Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 5, May 2017, pg.361 – 372

TCP Window Based End-to-End Mechanisms

Dr. Elsadig Gamaleldeen Elsadig

Omdurman Ahlia University, Applied Computer and Physics College, Sudan

Abstract:

The paper discusses the TCP window base end to end mechanisms that can be used and the method of transmissions, different versions of TCP that had been implemented, the modifications that had been done in the options field in the TCP header to optimize it and the ACK generation at the receiver side. Also discuss the congestion window and finally the two of the Explicit Notification (ELN) Strategies.

Introduction

The end-to-end principle is a design framework in computer networking. In networks designed according to this principle, application-specific features reside in the communicating end nodes of the network, rather than in intermediary nodes, such as gateways and routers, that exist to establish the network. End-to-end mechanisms solve the wireless loss problems at the transport layers of the sender and receiver [1] and it does not need any support from the link layer performing retransmissions at the TCP sender.

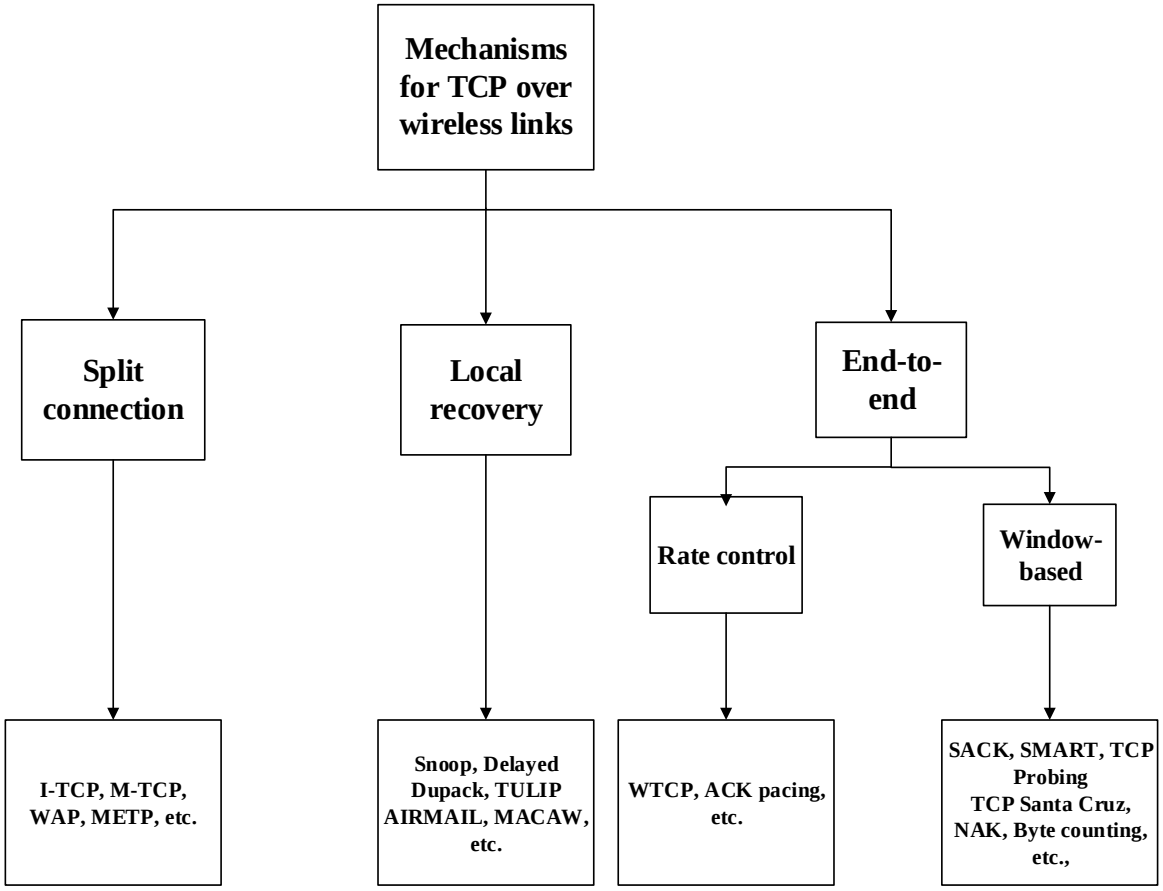

Fig 1 TCP over wireless links categorization mechanisms

1. Selective acknowledgments

TCP Selective Acknowledgments (SACK) [2] it is a window based that uses the options field of TCP to precisely inform the sender which segments were not received. This enables the sender to retransmit only those segments that are lost, thereby not to waste network bandwidth as explained in Fig 2.

