# 

Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.* 

# <sup>1</sup> TCP Performance Over Mobile IP Wired-cum-Wireless Networks

Dr. Md. Asif Nashiry<sup>1</sup>, Shauli Sarmin Sumi<sup>2</sup> and Subrata Kumar Das<sup>3</sup>

1

Received: 11 April 2012 Accepted: 3 May 2012 Published: 15 May 2012

#### 6 Abstract

7 Reliable transport protocol like TCP has served well the wired Internet where the packet

<sup>8</sup> losses are mainly due to congestion, but is not ready for Mobile IP wired-cum-wireless

<sup>9</sup> environments where the significant packet losses are due to bit errors and handoffs. In this

<sup>10</sup> paper, we have investigated the performance of TCP among the various TCP variants. We

<sup>11</sup> have observed both TCP senders (Newreno Vegas) and TCP receivers (Base Delayed-Ack).

<sup>12</sup> Using ns-2, we have evaluated the TCP throughput and packet delay over a single TCP

<sup>13</sup> connection. The simulation results suggest that a particular combination (one TCP sender

<sup>14</sup> with one TCP receiver) of TCP shows the best result in such Mobile IP networks.

15

2

3

4

5

16 Index terms—Wired-cum-Wireless Networks, TCP, Mobility, Mobile IP, Handoff

### 17 **1** Introduction

CP was basically developed assuming that it would run on wired networks. Wired networks usually have less bit 18 error rates and hence less packet loss. Also, since mobility was not considered, there are no packet losses and 19 delays caused by mobility. Hence, TCP was designed to assume that any segment loss is caused by the network 20 congestion and on seeing any segment loss; TCP invokes its congestion control measures. In various circumstances 21 wired and wireless networks are connected together to take the advantages of both. Usually, the wireless links 22 23 have high bit error rates. Also, temporary disconnections occur because of the factors like channel fading, and 24 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 25 assume the cause of any packet loss to be the network congestion. Then, they reduce the congestion size to a 26 minimum. Also, TCP invokes the slow-start mechanism. If the network links are slow, it takes long time to 27 grow the congestion window. Thus, reduction in the size of the congestion window reduces the transmission rate 28 and hence degrades the performance. Also, TCP undergoes a binary exponential backoff, causing long pauses 29 of communication. Because of this, some time may be unused even after the mobile node recovers from the 30 temporary disconnection. It has been observed that even a single wireless link can reduce the TCP performance 31 considerably. In theory we can use existing transport protocols like TCP on a wireless host to communicate with 32 a fixed network. Though this keeps the transport layer transparent to mobility, disconnection and other features 33 34 of wireless and mobile hosts reduce the performance. Thus TCP proved not so efficient in Mobile IP wired-35 cum-wireless environments and suffers from performance degradation introduced by the conditions that exist in 36 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 37 over Mobile IP wired-cumwireless network. The TCP variants include both TCP senders (TCP Newreno and 38 TCP Vegas) and TCP receivers (Base TCP Sink and Delayed-Ack TCP Sink). The reminder of this paper is 39 organized as follows: Section 2 depicts the related works; mobile IP and TCP variants are described in Section 3. 40 We define the simulation model in Section 4 and present the results and discussions in Section 5. The conclusion 41

42 and future work follow in Section 6.

### 43 2 II. RELATED WORKS

TCP/IP is the standard networking protocol stack for Internet. So it is expected to be deployed over wireless 44 networks to allow seamless integration with the Internet. However, earlier research suggested that TCP performs 45 poorly over cellular (single-hop) wireless networks (Holland & Vaidya, 1999). A good amount of research has 46 been done in the past to improve the performance of TCP in the conditions that prevail in mobile and wireless 47 environments. Split connection approach is one of them. It separates TCP connection, one for the wireless 48 network and the other for the wired network between a sender and a receiver. The motivation behind the use of 49 separate connections is the fact that the characteristics of a wireless network are different from the characteristics 50 51 of a wired network, hence a separate connection that is optimized to the kind of network it operates on, can be used. I-TCP (Bakre & Badrinath, 1995), ELN (Balakrishnan & Katz, 1998) can be classified in this group. 52 Enhanced link layer approach is another solution. Since the basic factor that causes degradation in wireless 53

54 environments is the nature of the transmission medium, an improvement in the link technology that operates over 55 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

