition of the wireless link may change with time (leading to variable wireless link delay), that the mobile device roams between cells, and that mobile applications have limited data bandwidth.

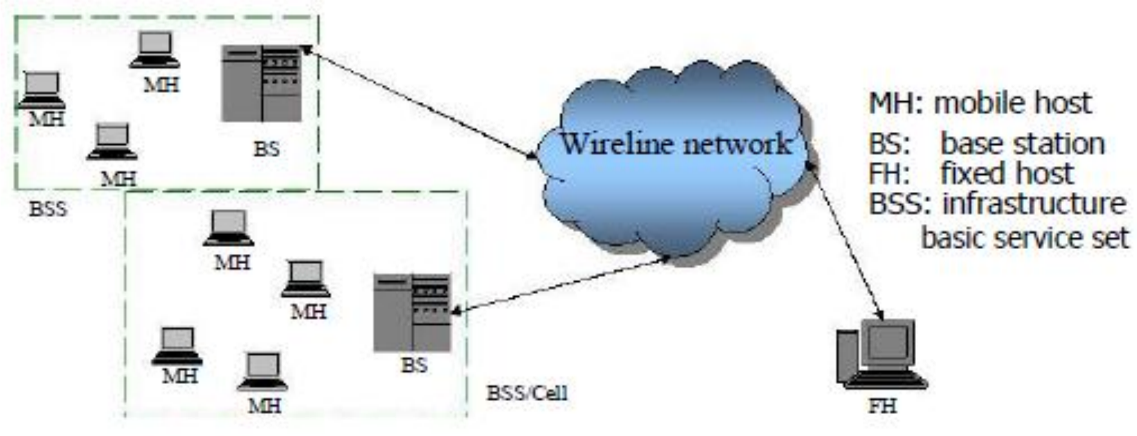

Figure1. Network architecture

B. TCP with Packet Control

TCP with packet control consists of ACK and data packet filters. The two filters improve TCP performance in mixed wireline/wireless networks and maintain TCP's end-to-end semantics. They deal with wireless links with long sudden delays and delay variations, handle handoffs, and maintain regular TCP functions. They do not depend on end-user TCP flavors.

The filters are to be deployed at the wireless network edge (typically the BS). This is a TCP-aware link layer solution. The algorithms keep track of TCP data and ACK packets received from the FH and the MH, respectively. Packet control filters forward packets to both client ends based on the information gathered in the BS.

1. ACK Filter: Packet control reacts to ACKs received from the MH using the ACK filter. It drops the old ACKs and duplicate ACKs classified according to the duplicate ACK threshold defined by the user. It remembers the last new ACK received from the wireless receiver, called the last received ACK. When an ACK arrives, its ACK number is checked against the last received ACK. We consider three cases:

Old ACK: The ACK is considered old if the ACK number has already been received and/or is smaller than the last received ACK. It is immediately dropped.

Duplicate ACK: If the newly received ACK number is identical to the largest ACK currently received, it is considered to be a duplicate. Packet control keeps track of the current number of duplicate ACKs received at the BS. Based on the number of duplicate ACKs received and the user-defined duplicate ACK threshold, duplicate ACKs are evenly dropped and are not sent to the sender. The number of ACKs to be dropped is equal to the difference between the user-defined duplicate ACK thresholds at the BS and at the FH. For example, if the user-defined duplicate ACK threshold is 6 and TCP has defined the three duplicate ACK threshold, every second duplicate ACK is dropped.

New ACK: If the ACK number has not been previously received, the ACK is considered new. The last wireless ACK is updated, the counter for the current number of duplicate ACKs is reset, and the ACK is forwarded to the sender.

The design of the ACK filter is based on the observation that a wireless link has a high number of re-ordered segments, which is the primary cause of spurious fast retransmit. By filtering some duplicate ACKs at the BS, the spurious fast retransmit may be reduced. If there is no packet loss in the network, filtering duplicate ACKs results in better TCP performance.

