GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY
Volume 12 Issue 9 Version 1.0 April 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

TCP Performance Over Mobile IP Wired-cum- Wireless Networks

By Md. Asif Nashiry, Shauli Sarmin Sumi, Subrata Kumar Das
& Md. Kamrul Islam

Jessore Science & Technology University, Jessore, Bangladesh

Abstract - Reliable transport protocol like TCP has served well the wired Internet where the packet losses are mainly due to congestion, but is not ready for Mobile IP wired-cum-wireless environments where the significant packet losses are due to bit errors and handoffs. In this paper, we have investigated the performance of TCP among the various TCP variants. We have observed both TCP senders (Newreno & Vegas) and TCP receivers (Base & Delayed-Ack). Using ns-2, we have evaluated the TCP throughput and packet delay over a single TCP connection. The simulation results suggest that a particular combination (one TCP sender with one TCP receiver) of TCP shows the best result in such Mobile IP networks.

Keywords : *Wired-cum-Wireless Networks, TCP, Mobility, Mobile IP, Handoff.*

GJCST Classification: *C.2.1,C.2.5*

TCP PERFORMANCE OVER MOBILE IP WIRED-CUM- WIRELESS NETWORKS

Strictly as per the compliance and regulations of:

TCP Performance Over Mobile IP Wired-cum-Wireless Networks

Md. Asif Nashiry [α], Shauli Sarmin Sumi [σ], Subrata Kumar Das [ρ] & Md. Kamrul Islam [ω]

Abstract - Reliable transport protocol like TCP has served well the wired Internet where the packet losses are mainly due to congestion, but is not ready for Mobile IP wired-cum-wireless environments where the significant packet losses are due to bit errors and handoffs. In this paper, we have investigated the performance of TCP among the various TCP variants. We have observed both TCP senders (Newreno & Vegas) and TCP receivers (Base & Delayed-Ack). Using ns-2, we have evaluated the TCP throughput and packet delay over a single TCP connection. The simulation results suggest that a particular combination (one TCP sender with one TCP receiver) of TCP shows the best result in such Mobile IP networks.

Keywords: Wired-cum-Wireless Networks, TCP, Mobility, Mobile IP, Handoff.

I. INTRODUCTION

TCP was basically developed assuming that it would run on wired networks. Wired networks usually have less bit error rates and hence less packet loss. Also, since mobility was not considered, there are no packet losses and delays caused by mobility. Hence, TCP was designed to assume that any segment loss is caused by the network congestion and on seeing any segment loss; TCP invokes its congestion control measures. In various circumstances wired and wireless networks are connected together to take the advantages of both. Usually, the wireless links have high bit error rates. Also, temporary disconnections occur because of the factors like channel fading, and handoffs when a mobile node is in motion. The handoff period depends on both the link level handoff protocol being used as well as the IP level handoff protocol being used, if any. The standard TCP implementations assume the cause of any packet loss to be the network congestion. Then, they reduce the congestion size to a minimum. Also, TCP invokes the slow-start mechanism. If the network links are slow, it takes long time to grow the congestion window. Thus, reduction in the size of the congestion window reduces the transmission rate and hence degrades the performance. Also, TCP undergoes a binary exponential backoff, causing long pauses of communication. Because of this, some time may be unused even after the mobile node recovers from the temporary disconnection. It has been observed that even a single wireless link can reduce the TCP performance considerably. In theory we can use existing transport protocols like TCP on a wireless host to communicate with a fixed network. Though this keeps the transport layer transparent to mobility, disconnection and other features of wireless and mobile hosts reduce the performance. Thus TCP proved not so efficient in Mobile IP wired-cum-wireless environments and suffers from performance degradation introduced by the conditions that exist in those environments. Hence, an optimized reliable transport protocol is very important for the development of the wireless Internet. This paper provides a realistic comparison of the performances of the various TCP variants over Mobile IP wired-cum-wireless network. The TCP variants include both TCP senders (TCP Newreno and TCP Vegas) and TCP receivers (Base TCP Sink and Delayed-Ack TCP Sink). The reminder of this paper is organized as follows: Section 2 depicts the related works; mobile IP and TCP variants are described in Section 3. We define the simulation model in Section 4 and present the results and discussions in Section 5. The conclusion and future work follow in Section 6.

