Evaluating the Performance of Transport Protocols over MMP

Ana Victoria Delgado Martín, Andrej Mihailovic, Nikos Georganopoulos, A. H. Aghvami

Abstract: *TCP is a reliable connection-oriented transport protocol that performs well in traditional networks. However, in networks with wireless and other lossy links the protocol suffers from losses and delays due to frequent handoffs like in wireless networks running Mobile IP. Many Micromobility Protocols have been proposed to reduce handoff latency and the load on the network when we move among small wireless cells. In this paper, the performance of several TCP and UDP schemes over one of these protocols: MMP (Multicast for Mobility Protocol) is compared and different improvements to TCP in a multiple handoff network are proposed. The simulation results are presented using the Network Simulator (ns2) and different metrics for comparison. Our results show that the number of UDP packets lost is reduced when using MMP instead of Mobile IP and that TCP provides better throughput performance for a traditional scheme like Tahoe TCP.*

1. Introduction.

Current trends in wireless communications have shown that future internetworks will include large number of portable devices moving among small wireless cells. That is why an IP-Micromobility Protocol and a reliable transport protocol such as TCP play an important role in the operation of Internet.

Although an IP-Micromobility System such as MMP [1] (Multicast for Mobility Protocol) helps to decrease packet losses or delay during handoffs, other modifications in higher layers of the hierarchy are also needed to improve performance in the overall transmission. These modifications affect primarily the transport protocols and the way establishment, connection and transmission are brought about.

In theory, transport protocols should be independent of the technology of the underlying network layer. In particular, TCP should not care whether IP is running over fibre or over radio. In practice, it does matter because most TCP implementations have been carefully optimised based on assumptions that are true for wired networks but which fail for wireless ones. This results in a poorer performance than the one expected so it is necessary to introduce some modifications: while in a wired network when a packet is lost the sender should slow down, in a wireless network, the sender should try harder.

Although UDP does not suffer from the same problems as TCP, wireless communication also introduces difficulties for it. The main problem is that applications use UDP expecting it to be highly reliable. Although no guarantees are given, it is still expected to be near perfect. In a wireless environment, it will be far from perfect. For applications that are able to recover from lost UDP messages but only at a considerable cost, suddenly going from an environment where messages are rarely lost to one in which they are constantly lost can result in a performance disaster.

In this paper UDP and TCP are analysed over MMP and the possible effects of handoff losses in those implementations are studied. Attention is specially focused on TCP because its congestion control and transmission policy affect a connection with multiple handoffs in a particular way. In the following sections the tools used to obtain simulation results in *ns2* are described. Discussion and future work are eventually presented.

2. Multicast for Mobility Protocol (MMP): Architecture and Implementation.

Although many IP-Micromobility Protocols have been proposed to reduce delay and packet losses during handoffs, MMP has been chosen as the protocol to run the simulations due to the good results obtained in [1]. The details of the network implementation in *ns2* and how MMP works are described.

As presented in [1], MMP uses a sparse mode multicast routing protocol Core Base Trees (CBT) to handle the movement of mobile nodes within a foreign network. This scheme uses a shared-tree, to and from a centre point called the core of the network. MMP relies on Mobile IP Agent Discovery procedure in order for mobile hosts to discover relevant Mobility Agents and obtain a multicast care-of-address. The base stations will transmit periodic beacons with *Agent Advertisement* messages including a multicast care-of-address. After acquiring the care-of-address, the mobile host transmits a *Registration Request* to the base station. The base station will send a *Join Request* to the Core and create a permanent group. When the mobile host moves to another cell, it will initiate handoff when it receives a stronger beacon from another base station and it will include in its *Registration Request* the care-of-address of the group it belongs to, so the base station can extract the multicast address and send a *Join Request* to the core to join the group already established.

The test network used in *ns2* is shown in figure 1. The simulation starts when the mobile host is in cell 1 and moves from one base station to the other until it reaches cell 12. The cell size is 30 metres and the speed of the mobile host is 0.83m/s. Internet and foreign network data rates are set to 7Mbps and 2.5Mbps respectively. Point to point links have a delay of 10ms for all the links except for the links between the base stations and their correspondent router where the delay is 1ms.