2. Data Filter: When the packet control receives a data segment from the FH, it passes it to the MH. The data filter at the BS is designed to prevent the spurious fast retransmit caused by spurious timeout.

In the case of spurious timeout, retransmissions of the unacknowledged segments unnecessarily consume the scarce wireless link bandwidth and also trigger additional spurious fast retransmits. Therefore, their prevention is essential in solving spurious timeout. The data filter checks whether data segments have been acknowledged. The sequence number is checked against the last ACK received from the receiver. We consider two cases:

New data segment or unacknowledged segment: If the segment has not been acknowledged, it is forwarded to the receiver. The segment is either a new data segment or an unacknowledged segment. In the latter case, the system cannot distinguish whether the last transmission of the same segment has been received by the receiver or its ACK was lost. In both cases, even if the received segment is a retransmission, it should be forwarded.

Acknowledged segment: This segment is a retransmission due to spurious timeout. This occurs because the ACK from the BS is lost or has not arrived at the FH. In both cases, the segment should be dropped. We consider that a loss of ACKs could occur even though the BER and the possibility of congestion for ACKs are small in wireline networks. For every two identical retransmitted segments received, an ACK is sent from the BS to the sender. Hence, unnecessary retransmissions are eliminated and the problem of lost ACKs is resolved.

C. Design Considerations and Tradeoffs

Packet control filters designed to deal with the wireless link delays have to be simple to implement.

TCP option: Packet control has been designed as an option for TCP rather than a modification of TCP. Hence, it is less difficult to deploy in an existing network.

A link layer solution in BS: Packet control requires modification in the BS only. No modifications are required at the end users. Furthermore, it can be deployed incrementally because it does not require changes in the protocol stack.

Scalable: With proper implementation, packet control filters only require retaining few constant state variables, and, hence, require minimal additional memory in the BS.

Handoff: Packet control does not require additional operations during handoffs, such as additional memory requirements or message exchanges, and will not adversely affect handoffs.

4. IMPLEMENTATION OF TCP WITH PACKET CONTROL

We implemented TCP packet control in the ns-2.26 simulator on Red Hat Linux 9.

Fig. 2 illustrates the logic flow of the ACK filter. The variable *numOfLastDupAck* indicates the number of duplicate ACKs that have been received for the last received wireless ACK. It is updated when a new ACK is received. It is then used, along with the user-defined duplicate ACK threshold (*redefine3DupAck*), to determine whether an ACK should be sent or dropped. The next duplicate ACK to be sent (*nextDupAckToSend*) is calculated as:

$$numOfDupAckSent = \frac{\frac{numOfLastDupAck - 1}{redefine3DupAck}}{3}$$

$$nextDupAckToSend = (numOfDupAckSent + 1) \cdot \frac{redefine3DupAck}{3}$$

Where, *numOfDupAckSent* is the number of duplicate ACKs that should be sent, triggered by the previous duplicate ACKs received. This ensures that duplicate ACKs will be evenly sent to the sender according to the user-defined duplicate ACK threshold in the BS.

