Wi-TCP: A TCP in Wireless Environment

Padma Bonde
Department. of Computer Science and Engineering
SSGI,Bhilai-INDIA

Bhavana Jharia
Department of Electronics and Telecommunication,
Ujjain Engineering College,
Ujjain -INDIA

A. K. Shrivastav
Directorate of Technical Education,Government of Madhya Pradesh,
Bhopal -INDIA

ABSTRACT

Due to the strong drive towards wireless Internet access through mobile terminals, it has become necessary to carefully handle the issues in the wireless environment such as frequent handover/handoff (user mobility), temporary disconnection, burst error and fading. These characteristics of wireless environment deteriorate the performance of wireless systems sharply in terms of TCP throughput as TCP is basically designed for wired networks to provide reliable delivery by using congestion control and error control mechanisms. In mobile IP based network, mobility plays an important role as handover issue makes negative impact on system performance and to enhance the behavior of TCP during handoff, a novel scheme is proposed. Wi-TCP is a new wireless end to end transmission control protocol, designed to support the TCP handoffs in mobile IP based network by utilizing the basic features of Mobile IP, Route Optimized Mobile IP and TCP.

KEYWORDS

Wi-TCP(Wireless Transmission Control Protocol),EHO-ACK(Exclusive Handover Acknowledgement),EHC-ACK(Exclusive Handover Completion Acknowledgement), HF_H(Home Agent to Foreign Agent Handover), FF_H (Foreign agent to Foreign Agent Handover), FH_H(Foreign Agent to Foreign Agent Handover)

1. INTRODUCTION

Most of the Internet applications such as file transfer, web browsing, and e-mail use TCP as a transport layer protocol which controls large fraction of traffic that travelling over Internet. Today mobile wireless access to Internet is experiencing a great deal of popularity, but TCP performs poorly in this environment because it treats any packet loss in the network as a result of network congestion and slows down its transmission rate, or even cause the TCP sender to experience series of unnecessary timeouts. As a consequence, it is important to study the TCP behavior in mobile wireless environment during handoff. However in the conventional Internet architecture mobile host can never inherently avoid the degradation of communication quality during handover. In view of the fact that the regular TCP such as TCP Tahoe [1], Reno[2] and New Reno [3] cannot perform well in mobile wireless networks; as regular TCP was not designed for keeping mobile node in consideration. If applied, it may encounter packet losses not related to congestion. Still, these losses will trigger congestion control procedures at the node. This will result in significant reductions in throughput and unacceptable interactive delays for active connections, thereby severely degrading the performance. Regular TCP cannot handle mobility and it detects losses via time-outs during longer handover latency and concludes that packet loss is due to congestion results in continuous erroneous retransmission. Furthermore even during short disconnection duplicate acknowledgements (ACK) may be generated that results in erroneous fast recovery and fast re-transmission mechanism which again degrades TCP behavior. This is a fundamental design problem of TCP in mobile wireless environment. Moreover, it has been a challenging issue to reduce packet loss during handover in mobile wireless networks. In view of the above fact, Wi-TCP mechanism recovers losses efficiently while entire or partially window has lost.

2. EXISTING SOLUTIONS

During the past few years many protocols have been proposed to improve the performance of TCP in mobile wireless networks. Study of different approaches on adaption of TCP in wireless networks revealed that some of these have worked to handle high Bit Error Rate (BER) in wireless channel. Several approaches are dealing with short interruptions of the connection either due to handoff in intra sub-networking or due to transmission error on wireless channel while few have been concentrated on inter sub-networking handoff. Further, on the basis of the method applied during implementation to achieve the goal, solutions can be categorized into the approaches like split-connection, explicit notification, end-to-end and link layer approach. Split connection approach describes how optimization at the transport layer can be achieved by splitting the connection at the base station. In the split connection approach, the main idea is to isolate mobility and wireless related problems from the existing network protocols. This is done by splitting the TCP connection between the mobile node and the fixed node into two separate connections: a wired connection between the fixed host and the base station and a wireless connection between the base station and the mobile host [4-7]. In this way, a wired connection is not required to have any changes in the existing software on the fixed hosts; and the wireless connection can use a mobile protocol specialized to provide better performance. In order to distinguish the congestion related losses from the wireless losses, explicit notifications can be used between sender and intermediate node, or receiver. Various solutions have been outlined towards the above said objective [8-14]. In some approaches signal strength usually is used as a performance measure of the wireless link condition. However practically it is very difficult for Mobile Node/Mobile Host (MN/MH) to properly detect communication quality, because signal strength fluctuates due to distance and interfering object.