II. RELATED WORKS

TCP/IP is the standard networking protocol stack for Internet. So it is expected to be deployed over wireless networks to allow seamless integration with the Internet. However, earlier research suggested that TCP performs poorly over cellular (single-hop) wireless networks (Holland & Vaidya, 1999). A good amount of research has been done in the past to improve the performance of TCP in the conditions that prevail in mobile and wireless environments. Split connection approach is one of them. It separates TCP connection, one for the wireless network and the other for the wired network between a sender and a receiver. The motivation behind the use of separate connections is the fact that the characteristics of a wireless network are different from the characteristics of a wired network, hence a separate connection that is optimized to the

---

Author α : Md. Asif Nashiry, Department of Computer Science & Engineering, Jessore Science & Technology University, Jessore, Bangladesh. E-mail: asif.nashiry@yahoo.com

Author σ : Shauli Sarmin Sumi, Department of Information & Communication Engineering, Islamic University, Kushtia-7003, Bangladesh. E-mail: shaulisumi@yahoo.com

Author ρ : Subrata Kumar Das, Department of Computer Science & Engineering, Jessore Science & Technology University, Jessore, Bangladesh. E-mail: sdas_ce@yahoo.com

Author ω : Md. Kamrul Islam, Department of Computer Science & Engineering, Jessore Science & Technology University, Jessore, Bangladesh. E-mail: polash_2k4@yahoo.com

kind of network it operates on, can be used. I-TCP (Bakre & Badrinath, 1995), ELN (Balakrishnan & Katz, 1998) can be classified in this group.

Enhanced link layer approach is another solution. Since the basic factor that causes degradation in wireless environments is the nature of the transmission medium, an improvement in the link technology that operates over the physical layer can improve the overall performance. This was the idea behind enhanced link layer solutions. Mechanisms like ARQ, FEC (Chockalingam Zorzi & Tralli, 1999) were suggested to improve the performance at the lower layers which, in turn, can reduce the chances of TCP getting timed out and does not invoke its congestion control measures.

Most of the above-mentioned solutions require special support from the network infrastructure in some form or the other. Some other approaches were suggested that do not require any special support from the intermediate infrastructure and changes are confined to the end nodes. Path Prediction (Hadjiefthymiades, Papayiannis & Merakos, 2002) and ACK Regulator (Chan & Ramjee, 2002) come under this category. These solutions are not bound by the problem of encrypted traffic, and they do not assume any special support from the network infrastructure.

As suggested earlier, most related work focuses on to improve the TCP performance in wireless network. Also most of the earlier solutions (except ELN) try to optimize the mobile host acting as a TCP receiver but do not consider the case of a mobile TCP sender, which can also be a common case in the future. Also a few of the earlier work measured TCP performance with combining both the TCP senders and the TCP receivers. This paper provides a realistic comparison of performance of the various TCP variants over Mobile IP wired-cum-wireless network. The TCP variants include both TCP senders (TCP Newreno and TCP Vegas) and TCP receivers (Base TCP Sink and Delayed-Ack TCP Sink).

III. MOBILE IP AND TCP VARIANTS

In order to achieve the mobility function, the Internet Protocol (IP) has extended to become the Mobile Internet Protocol (Mobile IP or MIP). Mobile IP provides hosts with the ability to change their point of attachment to the network without compromising their ability to communicate. The mobility support provided by Mobile IP is transparent to the other protocol layers so as not to affect the operation of applications which do not have the mobile capability.