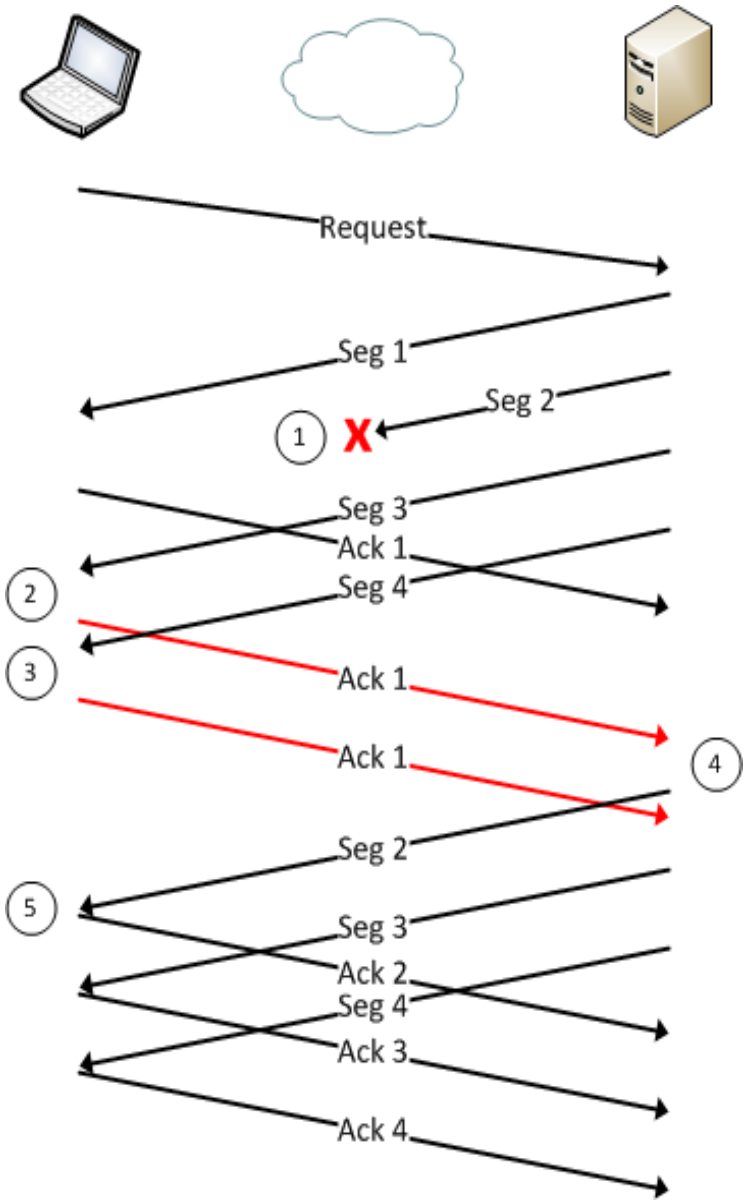

Fig 2 TCP selective acknowledgments mechanism

TCP SACK was proposed as a solution to recover from multiple losses in the same window without degrading the throughput of the connection [3]. The same idea had been applied to networks with wireless hops [4] to aid the sender in recovering from non-congestion related losses. TCP still performs congestion control upon detecting packet loss in the wireless link and there is no way to deduce where the loss occurred [5].

2. SMART Retransmissions

A Simple Method to Aid Retransmissions (SMART) [6] decouples flow and error control Fig 2.3. Each ACK packet from the receiver carries both the standard cumulative ACK [7] and the sequence number of the packet that initiated the ACK [8]. This informs the sender of the packet that is lost, so that the sender can selectively retransmit [2]. SMART uses a different exploratory method to detect lost retransmissions. SMART also avoids the dependence on timers for the most part except for the worst case when all packets and all ACKs are lost [5].

The scheme uses two different buffers and windows: an error-control window at the receiver for buffering the out-of-sequence packets, and a flow-control window at the sender for buffering unacknowledged packets. Thus, error control and flow control are decoupled [3].