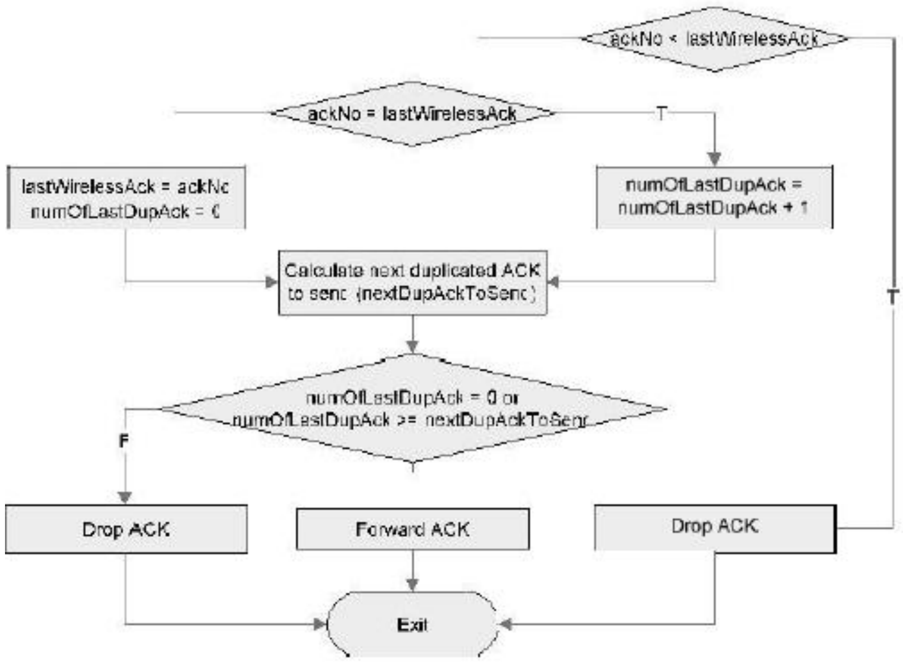

Fig. 2. Packet control: ACK filter.

Fig. 3. shows the logic flow of data filter. The variable *lstRetransWiredDataPkt* stores the segment numbers of retransmitted segments from the FH. A retransmitted segment is defined as a segment with the sequence number smaller or equal to the largest ACK number that has already been sent to the FH. These retransmitted segments are dropped. The number of retransmissions for each retransmitted segment *(m_iNumOfRtm*) is kept for each segment in the list. An ACK for a segment is generated and sent to the FH for every second retransmission of the same segment. This handles the rare situations when an ACK is lost on the path from the BS to the FH.

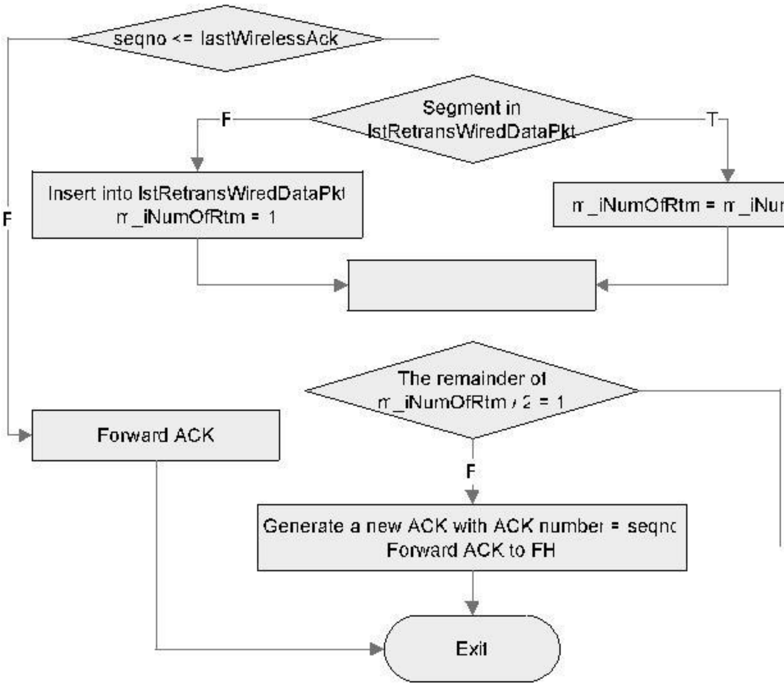

Fig. 3 Packet control: Data filter

5. PERFORMANCE OF TCP WITH PACKET CONTROL

The simulated network is shown in Fig. 4. A wired link connects the FH to the BS, while a wireless link connects the BS and the MH.