Among various IP mobility proposals, Mobile IPv4 (Charles, 1996) & (Perkins, 2002) is the oldest and probably the most widely known mobility management proposal with IP. MIPv4 introduces three new entities required to support the protocol: the Home Agent (HA), the Foreign Agent (FA) and the Mobile Host (MH). Home and foreign agents are introduced for mobility management. Each time a mobile host connects to a network at a new location, it will obtain a temporary address, called Care-of Address (COA) from a foreign agent in the local network. Then the mobile host must inform its home agent of the new address by a registration procedure, which begins when the mobile host, possibly with the assistance of the foreign agent, sends a registration request with the COA. When the home agent receives this request, it may typically add the necessary information to its routing table, approve the request, and send a registration reply back to the mobile host.

A basic function of TCP is to provide reliable communication over an unreliable network layer. TCP ensures that the data is delivered from the sending process to the receiving process correctly and in order. TCP controls the traffic flow and network congestion by maintaining window (buffer to store data packets); and various TCP variants are classified according to the mechanism for maintaining this window.

a) TCP New-Reno

A modification to Reno TCP called New-Reno TCP (Hoe, 1996). Since most TCP sessions last only for a short period of time, the initial slow start period is significant for the overall performance. A method is proposed to estimate an optimum slow start threshold (ssthresh) value by calculating the byte equivalent of bandwidth delay product of the network when a new connection is made. TCP New-Reno also deals with multiple packet losses from a single window. If two or more segments have been lost from the transmitted data (window), the Fast Retransmission and Fast Recovery algorithms will not be able to recover the losses without waiting for retransmission time out. New-Reno overcomes this problem by introducing the concept of a Fast Retransmission Phase, which starts on detection of a packet loss (receiving 3 duplicate ACKs) and ends when the receiver acknowledges reception of all data transmitted at the start of the Fast Retransmission phase.

The transmitter assumes reception of a partial ACK during the Fast Retransmission phase as an indication that another packet has been lost within the window and retransmits that packet immediately to prevent expiry of the retransmission timer. New Reno sets the congestion window (cwnd) to one segment on reception of 3 duplicate ACKs (i.e. when entering the Fast Retransmission Phase) and unacknowledged data are retransmitted using the Slow Start algorithm. The transmitter is also allowed to transmit a new data packet on receiving 2 duplicate ACKs. While the transmitter is in the Fast Retransmission Phase, it continues to retransmit packets using Slow Start until all packets have been recovered (without starting a new retransmission phase for partial ACKs). Although this modification may cause unnecessary retransmissions, it reduces transmitter timeouts and efficiently recovers multiple packet loss using partial ACKs.

b) TCP Vegas

TCP Vegas (Brakmo & Peterson, 1995) does not involve any changes to the TCP specification; it is merely an alternative implementation that is able to interoperate with any other valid implementation of TCP. In fact, all the changes pare confined to the sending side. There are some techniques that Vegas employs to increase throughput and decrease losses.

It calculates the RTT more accurately. Using an accurate RTT estimate serves two purposes. First, it leads to a more accurate timeout calculation. Second, it is used in conjunction with the mechanism described next to decide to retransmit a dropped segment in a more timely fashion. It includes the new mechanism for deciding to retransmit Vegas treats the receipt of certain ACKs as a trigger to check if a timeout should happen. Moreover, Vegas only decreases the congestion window if the retransmitted segment was previously sent after the last decrease. Any losses that happened before the last window decrease do not imply that the network is congested for the current congestion window size, and therefore do not imply that it should be decreased again. Another most important feature of TCP Vegas is that the detection of congestion. Vegas calculates the expected throughput by:

Expected = WindowSize / BaseRTT

where WindowSize is the size of the current congestion window and BaseRTT is the RTT of a segment when the connection is not congested. It then calculates the current Actual throughput of the network. Then, Vegas compares Actual to Expected, and adjusts the window accordingly.

c) Base TCP Sink

The base TCP sink object is responsible for returning ACKs to a peer TCP source object. It generates one ACK per packet received. The size of the ACKs may be configured. The creation and configuration of the TCP sink object is generally performed automatically by a library call.

d) Delayed-Ack TCP Sink

A Delayed-Ack sink object is available for simulating a TCP receiver that ACKs less than once per packet received. This object contains a bound variable, which gives the number of seconds to wait between ACKs. The Delayed-Ack sink implements an aggressive ACK policy whereby only ACKs for in-order packets are delayed. Out-of-order packets cause immediate ACK generation. It is possible to configure the bound variable (interval) of the Delayed-Ack TCP Sink.