The TCP sender attempts to handle the losses in a way that improves the performance over regular TCP by maintaining end-to-end semantics which is the most essential feature of TCP[15-20].The link layer approach tries to increase the quality of the lossy wireless link. Thus, it hides the characteristics of the wireless link from the transport layer and tries to solve the problem at the link layer. Link-layer optimizations for improved TCP performance are presented in [21].After reviewing the research studies, it is observed that in unpredictable mobile wireless environments, mobility itself can cause consequent packet loss that makes negative impact on TCP performance. There are certain issues related to user mobility such as handoff, handover latency, triangular routing in Mobile IPv4[22] and further re-transmission during handoff and reduction in congestion window at sender are to be redefined to enhance TCP behavior. Therefore an efficient and flexible management for handover loss recovery is to be required in designing TCP for mobile wireless network.

3. MECHANISM OF Wi-TCP

Wi-TCP comprises of two modules one is implemented on sender side while other on intermediate node. In order to implement a concept, cross layer feedback module and schemes (HF_H, FF_H and FH_H) are introduced at IEEE 802.11 WLAN base stations to initiate handoff notifications. While implementing cross layer feedback module, MAC layer self's information has been considered as a decisive factor to detect handoff. Further on the basis of feedback from MAC layer, EHO-ACK and EHC-ACK are initiated at home agent/foreign agent which enabling the TCP sender to exclusively identify the handoff. Further module has designed at sender which exclusively identifies the handoff. On receiving handoff notification, sender avoids retransmission and reduction in congestion window consequently, suspends transmissions, and stores congestion window and slow start threshold value. Reception of EHC-ACK resulting in resumes transmission with the same rate which was prior to handoff and hence transmission rate reaches to steady state immediately after reconnection. To reduce the end to end delay, data has been encapsulated after reconnection. Further recovery mechanism has been designed to recover losses during handoff. It makes sure the continuity of TCP session over Mobile IP and prevents successive timeouts and slow recovery of the congestion window during communication pauses. Implementation of Wi-TCP does not required modification at the mobile host and moreover it avoids buffering, retransmission of packets on wireless link, packet forwarding, monitoring and maintenance of records of each TCP connection. Wi-TCP requires minor modifications on base station and sender. Figure 1 shows complete mechanism of Wi-TCP with two handover events with foreign agents FA1 and FA2.

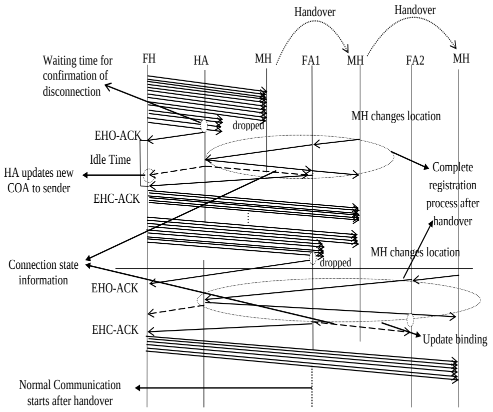
Handover
Handover
Waiting time for confirmation of disconnection
FH
HA
MH
FA1
MH
FA2
MH
dropped
MH changes location
EHO-ACK
Idle Time
HA updates new COA to sender
Complete registration process after handover
EHC-ACK
MH changes location
Connection state information
dropped
EHO-ACK
EHC-ACK
Update binding
Normal Communication starts after handover

Figure 1 Mechanism of Wi-TCP

4. SIMULATION SETUP

The topology was simulated using network simulator NS-2 [23-24] that is shown in Figure 2 and further performance has been verified with regular TCP. The system model consists of wired link of 10 Mbps, 10 ms delay between FH and router R , wired link of 10 Mbps and 5ms between R and HA/FA and a 2Mbps wireless channel. The packet size is set to 1040 bytes (including TCP + IP header) and the queue size at all links is fixed at 50. The queues used in all links are drop tail except wireless link have priority queue. In the given scenarios TCP traffic is used. An end to end TCP connection is assumed with a maximum window size of 20. In case of overlap cell, position of HA is (100, 2), FA1 is (500, 2) and FA2 is (900, 2) are defined in Figure 3.