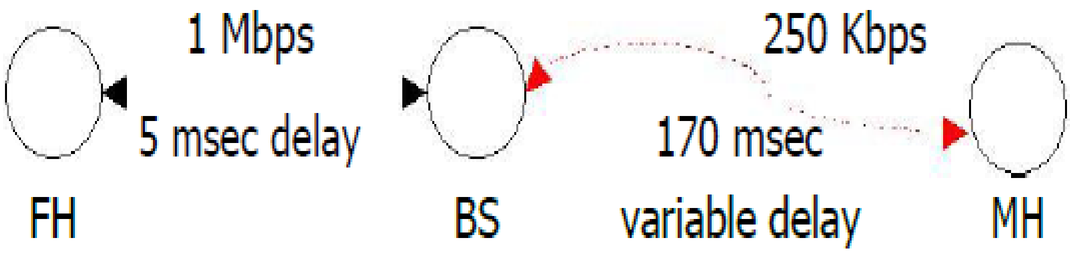

Fig. 4 Simulated network setup

A. Scenario I: Link Delay Variation with Small Delay

This scenario is used to investigate TCP's reaction to link delay variations. For 20 seconds, FTP data are being sent from the FH to the MH in TCP packets of 1,040 bytes (default in ns-2). Link delay variation is introduced at 0.5 s. Links employ DropTail queues. The simulation results show improvement in TCP performance. The number of *cwnd* reductions vs. time is shown in Fig. 8. Due to the spurious fast retransmit caused by link delay variation, TCP without packet control has the largest number of *cwnd* reductions, which also implies the largest number of fast retransmits. TCP with packet control and the user-defined duplicate ACK threshold set to 12 has the smallest number of *cwnd* reductions. The larger the duplicate ACK threshold, the more duplicate ACKs will be dropped and fewer fast retransmits will be performed by TCP. These fast retransmits are spurious and reducing them results in higher TCP performance. Graph with no packet control and graph with user-defined duplicate ACK threshold of three overlap, validating the implementation of the filters. Since TCP sender also has the threshold of three, no duplicate ACKs are dropped and the two cases coincide.

Variations of *cwnd* are shown in Fig. 6. Since larger duplicate ACK threshold in packet control results in fewer spurious fast retransmits, *cwnd* remains large. *Cwnd* is directly related to TCP's throughput. TCP's performance may also be examined by observing the goodput shown in Fig. 7. With an appropriate user-defined duplicate ACK threshold, TCP with packet control successfully reduces the number of spurious fast retransmits. It may improve TCP goodput by ~100%.