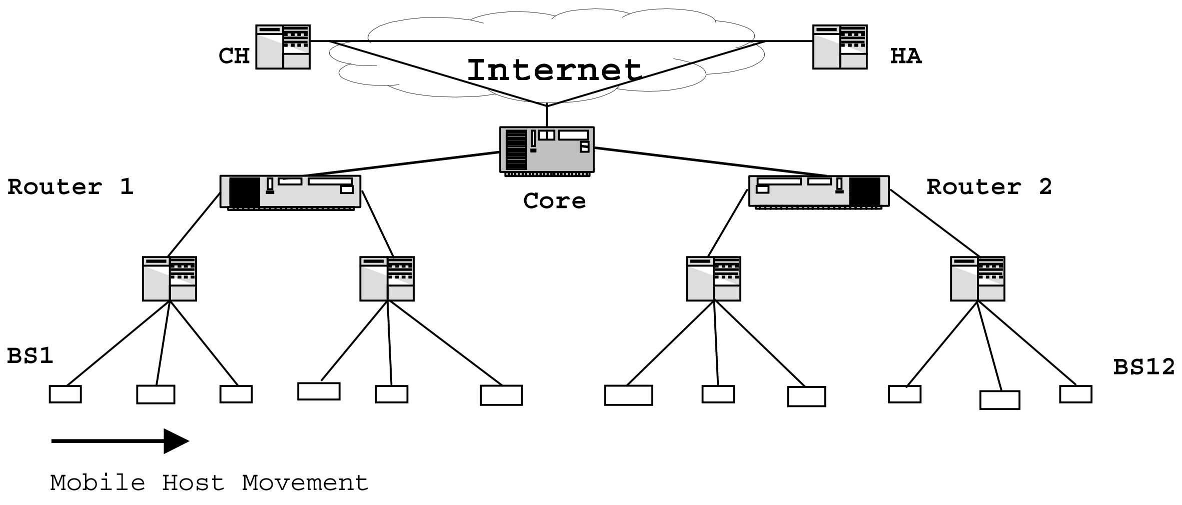


Figure 1: Testbed Configuration. HA stands for Home Agent, CH for Correspondent Host, BS for Base Station

3. UDP Performance.

UDP [2] traffic was used as the traffic load generated by the Corresponding Host with throughputs between 1024Kbits/s and 2512Kbits/s for a packet size of 64 bytes and a constant and exponential traffic generation.

The results show that packet losses are insignificant during handoffs even for the worst case when the handoff distance, that is, the number of hops a *CBT Join Request* traverses before an old on-tree router is reached, is three. The exponential character of the graphic demonstrates the good performance of MMP until the maximum throughput of the links is reached. The average packet losses in the case of exponential traffic mode shows the same results as for the constant model hence the extreme cases of maximum number of lost packets are considered.

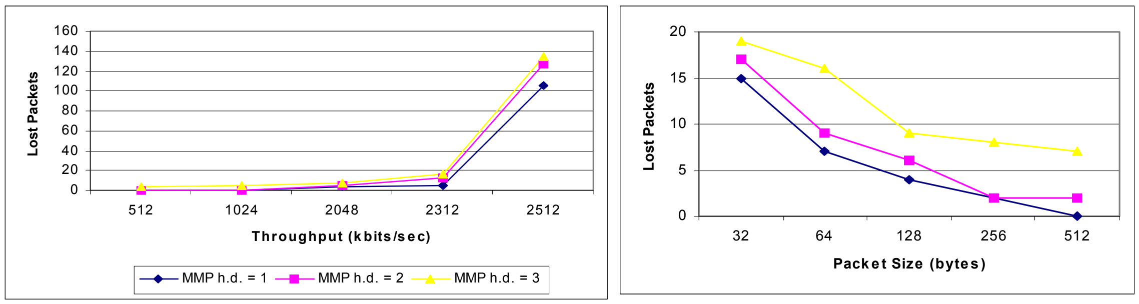


Figure 2: Lost packets for CBR Traffic Model. For constant packet size of 64 bytes and constant throughput of 2312Kbits/s.

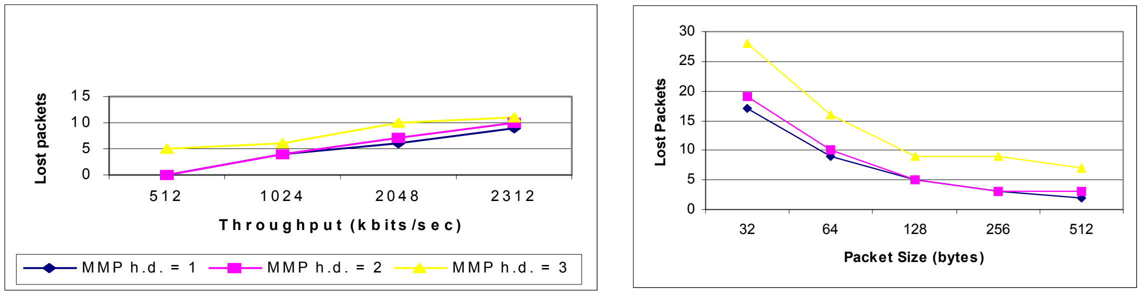


Figure 3: Lost packets for Expoo Traffic Model. For constant packet size of 64 bytes and constant throughput of 2312Kbit/s.