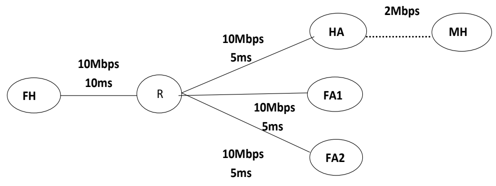

Figure 2 Simulation setup

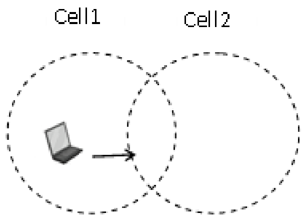

Figure 3 Movement of MH in Overlap Wireless Cell Configuration

5. PERFORMANCE ANALYSIS

Wi-TCP has been verified in overlap coverage, when mobile host moves through overlap area. To assess the performance of Wi-TCP in each simulation environment, scenario considered is fixed correspondent host transmits 3 Mbytes file to the mobile host using TCP. Further attempts are made to verify handoffs to simulate HF_H, FF_H and FH_H schemes in overlap cell configuration for evaluation of end to end throughput at the receiving host. To analyze the negative effect of handover during transmission of file and how HF_H scheme overcomes this effect in overlap is discussed in the following section.

While investigating behavior of TCP, some facts are observed concerning to each cell configuration, for example each cell pertain specific property that consequently affects the TCP performance on large scale. In mobile IP based network, mobile node need to rely on agent advertisement when it is being disconnected. MIPv4's shorter registration period, causes greater susceptibility to miss out router advertisements. Overlapped region is highly susceptible to collision at the MAC layer particular to MAC 802.11 broadcasts packets, such as router advertisements which do not get retransmitted, if lost due to collision. When mobile host moves in an overlap cell it has been observed that random collisions of advertisement makes overlap region fairly unstable. Therefore, frequently updated registrations are required, where MH can easily miss out consecutive agent advertisements which cause the time out of registration request and it further increases handoff latency. MN has been disconnected for longer time even in overlap region, if any one of the following situations occurs, if agent advertisement is missed by mobile node or registration time out occur or due to collision of agent advertisements while crossing the boundary.

At t = 2.00 sec MH starts downloading file from FH. Observations illustrate that within 0.31432 sec, total 38 packets are transferred in slow start phase and sender enters into congestion avoidance phase at 2.31432 sec. At t = 4 sec MH starts moving towards new FA. While tracing data, it is seen that when MH switching between overlapped cells MH stayed in contact with the previous agent during handoff and it gets attached to another base station when it crosses the current cell boundary, resulted in short disconnection. For a brief period after switching cells, the MH continued to receive packets from the new agent after the Mobile IP adjustments takes place. In overlap region it is seen that disconnection period is quit longer than expectation, the reason is advertisements are dropped due to collision of MAC layer packets. Here at approximately t = 11.53439 sec. MH gets disconnected resulted in HA sends EHO-ACK to sender to stop communication at t = 12.0345997483 sec and sender enters into idle state. Finally at t = 15.089167878 sec, EHC-ECK sends by foreign agent to resume communication. Figure 4 shows the loss recovery in HF_H scheme during handoff and observations shows that during handoff packets 760~779 are in transit. Figure 4 noted the TCP communication resume with sending first unacknowledged packet 760 at 15.10424442 sec and after receiving ACK 767, communication resume with full speed by sending packets 768~787. Observations articulate that within single round trip time Wi-TCP recovers lost packets and in second round it reaches to 787. In Figure 4, point A represents transmission of packet 760~779, due to handoff from home network to foreign network packets 768~780 and ACKs 760~767 are dropped are shown as point B and C. Point D and E represents retransmission after handover completion. From the analysis it is clear that even though overlap region is turbulent, Wi-TCP achieves fast recovery of lost packets and maximum throughput among all cell configurations.

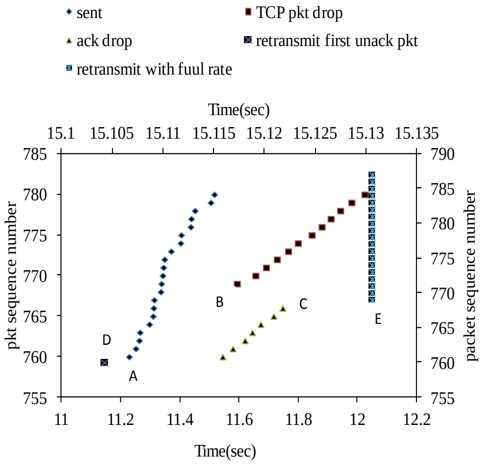

Figure 4 Loss Recovery in HF_H scheme (over-lap cell)

To compare the performance of Wi-TCP, a TCP Reno is applied in the same scenario. During analysis it observed that short period of disconnection generally occurs in overlap coverage .