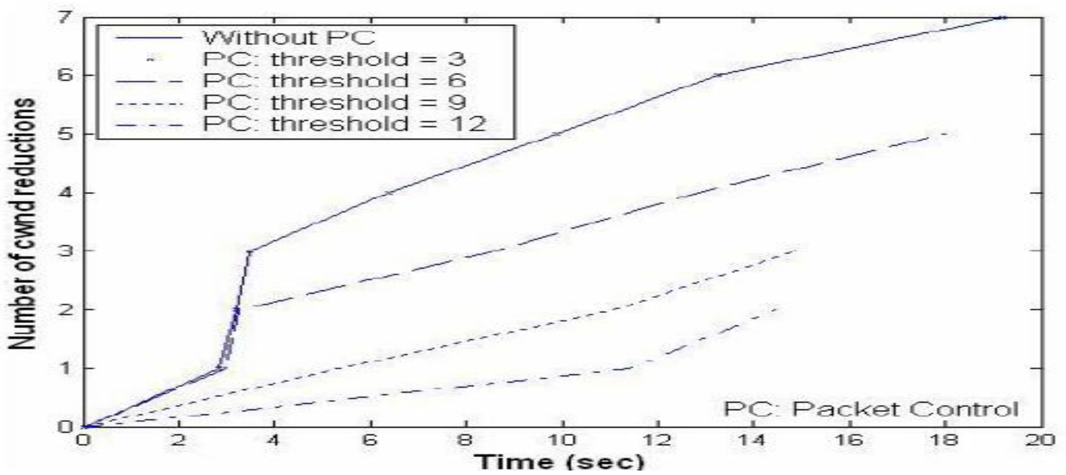

Fig. 5 Link delay variation: number of *cwnd* reductions

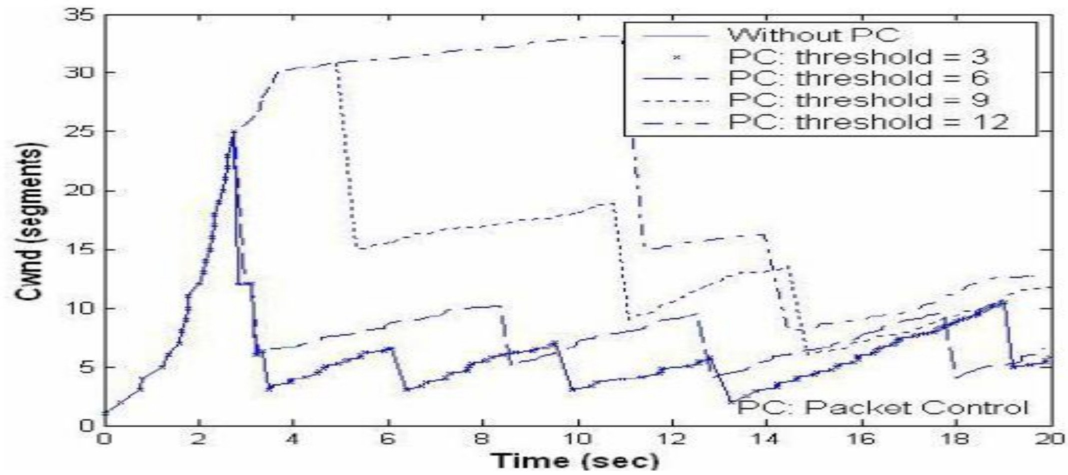

Fig. 6 Link delay variation: *cwnd*

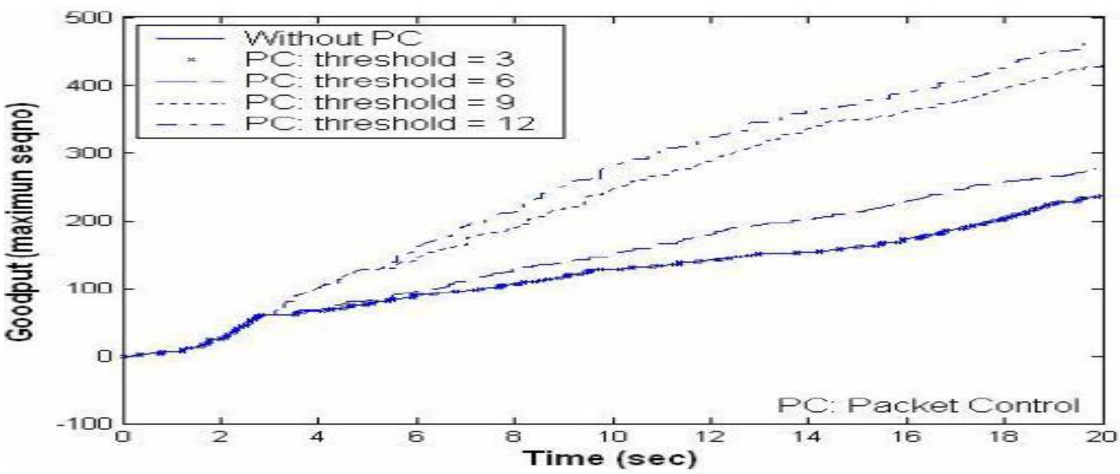

Fig. 7 Link delay variation: goodput

B. Scenario II: Link Delay Variation with Small Delay and Link Errors

This scenario investigates the case with 1% bidirectional packet loss in the wireless link. Link layer retransmission protocols in 3G networks, such as RLP in 3G1X and RLC in UMTS, ensure that packet loss probability is less than 1% on the wireless link. TCP goodput is shown in Fig. 8. With user-defined duplicate ACK threshold of 9, TCP with packet control achieves ~30% improvement.