57 at the lower layers which, in turn, can reduce the cha 58 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

<sup>69</sup> provides a realistic comparison of performance of the various TCP variants over Mobile IP wired-cum-wireless

<sup>70</sup> network. The TCP variants include both TCP senders (TCP Newreno and TCP Vegas) and TCP receivers (Base

71 TCP Sink and Delayed-Ack TCP Sink).

### 72 3 III. MOBILE IP AND TCP VARIANTS

73 In order to achieve the mobility function, the Internet Protocol (IP) has extended to become the Mobile Internet 74 Protocol (Mobile IP or MIP). Mobile IP provides hosts with the ability to change their point of attachment to 75 the network without compromising their ability to communicate. The mobility support provided by Mobile IP 76 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.

## 91 4 a) TCP New-Reno

A modification to Reno TCP called New-Reno TCP (Hoe, 1996). Since most TCP sessions last only for a short 92 period of time, the initial slow start period is significant for the overall performance. A method is proposed 93 94 to estimate an optimum slow start threshold (ssthresh) value by calculating the byte equivalent of bandwidth 95 delay product of the network when a new connection is made. TCP New-Reno also deals with multiple packet 96 losses from a single window. If two or more segments have been lost from the transmitted data (window), the 97 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 98 Phase, which starts on detection of a packet loss (receiving 3 duplicate ACKs) and ends when the receiver 99 acknowledges reception of all data transmitted at the start of the Fast Retransmission phase. 100

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 103 ACKs (i.e. when entering the Fast Retransmission Phase) and unacknowledged data are retransmitted using the 104 Slow Start algorithm. The transmitter is also allowed to transmit a new data packet on receiving 2 duplicate 105 ACKs. While the transmitter is in the Fast Retransmission Phase, it continues to retransmit packets using 106 Slow Start until all packets have been recovered (without starting a new April retransmission phase for partial 107 ACKs). Although this modification may cause unnecessary retransmissions, it reduces transmitter timeouts and 108 efficiently recovers multiple packet loss using partial ACKs. b) TCP Vegas TCP Vegas (Brakmo & Peterson, 109 1995) does not involve any changes to the TCP specification; it is merely an alternative implementation that is 110 able to interoperate with any other valid implementation of TCP. In fact, all the changes pare confined to the 111 sending side. There are some techniques that Vegas employs to increase throughput and decrease losses. 112

It calculates the RTT more accurately. Using an accurate RTT estimate serves two purposes. First, it 113 leads to a more accurate timeout calculation. Second, it is used in conjunction with the mechanism described 114 next to decide to retransmit a dropped segment in a more timely fashion. It includes the new mechanism for 115 deciding to retransmit Vegas treats the receipt of certain ACKs as a trigger to check if a timeout should happen. 116 Moreover, Vegas only decreases the congestion window if the retransmitted segment was previously sent after 117 the last decrease. Any losses that happened before the last window decrease do not imply that the network is 118 119 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 120 throughput by: Expected = WindowSize / BaseRTT where WindowSize is the size of the current congestion 121 window and BaseRTT is the RTT of a segment when the connection is not congested. It then calculates the 122 current Actual throughput of the network. Then, Vegas compares Actual to Expected, and adjusts the window 123 accordingly. 124

### <sup>125</sup> 5 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

128 object is generally performed automatically by a library call.

### <sup>129</sup> 6 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.

### 135 7 IV. SIMULATION MODEL & DESIGN

To design the Mobile IP wired-cum-wireless network topology we have considered various simulation parameters, 136 which are supported by the network simulator-2 (NS-2) (Fall & Varadhan, 2005). We have done simulation for 137 two different topologies and call them topology 'A' and topology 'B' respectively. There is a roaming mobile node 138 called Mobile Host (MH) that moves between its home agent and foreign agent in a certain velocity. Data will 139 be exchanged (50,100), (200,300) and (500,300) respectively in the 700mX800m topology. In this network, the 140 mobile node MH starts moving from position (50,100) at 10 sec towards destination (650,100) at fixed speeds (0, 100) at (0,141 5, 10, 15, 20, 25, or 30) m/s. After reaching the destination position (650,100), MH will stay there 1sec (pause 142 time) and then again moves back to its initial starting position (50,100) at the same velocity. In this way, the 143 mobile node MH changes its home agent (HA) and foreign agent (FA) several times depending upon its speed. 144 We have used DSDV ad hoc routing protocol. FTP begins transferring packets of size 1000 bytes after 10 second. 145 Each wireless node has a buffer size of 50 packets. 146 Each simulation ends at 150 seconds. 147