IV. SIMULATION MODEL & DESIGN

To design the Mobile IP wired-cum-wireless network topology we have considered various simulation parameters, which are supported by the network simulator-2 (NS-2) (Fall & Varadhan, 2005). We have done simulation for two different topologies and call them topology 'A' and topology 'B' respectively. There is a roaming mobile node called Mobile Host (MH) that moves between its home agent and foreign agent in a certain velocity. Data will be exchanged between the roaming mobile node (MH) and the wired node W(0) via the two base-station nodes (HA or FA). Base station nodes are like gateways between wireless and wired domains and allow packets to be exchanged between the two different types of nodes. The initial positions, (x,y), of the nodes MH, HA and FA are

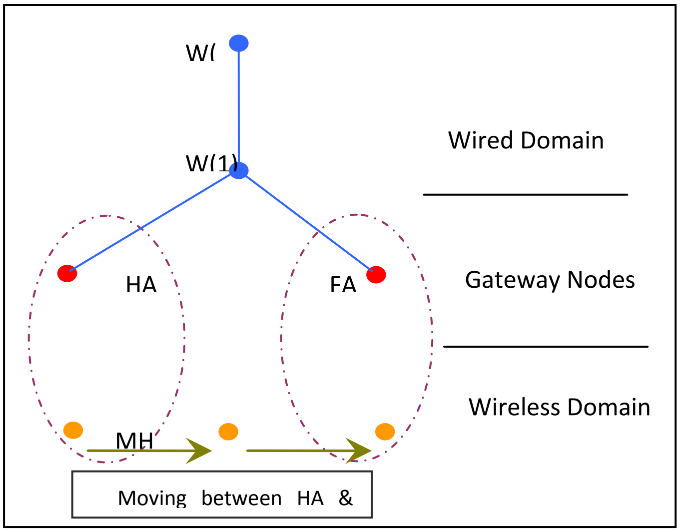

Figure 1: Topology 'A' for Mobile IP wired-cum-wireless simulation

(50,100), (200,300) and (500,300) respectively in the 700mX800m topology. In this network, the mobile node MH starts moving from position (50,100) at 10 sec towards destination (650,100) at fixed speeds (0, 5, 10, 15, 20, 25, or 30) m/s. After reaching the destination position (650,100), MH will stay there 1sec (pause time) and then again moves back to its initial starting position (50,100) at the same velocity. In this way, the mobile node MH changes its home agent (HA) and foreign agent (FA) several times depending upon its speed. We have used DSDV ad hoc routing protocol. FTP begins transferring packets of size 1000 bytes after 10 second. Each wireless node has a buffer size of 50 packets. Each simulation ends at 150 seconds.

There are fewer changes in Topology 'B'. We have used three base-station nodes in topology 'B' and call them Home Agent (HA), Foreign Agent-1 (FA1) and Foreign Agent-2 (FA2) respectively. The initial positions, (x,y), of the nodes MH, HA, FA1 and FA2 are (50,100), (200,300), (500,300) and (800,300) respectively in the 1000mX800m topology. In this case, simulation starts at 10 sec and ends at 190 sec. Here the mobile node MH starts moving from position (50,100) at 10 sec towards destination position (950,100) at a fixed speed of (0, 5, 10, 15, 20, 25, or 30) m/s while nodes HA, FA1 and FA2 serve as base station (gateway) nodes. As before, the pause time is 1 sec, that is, the mobile node MH will stay 1 sec at position (950,100) and then moves back to its initial position (50, 100) at the same velocity. Staying there 1 sec, it moves again towards the destination position.