In case of short handoff, it is also possible that only a portion of the packets in the current window are lost because MIP handoff completes reasonably fast in overlap coverage. Out of order received packets generate duplicate ACKs and consequently trigger fast retransmit (which can eliminate the long post-handoff cause) and fast recovery at the sender, once duplicates ACKs reached to the sender. Each TCP version has its own congestion control algorithm hence it reacts differently when packet losses occur during handoff. To analyze the effect of fast retransmit and fast recovery, a movement of MH sets accordingly. During analysis it is observed that after simulation starts, and till movement of mobile host regular TCP behaves in same way where sender first enters in slow start phase, increasing the congestion window exponentially from 1 to 20 in 2.31432 seconds and sending packets 1 to 38. Then it enters into congestion avoidance phase, which increases the window linearly, sending packets 39–788. In case of reception of 3 duplicate ACKs, loss recovery mechanism in TCP Reno is different than TCP Tahoe. The loss of the dropped packets is discovered by the sender upon the receipt of 3 duplicate ACKs. As a result, the sender enters fast-recovery mode: it reduces cwnd by half and retransmits the first unacknowledged packet. The sender exits fast-recovery after receiving an ACK for the retransmitted packet, even if this ACK does not cover all the packets transmitted before entering fast-retransmit. Usually, after 1 or 2 phases of fast-recovery the sender does not receive enough duplicate ACKs to detect the loss of additional packets and it therefore encounters a timeout. In Figure 5 the timeout occurs after a single phase of fast-recovery. The typical behavior of TCP Reno under fast retransmit and fast recovery is depicted in Figure 5.Investigation of TCP Reno behavior is discussed when the handoff occurs for a while. Figure 5 shows sequence numbers of TCP and ACK packets. The handoff occurs after the TCP sender achieves its maximum window size 20 packets 369~388 are in-flight, and then 12 packets 369~380 are dropped from window of data during the handoff. Other 8 packets (381~388) arrived at the mobile host successfully. After reconnection each time packets not dropped during the handoff arrives at the TCP receiver, it sends an ACK for packet 369. Then, the TCP sender begins Fast Retransmit upon receiving three duplicate ACKs for packet 369. At this time, ssthresh and congestion window are 10 and 13 packets, respectively. Congestion window grows by one packet upon receiving following duplicate ACKs for packet 369.ACK for 370 arrives, but this ACK does not cover all of data outstanding prior to the Fast Retransmit so that the TCP sender cannot send new packets. It therefore begins slow–start after the retransmission timeout for packet 370. Then it sends single packet 370 and in the next round trip time, it retransmits 2 packets 371~372 after receiving ACK 370. When packet 372 arrives at the TCP receiver, it sends an ACK for 372.This ACK forces the TCP sender to transmit 4 packets 373~376. Then, the TCP sender transmits 8 packet 377~384. Upon receiving three duplicate ACKs for packets 384 triggered by packets 377~384, receiver sends ACKs for 389, because packets 381~388 which are already received by MH. Receiver generates duplicate ACKs for 389. To reach a recovery of packet 388, around 4 round trips are required. In the case of TCP Reno, the performance is considerably improved by the fast retransmission and fast recovery method than TCP Tahoe.

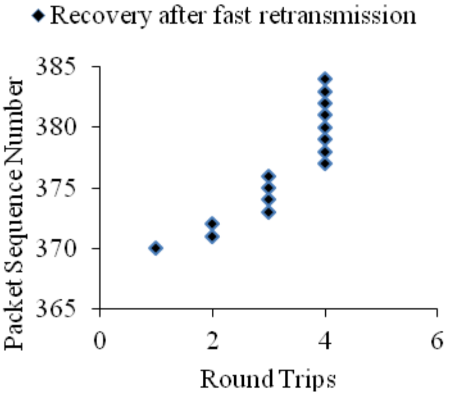

Figure 5 Loss Recoveries TCP Reno (Fast retransmit and Fast Recovery)

6. CONCLUSION

In wireless access technology, handoff delay can significantly degrade the end to end performance of TCP. In this paper, we have presented a comparative analysis of TCP Reno and Wi-TCP. Simulation results show that Wi-TCP significantly improves the TCP performance compared to that of regular TCP over Mobile IP supported WLAN during handoff.During disconnection due to handoff, Wi-TCP performs better in terms of throughput and transmission time compared to TCP Reno and moreover it does not require modifications at sender and mobile host. During analysis, it is perceived that overlap cells are fairly unstable compared to adjacent and non-overlap cells because there is high chances to miss out agent advertisement packets due to collision of TCP packets, acknowledgements and agent advertisement packets itself, may be resulting in time-out at sender.After reconnection our investigations have shown, that the reaction of TCP's during mobile hand-offs depending on its version, and hence resulted throughput is different.