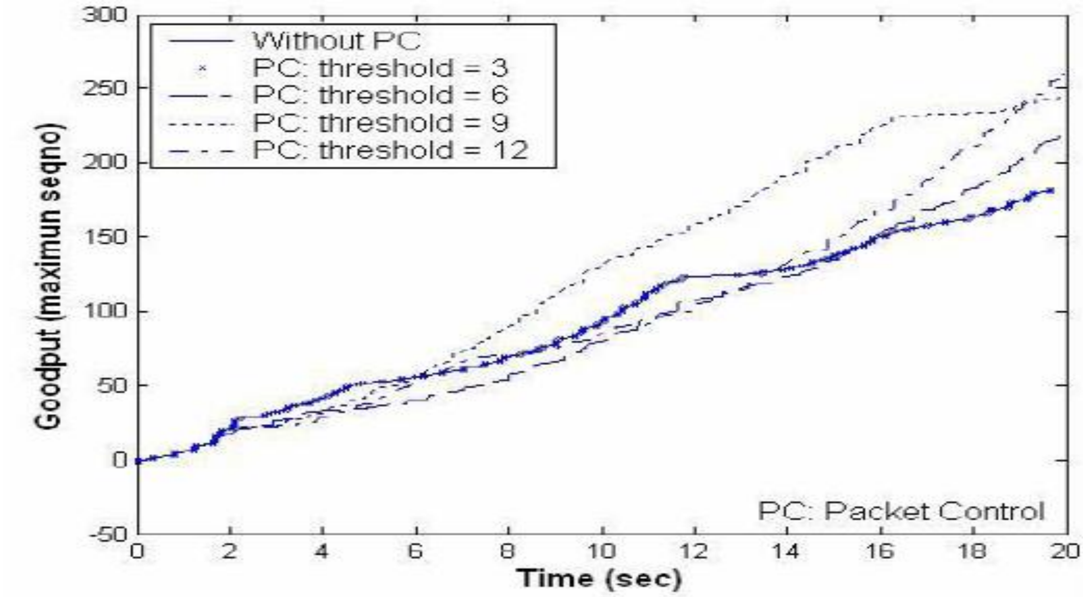

Fig. 8 Link delay variation (1% segment loss): goodput

C. Scenario III: Spurious Timeout

We also investigate TCP's reaction to sudden large delay increase. A delay of 6 s is introduced at 5 s in the wireless link. The reaction of TCP without packet control to the long sudden delay was shown in Fig. 3. Identical simulation scenario, with packet control enabled, is used to generate the results shown in Fig. 9. They illustrate that TCP recovers faster (indicated by the third arrow) than in the case shown in Fig. 3.

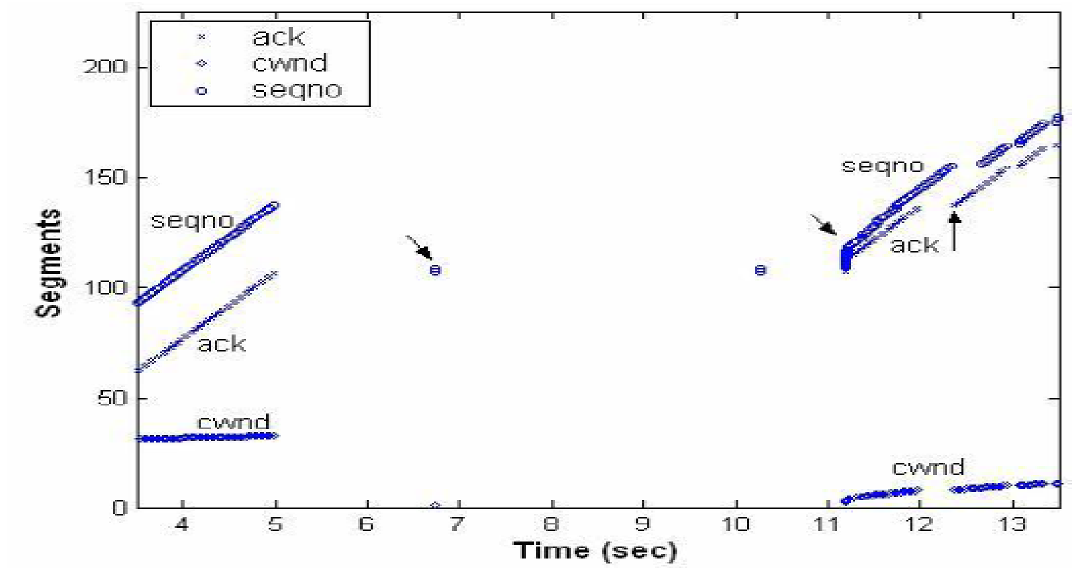

Fig. 9 TCP with packet control: spurious timeout

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (74-83), Month: July - September 2014, Available at: www.researchpublish.com