V. PERFORMANCE ANALYSIS & DISCUSSION

This section describes a realistic analysis comparing the performance of the various TCP variants over multi-hop Mobile IP wired-cum-wireless networks. We have combined the TCP senders (New-Reno & Vegas) with the TCP receivers (Base & Delayed-Ack). We have considered the mobile node as a TCP sender.

a) Throughput Analysis

We have evaluated the throughput for topology 'A' obtained with the TCP variants in terms of the total number of packets received at the destination node per unit time over intervals of 7 seconds for mobility 0 meter/sec, 15 meter/sec and 30 meter/sec and the results are plotted in figures.

From figure 2, we observe that after the initial connection establishment, all the variants achieve almost fixed throughput to the whole simulation time. It is common that if the window size increases, the throughput will also increase.

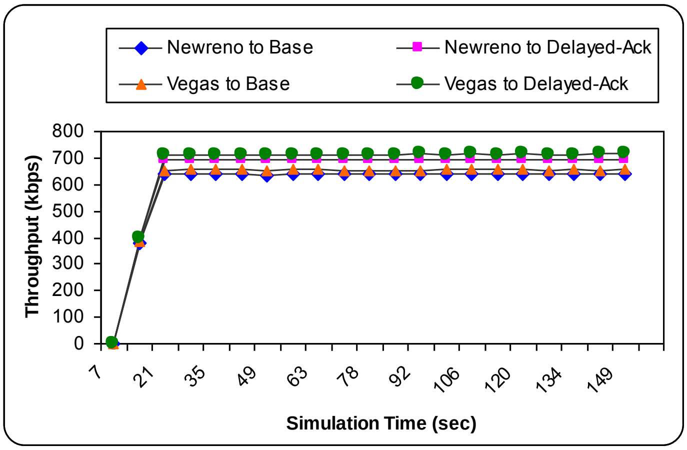

Figure 2 : Throughput (kbps) versus Simulation Time for topology 'A' for mobility 0 (meter/sec)

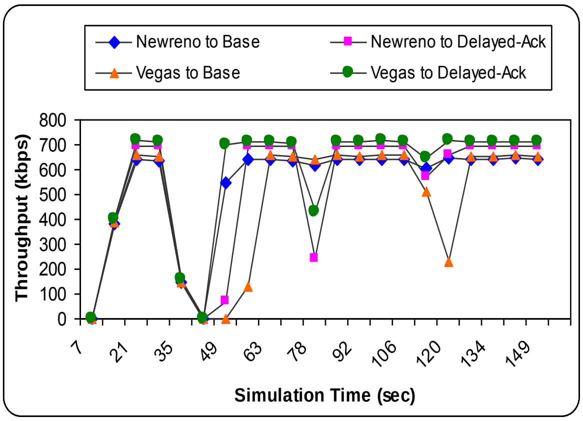

Figure 3(a) : Mobility 15 (m/s)

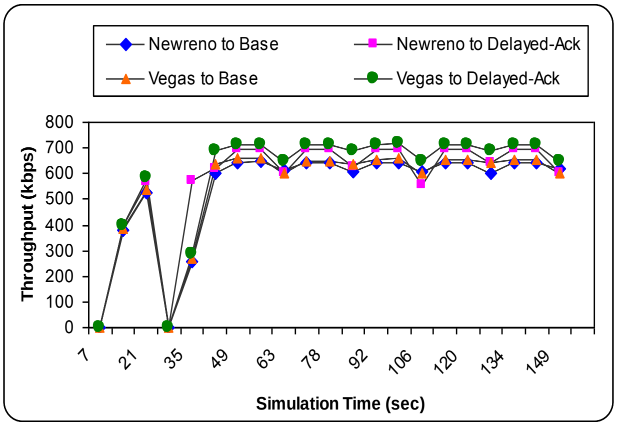

Figure 3(b) : Mobility 30 (m/s)