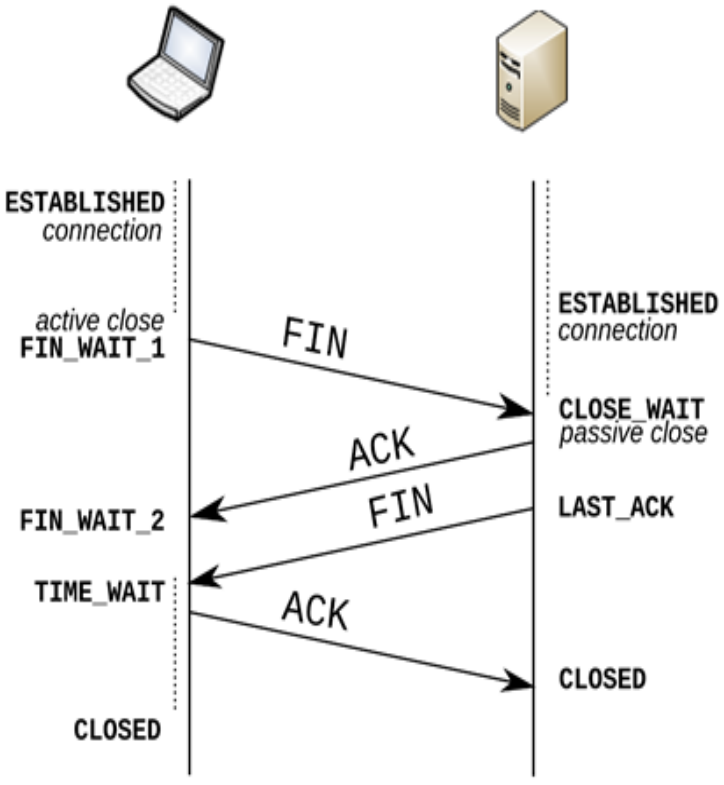

Fig 3 TCP SMART retransmissions

3. TCP probing

With TCP probing [8], when a data segment is delayed or lost, the sender, instead of retransmitting and reducing the congestion window size enters a probe cycle [3]. A probe cycle entails exchanging probe segments between the sender and receiver to monitor the network. The probes are TCP segments with header options and no payload. This helps reduce congestion because the probe segments are small compared to the retransmitted segments [9]. The cycle terminates when the sender can make two successive RTT measurements with the aid of receiver probes. In case of persistent errors [10], TCP decreases its congestion window and threshold. But for transient random errors [11], the sender resumes transmission at the same window size it used before entering the probe cycle.

4. TCP Santa Cruz

TCP Santa Cruz [4], like TCP probing, makes use of the options field in the TCP header. The congestion control algorithm is based on relative delays that packets experience with respect to one another in the forward direction and it was first introduced in TCP Vegas [9]. Therefore, ACK losses and reverse path congestion do not affect the throughput of a connection [2]. The scheme also improves the error recovery mechanism by making better RTT estimates than other TCP implementations. Retransmissions are included in making RTT estimates to take into consideration the RTT during congestion periods. TCP Santa Cruz uses Selective Acknowledgments (SACK) [10].

5. Negative acknowledgments

Negative ACKs (NAKs) can be included in the options field of the TCP header to explicitly indicate which packet has been received in error so that retransmission of that packet can be initiated quickly [11]. This is under the assumption that a

corrupted packet can still reach the destination and that the source address of the packet is still known [12]. The sender, on receiving a NAK, can retransmit the packet without modifying the congestion window size. RTT measurement from the retransmitted packets is ignored to avoid affecting the RTT estimate [13].

6. Wireless TCP (WTCP99)

Wireless TCP (WTCP99) [12, 13] uses rate-based transmission control not a self-clocking window-based scheme like TCP. The transmission rate is determined by the receiver using a ratio of the average inter-packet delay at the receiver to the average inter-packet delay at the sender [1]. The sender transmits its current inter-packet delay with each data packet. The receiver updates the transmission rates at regular intervals based on the information in the packets [7], and conveys this information to the sender in the ACKs.

WTCP99 computes an appropriate initial transmission rate for a connection based on a packet-pair approach [9]. This is useful for short-lived connections in Wireless Wide Area Networks (WWANs) where Round Trips are long. WTCP99 achieves reliability by using selective ACKs [9]. No retransmission timeouts are triggered as it is difficult to maintain good Round Trip Time estimates [12]. Instead, the sender goes into "blackout mode" when ACKs do not arrive at sender-specified intervals. In this mode, the sender uses probes to obtain ACKs from the receiver, similar to TCP probing [13].