7. REFERENCES

[1] Jacobson, V., "Congestion Avoidance and Control", ACM SIGCOMM 1988, Computer Communication Review, pp. 314-329

[2]Jacobson, V. (1990). Modified TCP congestion avoidance algorithm, end2end mailing list (ftp://ftp.isi.edu/end2end-interest-1990.mail), April 30.

[3] S. Floyd, & Henderson, T. (April 1999). RFC 2582 - The NewReno Modification to TCP's Fast Recovery Algorithm

[4] Bakre, A., & Badrinath, B. R. (1995). I-TCP: indirect TCP for mobile hosts. Paper presented at the Distributed Computing Systems, 1995., Proceedings of the 15th International Conference.

[5] Brown K. and Singh S. "M-TCP: TCP for Mobile Cellular Networks", ACM SIGCOMM Computer Communication Review Volume 27 Issue 5, Oct. 1997

[6] Wang K. Y.,Tripathi S. K., "Mobile-End transport Protocol: An alternative to TCP/IP OverWireless Links," Proceedings of INFOCOM, vol. 3, pp. 1046–1053, 1998

[7]Schmidt, D., Tabery, P., Schwingenschlgl ,C., and Bachmeir C., "Handoff-Triggered TCP (hot-TCP):Performance and Fairness Evaluation",IEEE Vehicular Technology Conference, VTC Fall 2004, Los Angeles, USA, September 2004

[8] Balakrishnan, H., & Katz, R. H. Explicit Loss Notification and Wireless Web Performance. Proc. IEEE Globecom Internet Mini-Conference ,1998

[9] Singh Ajay Kr. and Iyer S: "ATCP: Improving TCP performance over mobile wireless environment", Fourth International IEEE Workshop on Mobile and Wireless Communications Network, 2002.

[10] Haruki I., Ichiro Y.and Jiro k.: "An Efficient TCP with Explicit Handover Notification for Mobile Networks" WCNC 2004, IEEE Communication Society

[11] Buchholez G. O., Thomas Z.,Do T.,V., "TCP-ELN:On the protocol aspects and performance of explicit loss notification for TCP over wireless networks",WICON 05 pp 172-179.

[12] Daniel, L., Kojo,M., "Adapting TCP for Vertical Handoffs in Wireless Networks", IEEE 31st Conference on Local Computer Networks (LCN), Tampa, USA, pp. 151 - 158, Nov. 2006.

[13] Wang, N. C. , Wang Y. Y., and Chang S. C., "A Fast Adaptive Congestion Control Scheme for Improving TCP Performance during Soft Vertical Handoff", IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong, China, pp. 3641 - 3646, Mar. 2007.

[14] Daniel, L., Kojo,M., "Combating Packet Reordering in Vertical Handoff using Cross-layer Notifications to TCP", IEEE 4th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Avignon, France, pp. 297 - 303, Oct. 2008.

[15] Yavatkar, R., & Bhagawat, N. "Improving end-to-end performance of TCP over mobile internetworks", Mobile Computing Systems and Applications, 1994.

[16] Chawla R., Nandi S., "TCP FECN:A unified solution for wireless network",ICCS 2002,volume 2,pp815-819

[17]Goff, T., Moronski, J., Phatak, D. S., & Gupta, V. (2000). Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments. Paper presented at the INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.

[18] Singh Ajay Kr. and Iyer S: "ATCP: Improving TCP performance over mobile wireless environment", Fourth International IEEE Workshop on Mobile and Wireless Communications Network, 2002.

[19] Li, Y and Lillykutty, J. "Proactive-WTCP an end to end mechanism to improve TCP Performance over Wireless links", IEEE LCN 03, 2003

[20] EOM, Doo Seop, Sugano ,M., Murata, M., Miyahara, H , "Improving TCP handoff performance in Mobile IP based Network", Elsevier communication System 25 (2002),pp 635-646

[21] Song Y.,Suh Y., "Rate-Control Snoop:A Reliable Transport Protocol for heterogeneous Network with wired and wireless Links", Wireless Communication and Networking WCNC 2003,pp1334-1338

[22] Perkins, C.E., "Mobile IP”, IEEE Communication Magazine, May 1997

[23] The ns Manual May 9, 2010

[24] LBNL’s Network Simulator