Figure 3 : Throughput (kbps) versus Simulation Time for topology 'A'

Among the four variants, Vegas to Delayed-Ack shows the best throughput. From figure 3 (a), it is seen that the throughput of each of the TCP variant decreases drastically. By comparing this figure to figure 2, it is observed that the throughput performance is worse in this case. From the start of the simulation to 35 seconds, throughput is the same (almost 700 kbps) as before (figure 2). But after 35 seconds it shows different result. The mobility of the mobile host is the cause of this degradation. In the case of figure 2, the mobile host remains stationary. But in this case the mobile host changes its base station for a number of times during simulation time. When changing base station a mobile host requires a handoff. The mobile host changes base station three times; so require three handoffs. And at the time of each handoff packets are dropped and throughput decreases.

Figure 3(b) shows the similar type of result except the throughput is reduced more times than that of figure 3(a). Since in this case, the mobility is 30 meter/sec, more handoff is required and at the time of each handoff throughput is reduced. One interesting feature which is observed from figures 3(a) and 3(b) is that during first handoff throughput performance decreases drastically but during the second, third and successive handoffs throughput decreases slightly. But in such an environment Vegas to Delayed-Ack performs well among the TCP variants. Its throughput reduces less than that of the others. Concentrating on the receiver portion in figure 2, 3(a) and 3(b), we see that the performance of the Base TCP receivers is lower than that of the Delayed-Ack TCP receivers; no matter whether the TCP sender is New-Reno or Vegas. Since all the terminals in Mobile IP network share the same radio channel to send and receive packets, the collision of data packets and acknowledgement (ACK) packets in the radio channel can severely reduce TCP throughput. But Delayed-Ack receiver employs less ACK packets. So, for transferring TCP data packets, TCP sender can utilize the channel more. Hence throughput of any TCP sender synchronized with Delayed-Ack receiver is better than that of TCP sender with Base receiver.

Now with fixed Delayed-Ack receiver, we are going to observe the performance of New-Reno and Vegas sender. By evaluating the throughput from three different mobility scenarios it is seen that TCP Vegas sender performs well. As stated earlier, TCP Vegas estimates the network status to adjust its window value. This property enriches its performance in such a heterogeneous environment.

We also average the throughput over the simulation duration and plot the average throughput obtained with the TCP variants as a function of node mobility for both topology ‘A’ and ‘B’ in figure 4 and figure 5 respectively.

According to figure 4 and 5, when the source node is stationary (node mobility 0 meter/sec), all the TCP variants achieve their own highest throughputs in both topology ‘A’ and topology ‘B’, because of occurring no handoffs in this situation.

The throughputs achieved with New-Reno to Base TCP and Vegas to Base TCP are lower than those achieved with New-Reno to Delayed-Ack TCP and Vegas to Delayed-Ack TCP in both topology ‘A’ and topology ‘B’. It is due to the fact that ACKs and TCP packets share the same channel; so decreasing the ACKs by using the Delayed-Ack receiver instead of Base receiver it is possible to improve the throughput of TCP.

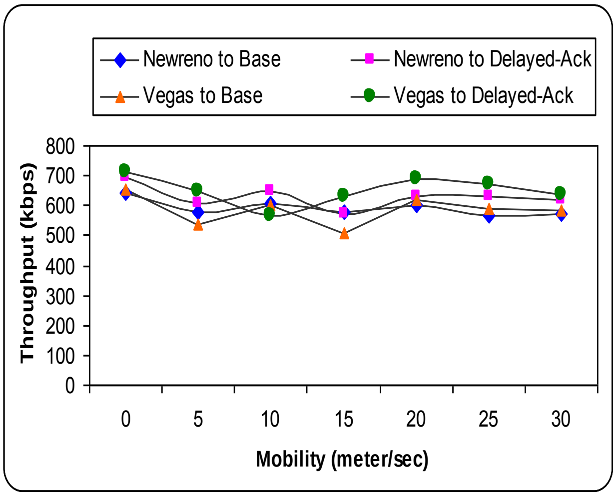

Figure 4 : Throughput vs Mobility for topology ‘A’