7. ACK pacing

ACK pacing [12] is a rate based approach to ACK generation at the receiver or "pace", data sent into the network over an entire Round-Trip Time, so that data is

not sent in a burst. ACK pacing results in rate controlled sender packets[14], and hence avoids burst traffic that can result in packet losses, delays, and lower throughput TCP Pacing [13], also known in literature as "spacing" or "packet spacing", is a technique consisting in spreading the transmission of TCP segments across the entire duration of the estimated Round Trip Time instead of having a burst at the reception of acknowledgements from the TCP receiver [12].A TCP sender places an entire window of segments on the network as soon as it receives acknowledgements from the receiver. Pacing, however, does not help distinguish between congestion losses and wireless losses [7].

8. TCP Westwood

TCP Westwood (TCPW) [5] estimates the effective bandwidth at the TCP sender, without any modifications to the TCP receiver. Bandwidth estimation is based on the returning ACKs. A duplicate ACK (DUPACK) updates the Bandwidth Estimate (BWE) since it notifies the sender that a data packet was received at the destination, even though the packet was out-of-order [2,11]. When n DUPACKs are received [26]; the slow start threshold is updated as follows:

$$\text{ssthresh} = \frac{\text{BWE} \times \text{RTTmin}}{\text{ss}}$$

- ssthresh is a Slow Start Threshold
- RTT min is the Minimum Measured RTT
- SS is the TCP Segment Size.

The congestion window is set to ssthresh if it currently exceeds the new ssthresh value. Thus, the connection speed remains close to network capacity if BWE is high [11]. This approach is called "faster recovery." In the case of timer expiration,

ssthresh is set to minimum and the congestion window is set to 1 as in TCP Reno [14].

The slow start and congestion avoidance phases are also similar to TCP Reno, and therefore TCPW is "TCP-friendly" [13]. Fairness among users cannot be guaranty if bottlenecks exist on the backward links, since DUPACKs of certain connections are dropped. TCPW is unsuitable for networks with load that is highly dynamic [15], since measurements may not reflect the current network state.

9. pTCP

pTCP [16] was specifically designed for mobile hosts with multiple heterogeneous interfaces. The basic assumption in this work is that a mobile host wishes to use its interfaces simultaneously for a single application connection [6]. pTCP stripes data across these interfaces in order to achieve bandwidth aggregation at the receiver [3]. Both application-layer and link-layer data striping approaches are demonstrated to be less than transport layer approaches due to reasons such as (1) different bandwidth and data rates, (2) increased application complexity, and (3) presence of multiple congestion control schemes that maybe used at the different interfaces. The basic philosophy of pTCP mandates decoupling reliability and congestion control, striping data based on the congestion window of the individual pipes, reassigning windows dynamically, and supporting different congestion control mechanisms [15]. Mobility and handoffs may reduce the effectiveness of this approach because some pipes may be stalled during handoffs [7]. pTCP may exhaust host resources due to maintaining significant state. It also has the disadvantage of not being TCP-friendly.

10. TCP-Peach

TCP-Peach [3] is primarily targeted at satellite networks and avoids problems such as the long duration of the slow start phase caused by the long satellite propagation delays, the low rate imposed on the sender due to the long RTT, and the high error rate of the wireless environment [11]. TCP-Peach uses a sudden start mechanism instead of slow start. In sudden start, a number of dummy segments are sent during each RTT (in addition to data segments) and the congestion window grows with returning ACKs for these dummy segments as well as data [8]. These dummy segments have lower priority than data and therefore intermediate routers can drop them in times of congestion. TCP-Peach also proposes a rapid recovery algorithm for handling packet losses that are not due to congestion [14]. This algorithm is executed after the execution of the fast recovery algorithm when DUPACKs are received [4]. In the rapid recovery phase, a number of dummy segments are sent to probe the availability of network resources. According to the number of received ACKs for the dummy segments, the sender can determine whether the loss was due to congestion or channel errors [18].

A problem with TCP-Peach is that it requires awareness from the sender, receiver, and intermediate routers [17] (for priority drop of dummy segments). This limit it's deploy ability. Another point is that its approach for increasing the congestion window using dummy segments is aggressive, and thus not TCP friendly [4].

11. Explicit Bad State Notification (EBSN)

Losses on the wireless link of a connection may cause timeouts at the sender, as well as unnecessary retransmissions over the entire wired/wireless network. The EBSN scheme [2] proposes sending EBSN messages from the base station to the sender whenever the base station is unsuccessful in transmitting a packet over the

wireless network. This scheme is not end-to-end, but had been discuss here because it requires sender support like most end-to-end schemes [13].

EBSN receipt at the sender re-starts the TCP timer prevents the sender from decreasing its window when there is no congestion. Although this scheme requires modifications to the TCP implementation at the sender side [10], the required changes are minimal, no state maintenance is required and the clock granularity (Timeout Interval) has little impact on performance [11].