There are fewer changes in Topology 'B'. We have used three base-station nodes in topology 'B' and call 148 them Home Agent (HA), Foreign Agent-1 (FA1) and Foreign Agent-2 (FA2) respectively. The initial positions, 149 (x,y), of the nodes MH, HA, FA1 and FA2 are (50,100), (200,300), (500,300) and (800,300) respectively in the 150 1000mX800m topology. In this case, simulation starts at 10 sec and ends at 190 sec. Here the mobile node MH 151 starts moving from position (50,100) at 10 sec towards destination position (950,100) at a fixed speed of (0, 5, 5, 5)152 10, 15, 20, 25, or 30) m/s while nodes HA, FA1 and FA2 serve as base station (gateway) nodes. As before, the 153 pause time is 1 sec, that is, the mobile node MH will stay 1 sec at position (950,100) and then moves back to 154 its initial position (50, 100) at the same velocity. Staying there 1 sec, it moves again towards the destination 155 position. 156

### <sup>157</sup> 8 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.

#### <sup>161</sup> 9 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 165 throughput to the whole simulation time. It is common that if the window size increases, the throughput will also 166 increase. Among the four variants, Vegas to Delayed-Ack shows the best throughput. From figure 3 (a), it is seen 167 that the throughput of each of the TCP variant decreases drastically. By comparing this figure to figure 2, it is 168 observed that the throughput performance is worse in this case. From the start of the simulation to 35 seconds, 169 throughput is the same (almost 700 kbps) as before (figure 2). But after 35 seconds it shows different result. 170 The mobility of the mobile host is the cause of this degradation. In the case of figure 2, the mobile host remains 171 stationary. But in this case the mobile host changes its base station for a number of times during simulation 172 time. When changing base station a mobile host requires a handoff. The mobile host changes base station three 173 times; so require three handoffs. And at the time of each handoff packets are dropped and throughput decreases. 174 Figure 3(b) shows the similar type of result except the throughput is reduced more times than that of figure 175 3(a). Since in this case, the mobility is 30 meter/sec, more handoff is required and at the time of each handoff 176 throughput is reduced. One interesting feature which is observed from figures 3(a) and 3(b) is that during 177 first handoff throughput performance decreases drastically but during the second, third and successive handoffs 178 throughput decreases slightly. But in such an environment Vegas to Delayed-Ack performs well among the TCP 179 variants. Its throughput reduces less than that of the others. Concentrating on the receiver portion in figure 2, 180 3(a) and 3(b), we see that the performance of the Base TCP receivers is lower than that of the Delayed-Ack TCP 181 receivers; no matter whether the TCP sender is New-Reno or Vegas. Since all the terminals in Mobile IP network 182 share the same radio channel to send and receive packets, the collision of data packets and acknowledgement 183 (ACK) packets in the radio channel can severely reduce TCP throughput. But Delayed-Ack receiver employs 184 185 less ACK packets. So, for transferring TCP data packets, TCP sender can utilize the channel more. Hence 186 throughput of any TCP sender synchronized with Delayed-Ack receiver is better than that of TCP sender with 187 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 188 sender performs well. As stated earlier, TCP Vegas estimates the network status to adjust its window value. 189 This property enriches its performance in such a heterogeneous environment. 190

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 196 with New-Reno to Delayed-Ack TCP and Vegas to Delayed-Ack TCP in both topology 'A' and topology 'B'. 197 It is due to the fact that ACKs and TCP packets share the same channel; so decreasing the ACKs by using 198 the Delayed-Ack receiver instead of Base receiver it is possible to improve the throughput of TCP. The delay 199 performance of Vegas to Delayed-Ack TCP and Vegas to Base TCP is relatively similar and they show low delay 200 in all the mobility so they show relatively similar and high packet delivery ratio in both the topologies. Since the 201 average delay of Vegas to Delayed-Ack TCP is low it has given a very good throughput in the simulation time in 202 both topologies. 203