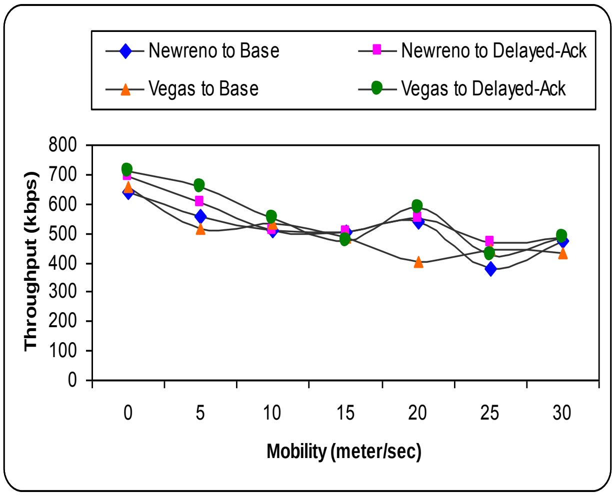

Figure 5 : Throughput vs Mobility for topology ‘B’

b) *Average Delay Analysis*

From figure 6 and figure 7, it is seen that the average delay of New-Reno sender (receiver may be Base or Delayed-Ack) is high as compared to Vegas sender in both topologies. Vegas reads and records the system clock each time a segment is sent. When an ACK arrives, Vegas reads the clock again and does the round trip time (RTT) calculation using this time and a timestamp recorded for the relevant segment. Using an accurate RTT estimates, Vegas leads to a more accurate timeout calculation. This is why Vegas shows very low delay as compared to New-Reno. The average delay of New-Reno to Delayed-Ack is the worst among all the variants. This is why it has given the lowest packet delivery ratio. The average delay of New-Reno to Base is better than that of New-Reno to Delayed-Ack but worse than Vegas to Base and Vegas to Delayed-Ack. The same thing has happened in case of packet delivery ratio, that is, the packet delivery ratio of New-Reno to Base is better than New-Reno to Delayed-Ack but worse than the Vegas to Base and Vegas to Delayed-Ack.

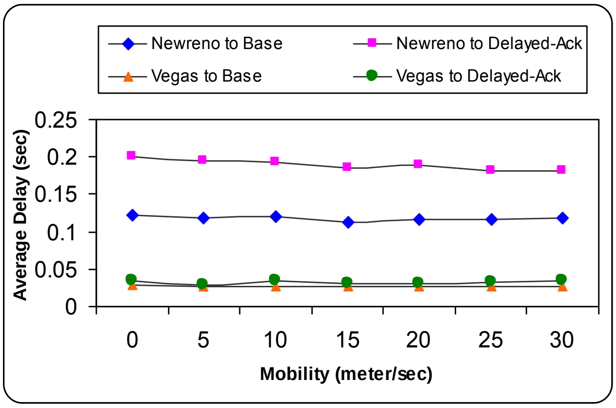

Figure 6 : Average delay vs Mobility for topology A

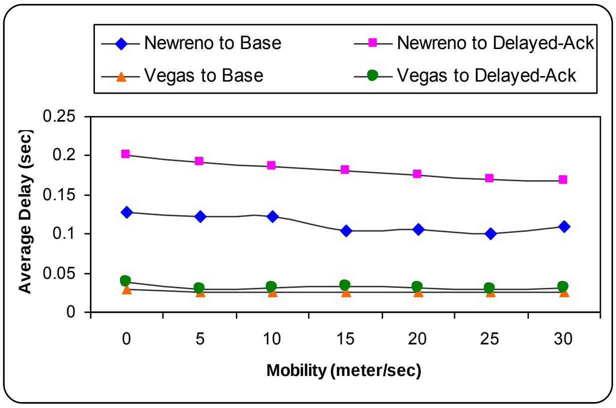

Figure 7: Average delay vs Mobility for topology B

The delay performance of Vegas to Delayed-Ack TCP and Vegas to Base TCP is relatively similar and they show low delay in all the mobility so they show relatively similar and high packet delivery ratio in both the topologies. Since the average delay of Vegas to Delayed-Ack TCP is low it has given a very good throughput in the simulation time in both topologies.