12. Explicit Loss Notification (ELN) Strategies

Like EBSN, ELN is not purely end-to-end. ELN has been proposed to recover from errors that occur when the wireless link is the first hop [7]. A base station is used to monitor TCP packets in either direction. When the receiver sends DUPACKs, the base station checks if it had received the packet that triggered by the DUPACKs [14]. If not, then the base station sets an ELN bit in the header of the ACK to inform the sender that the packet has been lost on the wireless link. The sender can then decouple retransmission from congestion control; it does reduce the congestion window. In contrast to Snoop [13], the base station need not cache any TCP segments in this case, because it does not perform retransmission.

References:

1. Craig Hunt, 2002. *TCP/IP Network Administration; O'Reilly Networking*. 3rd Edition. O'Reilly Media.
2. C. Villamizar and C. Song, 1994. *"High Performance TCP in Ansnet".* ACM Computer Communication Review, vol. 24, no. 5, pp. 45–60.

3. Brakmo, L. S. and L. L. Peterson, 1995. *"TCP Vegas: end-to-end congestion avoidance on a global Internet"*. IEEE Journal on Selected Areas in Communications vol. 13 no.8, pp.1465–1480.
4. Paganini, F., Z. Wang, S.H. Low and J.C. Doyle, 2003. *" A new TCP/AQM for stability and performance in fast networks"*. In Proceedings of IEEE Conference on Decision and Control (CDC), vol. 5 pp. 4684-4689. Paradise Island, Bahamas.
5. Monia Ghobadi, Yashar Ganjali, 2004. *Advances in Multimedia Information Processing - PCM 2004*: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30 - December 3, 2004, ... (Lecture Notes in Computer Science) (Pt. 2). 2005 Edition. Springer.
6. Kai Xu, Ye Tian, NirwanAnsari, 2007. *Human Interface and the Management of Information. Interacting in Information Environments*: Symposium on Human Interface, Held as Part of HCI ... Applications, incl. Internet/Web, and HCI). 2007 Edition. Springer.
7. Franklin F. Kuo, 1997. *Multimedia Communications: Protocols and Applications*. 1st Edition. Prentice Hall.
8. Ilyoung Chong, 2002. *Information Networking*: International Conference, ICOIN 2002, Cheju Island, Korea, And January 30-February 1, 2002: Revised Papers, Part 1: Wired Communications and Management (Pt. 1). 2002 Edition. Springer.
9. Mr. R. D. Mehta1, Dr. C. H. Vithalani, Dr. N. N. Jani, 2010. *"Enrichment of 'SACK' TCP performance by delaying fast recovery",* International Journal of Advanced Engineering Technology E-ISSN 0976-3945.
10. Rob Flickenger, 2008. *Wireless Networking in the Developing World:* 2nd Edition. 2nd Edition. Hacker Friendly Publishing.

11. *Mário Marques Freire*, 2000. *NETWORKING 2000. Broadband Communications, High Performance Networking, and Performance of Communication Networks: IFIP-TC6/European Commission. (Lecture Notes in Computer Science)*. 2000 Edition. Springer.
12. Constantinos Dovrolis, 2005. *"Passive and Active Network Measurement- New Methods for Passive Estimation of TCP Round-Trip Times"*: 6th International Workshop, PAM 2005, Boston, MA, USA.
13. Apostolis Salkintzis, 2005. *Emerging Wireless Multimedia: Services and Technologies*. 1st Edition. Wiley.
14. Mário Marques Freire, Prosper Chemouil, Pascal Lorenz and Annie Gravey, 2004. *"Universal Multiservice Networks"*: Third European Conference, ECUMN 2004, Porto, Portugal, October 25-27. 2004, Proceedings (Lecture Notes in Computer Science). 2004 Edition. Springer.
15. Russ White, 2014. *The Art of Network Architecture: Business-Driven Design (Networking Technology)*. 1st Edition. Cisco Press.
16. Azzedine Boukerche, 2005. *Handbook of Algorithms for Wireless Networking and Mobile Computing (Chapman & Hall/CRC Computer and Information Science Series)*. 1st Edition. Chapman and Hall/CRC.
17. Pei Zheng, 2009. *Wireless Networking Complete (Morgan Kaufmann Series in Networking).* 1st Edition. Morgan Kaufmann.
18. Michael Welzl, 2005. *Network Congestion Control: Managing Internet Traffic*. 1st Edition. Wiley.