#### <sup>204</sup> 10 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. 205 To achieve good TCP performance in Mobile IP wired-cum-wireless networks a variety of solutions have been 206 proposed and in most of the solutions the mobile node acts as a TCP receiver. However, little attention 207 has been paid to study the performance of TCP traffic over combining both TCP senders and receivers. We 208 have investigated the performance of TCP over combining both TCP senders (Newreno & Vegas) and TCP 209 receivers (Base & Delayed-Ack) using simulations in ns-2 for a range of node mobility with a single traffic source. 210 The performance metrics that we have considered include TCP throughput and packet delay. On throughput 211 performance, Vegas TCP to Delayed-Ack gains the best performance. On the sender side, the effective congestion 212 control technique of Vegas makes it optimal in such a heterogeneous environment. And in the receiver side, the 213 214 Delayed-Ack receiver makes the proper use of channel by sending fewer ACK packets.

215 We have also evaluated throughput with respect to node mobility. The result shows that throughput of each 216 TCP variant varies similarly with respect to each other against mobility of node. But Vegas to Delayed-Ack 217 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 218 large delay. But among them New-Reno to Delayed-Ack has the highest delay. After observing packet delivery 219 ratio and delay, it is seen that the variant that gains highest packet delivery ratio has lowest delay. Though all 220 the variants of TCP suffer in Mobile IP wiredcum-wireless networks, it is clear from the various results that if we 221 use Vegas TCP as a source (sender) and Delayed-Ack as a sink (receiver), then this combination shows the best 222

- 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.  $1 \ 2 \ 3$



Figure 1: Figure 1 :

Newreno to Base Vegas to Base (kbps) Throughput

225

Newreno to Delayed-Ack Vegas to Delayed-Ack

[Note: 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-]

Figure 2:

 $^1 \odot$  2012 Global Journals Inc. (US) Global Journal of Computer Science and Technology Volume XII Issue IX Version I

 $<sup>^{2}</sup>$ © 2012 Global Journals Inc. (US)

 $<sup>^3 @</sup>$  2012 Global Journals Inc. (US)

- [Holland and Vaidya ()] 'Analysis of tcp performance over mobile ad hoc networks'. G Holland , N Vaidya .
   *Proceedings of ACM Mobicom*, (ACM Mobicom) 1999. p. .
- 228 [Balakrishnan and Katz ()] 'Explicit Loss Notification and Wireless Web Performance'. H Balakrishnan, R Katz
- Proceedings of IEEE Globecom Internet Mini-Conference, (IEEE Globecom Internet Mini-ConferenceSydney,
   Australia) 1998.
- [Bakre and Badrinath ()] 'I-TCP: Indirect TCP for Mobile Hosts'. A Bakre , B R Badrinath . Proceedings of
   the 15th International Conference on Distributed Computing Systems (ICDCS), (the 15th International
   Conference on Distributed Computing Systems (ICDCS)) 1995. p. .
- [Hoe ()] 'Improving the start-up behavior of a congestion control scheme for TCP'. J Hoe . ACM SIGCOMM, 1996.
- [Fall and Varadhan ()] *ns Notes and Documentation*, K Fall , K Varadhan . http://www-mash.cs.Berkeley.
   edu/ns/ 2005. 28. Lawrence Berkeley National Laboratory
- 238 [Perkins and Charles ()] E Perkins, Charles. IP Mobility Support. RFC, 1996. 2002.
- 239 [Perkins ()] C Perkins . IETF RFC 3344. IP mobility support for IPv4, 2002.
- [Brakmo and Peterson ()] 'TCP Vegas: End to End Congestion Avoidance on a Global Internet'. L S Brakmo ,
   L L Peterson . *IEEE Journal on Selected areas in Communications* 1995. 13 (8) p. .
- <sup>242</sup> [Chan and Ramjee ()] 'TCP/IP Performance over 3G Wireless Links with Rate and Delay variation'. M C Chan , R Ramjee . *Proceedings of MOBICOM*, (MOBICOM) 2002.
- [Hadjiefthymiades et al. ()] Using Path Prediction to Improve TCP Performance in Wireless/Mobile Communications, S Hadjiefthymiades, S Papayiannis, L Merakos. 2002. IEEE Communication Magazine.
- 246 [Chockalingam et al. ()] 'Wireless TCP Performance with Link Layer FEC/ARQ'. A Chockalingam , A Zorzi ,
- 247 V Tralli . Proceedings of IEEE ICC'99, (IEEE ICC'99) 1999. p. .