VI. Conclusion and Future Works

While TCP is needed for a Mobile IP network, the current variants of TCP are not adequate for the task. To achieve good TCP performance in Mobile IP wired-cum-wireless networks a variety of solutions have been proposed and in most of the solutions the mobile node acts as a TCP receiver. However, little attention has been paid to study the performance of TCP traffic over combining both TCP senders and receivers. We have investigated the performance of TCP over combining both TCP senders (Newreno & Vegas) and TCP receivers (Base & Delayed-Ack) using simulations in ns-2 for a range of node mobility with a single traffic source. The performance metrics that we have considered include TCP throughput and packet delay. On throughput performance, Vegas TCP to Delayed-Ack gains the best performance. On the sender side, the effective congestion control technique of Vegas makes it optimal in such a heterogeneous environment. And in the receiver side, the Delayed-Ack receiver makes the proper use of channel by sending fewer ACK packets.

We have also evaluated throughput with respect to node mobility. The result shows that throughput of each TCP variant varies similarly with respect to each other against mobility of node. But Vegas to Delayed-Ack shows the best result as compared with other variants. Delay performance of each variant shows the confidential result. TCP Vegas has the lowest delay no matter whether the receiver is Base or Delayed-Ack. New-Reno has large delay. But among them New-Reno to Delayed-Ack has the highest delay. After observing packet delivery ratio and delay, it is seen that the variant that gains highest packet delivery ratio has lowest delay. Though all the variants of TCP suffer in Mobile IP wired-cum-wireless networks, it is clear from the various results that if we use Vegas TCP as a source (sender) and Delayed-Ack as a sink (receiver), then this combination shows the best result among the all TCP variants. Next, we plan to investigate the performance of all the TCP variants with multiple traffic sources in more complex heterogeneous networks and also consider some link layer and network layer feedbacks to improve their performance.

References Références Referencias

1. G. Holland and N. Vaidya (1999). Analysis of tcp performance over mobile ad hoc networks. Proceedings of ACM Mobicom, pp. 219–230.
2. Bakre, A., Badrinath B.R. (1995) .I-TCP: Indirect TCP for Mobile Hosts. Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS), p. 136-143.
3. Balakrishnan, H., Katz, R. (1998). Explicit Loss Notification and Wireless Web Performance. Proceedings of IEEE Globecom Internet Mini-Conference, Sydney, Australia.
4. Chockalingam, A., Zorzi, A., Tralli, V. (1999). Wireless TCP Performance with Link Layer FEC/ARQ. Proceedings of IEEE ICC'99, p.1212-1216.
5. S. Hadjiefthymiades, S. Papayiannis, and L. Merakos (2002). Using Path Prediction to Improve TCP Performance in Wireless/Mobile Communications. IEEE Communication Magazine.
6. M. C. Chan and R. Ramjee (2002). TCP/IP Performance over 3G Wireless Links with Rate and Delay variation. In Proceedings of MOBICOM.
7. Perkins E. Charles (1996). IP Mobility Support. RFC 2002.

8. Perkins, C (2002), IP mobility support for IPv4, IETF RFC 3344.
9. J. Hoe (1996). Improving the start-up behavior of a congestion control scheme for TCP. In ACM SIGCOMM.
10. L. S. Brakmo 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.
11. K. Fall and K. Varadhan (2005) “ns Notes and Documentation. Lawrence Berkeley National Laboratory, 28, http:/www- mash. cs. Berkeley. edu/ ns/.

Global Journals Inc. (US) Guidelines Handbook 2012

www.GlobalJournals.org