A comparison of TCP goodput is shown in Fig. 9. The improvement is highly dependent on the path characteristics.

D. Delay Generator

Delays in ns-2 simulations are generated by a delay generator. We implemented two types of delays: short delay variations and relatively long sudden delays.

Short delay variations are generated based on the measurements of packet delays in wireless data networks The configuration for evaluating TCP performance was based on CDMA 1xRTT network architecture. A mobile host in a wireless network was connected to a host PC in a wired local area network (LAN) through a single BS. The “ping” application was generated by the Internet control message protocol (ICMP) ECHO. Ping packets were sent from the PC in the LAN to the mobile terminal moving at pedestrian speed. The delays vary from 179 ms to 1 s in a 3G1X system and are in agreement with the “ping” latencies .

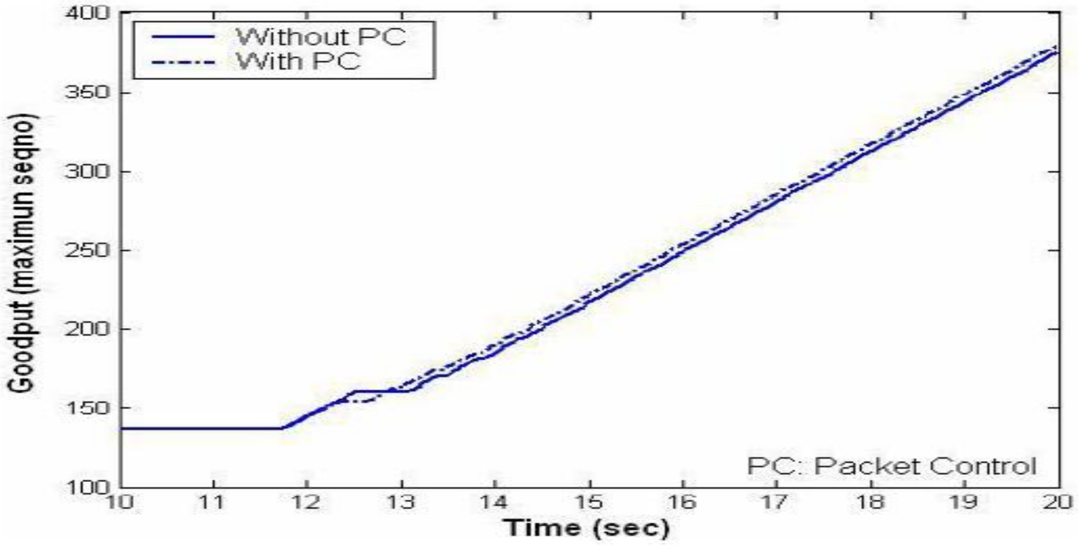

Fig. 10 TCP with packet control: goodput

Delay values used in simulations are shown in Table I. Each case is simulated with a uniform distribution generated by the ns-2 random generator. The wireline link delay was kept constant at 5 ms.

A long delay was generated by a timer. The one-way wireless link delay was kept constant at 170 ms. This value is close to the 300 ms average RTT. The sudden increase of delay was simulated for 6 s, which was sufficiently long to cause a regular TCP timeout with at least one exponential back-off.

TABLE I
WIRELESS DELAYS FOR MOBILE TERMINALS

RTT (%)	Total RTT (ms)	Wireline RTT (ms)	Wireless RTT (ms)	Wireless link delay (ms)
80	316 – 400	10	306 – 390	153 – 195
10	400 – 460	10	390 – 450	195 – 225
8	460 - 605	10	450 – 595	225 – 297
2	605 - 1252	10	595 - 1242	297 – 621

6. CONCLUSION

In this paper, we proposed packet control filters to improve TCP performance in wireless networks with delay variations and long sudden delays. TCP connections were simulated in a mixed wireline and wireless network using the ns-2 simulator. The simulation results show that the proposed algorithms reduce spurious fast retransmit and spurious timeouts in TCP. They improve TCP's throughput, good put, and bandwidth consumption. Good put of TCP Reno is improved by ~100% in networks with delay variations and by ~30% in networks with 1% packet losses in the wireless link. In cases of long sudden delays, TCP performance is also improved, depending on the path characteristics. Packet control filters can be conveniently deployed at the intermediate routers to control the transmission of TCP segments and ACKs. Future improvements may include more accurate delay generators and multi-connection simulation scenarios while using genuine wireless traffic traces for performance evaluations.

REFERENCES

[1] W. Stevens, TCP/IP Illustrated, Volume 1: The Protocols. Reading, MA: Addison-Wesley, Professional Computing Series, 1984.

[2] Information Sciences Institute, "Transmission control protocol," IETF RFC 793, Sept. 1981.

[3] V. Jacobson, "Congestion avoidance and control," in Proc. ACM SIGCOMM, Stanford, CA, Aug. 1988, pp. 314–329.

[4] I. Stojmenovic, Handbook of Wireless Networks and Mobile Computing. New York, NY: John Wiley & Sons, 2002.

[5] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," IETF RFC 2581, Apr. 1999.

[6] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," IETF RFC 2582, Apr. 1999.

[7] S. Floyd, "A report on recent developments in TCP congestion control," IEEE Communications Magazine, vol. 39, no. 4, pp. 84–90, Apr. 2001.

[8] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," IETF RFC 3481, Feb. 2003.

[9] A. Gurtov, "Effect of delays on TCP performance," in Proc. IFIP Personal Wireless Communications (PWC'01), Aug. 2001.

[10] A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in Proc. 15th International Conference on Distributed Computing Systems (ICDCS), Vancouver, BC, May 1995, pp. 136–143.

[11] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," IEEE Journal on Selected Areas in Communications, vol. 22, no. 4, pp. 747–756, May 2004.

[12] H. Balakrishan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP performance over wireless networks," in Proc. ACM MobiCom, Berkeley, CA, Nov. 1995, pp. 2–11.

[13] R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmission," ACM Computer Communications Review, vol. 30, no. 1, pp. 30–36, Jan. 2000.

[14] S. Fu, M. Atiquzzaman, and W. Ivancic, "Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment," in Proc. International Conference on Computer Communications and Networks, Miami, FL, Oct. 2002, pp. 575–578.

[15] K. Luo and A. O. Fapojuwo, "Impact of terminal mobility on TCP congestion control performance," in Proc. 3rd IASTED International Conference on Wireless and Optical Communications, Banff, Canada, July 2003, pp. 360–365.

[16] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," ACM SIGCOMM Computer Communication Review, vol. 27, no. 5, pp. 19–42, Oct. 1997.

[17] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby "Performance enhancing proxies," Internet Draft [Online]. Available: http://community.roxen.com/developers/idocs/drafts/draft-ietf-pilc-pep-04.html.

[18] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," Wireless Networks (WINET), vol. 8, no. 5, pp. 467–479, Sept. 2002.

[19] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bhargavan, "WTCP: a reliable transport protocol for wireless wide-area networks," in Proc. ACM MobiCom, Seattle, WA, Aug. 1999, pp. 231–241.

[20] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocol," ACM Transactions on Computer Systems, vol. 9, no. 4, pp. 364–373, Nov. 1991.

[21] ns–2 [Online]. Available: http://www.isi.edu/nsnam/ns.

[22] M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in Proc. ACM MobiCom, Atlanta, GA, Sept. 2002, pp. 71–82.