With a fixed throughput of 2312Kbits/s for both constant and exponential traffic generation, when a smaller packet size is used, the packet stream is more dense so the number of packets that are lost in a handoff for different handoff distances rises.

As it will be demonstrated in the next section, in contrast to TCP, UDP does not react to packet losses and thus often achieves a higher throughput than TCP. It never performs backoff and always sends data when transmission is possible, that is why, UDP presents the optimum throughput a TCP connection could achieve, if additional measures were taken to prevent TCP from backing off in case of non-congestion related packet loss.

4. TCP Performance.

TCP (Transmission Control Protocol) was formally defined in RFC 793 [3], after that, some bugs were detected and new versions of the protocol were introduced [4, 5, 6, 7, 8, 9,10]. In this section the performance of these versions in a micromobility environment are analysed.

To study the performance of TCP over MMP, the distribution of the sequence numbers of the packets received, the variation in the congestion window and in the estimation of the round trip time against simulation time are considered.

Figure 4 shows the typical throughput of a TCP connection in a micromobility environment, the "dents" in the throughput are caused by packet losses when a handoff occurs: TCP increases its rate until it reaches the maximum bandwidth of the link, experiences a loss, and then backs off again. It can also be appreciated that when the mobile host is moving from Base Station 6 to 7 (the 6th handoff), the "dent" is more abrupt because more packets are lost (handoff distance = 3)and the transmission is more affected.

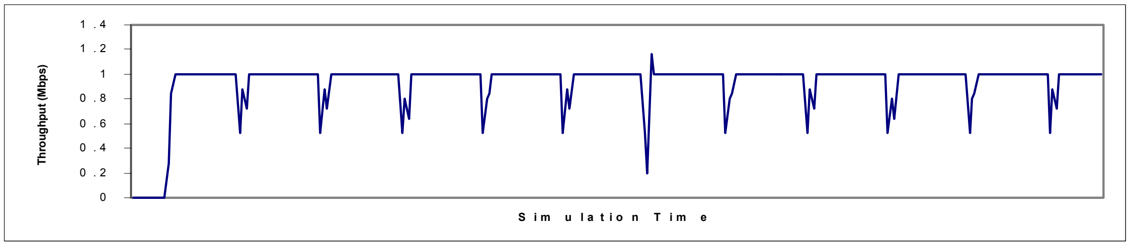


Figure 4: TCP Throughput versus simulation time

The performance of the various end-to-end protocols is summarised in figure 5 for the last seconds of the simulation. Those TCP schemes that reach a higher sequence number at the end of the simulation will also present a better throughput performance. Every sample represents a packet sent, so two samples with the same sequence number indicate a retransmit.

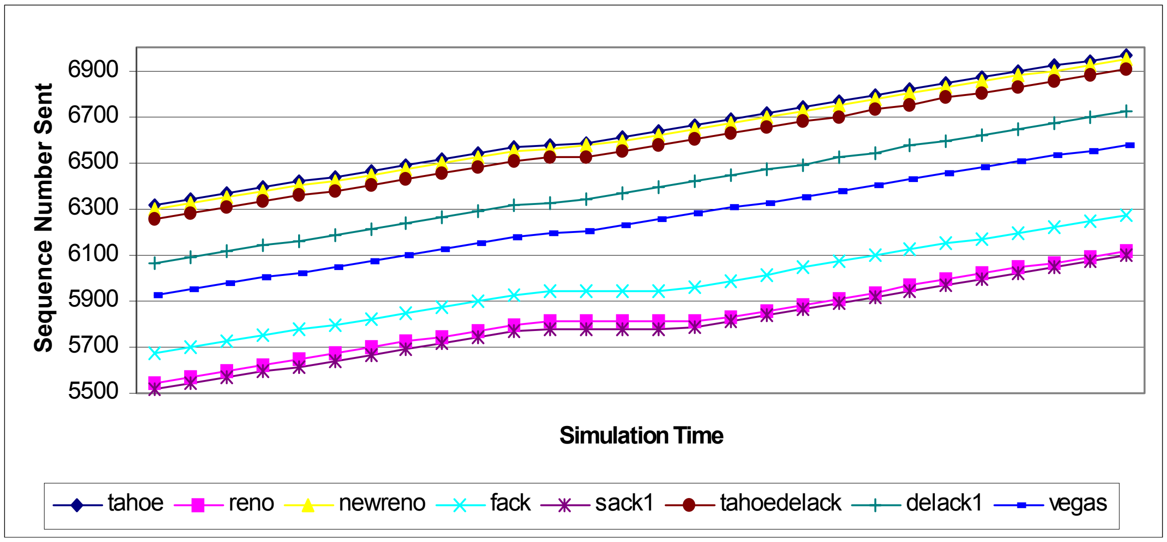
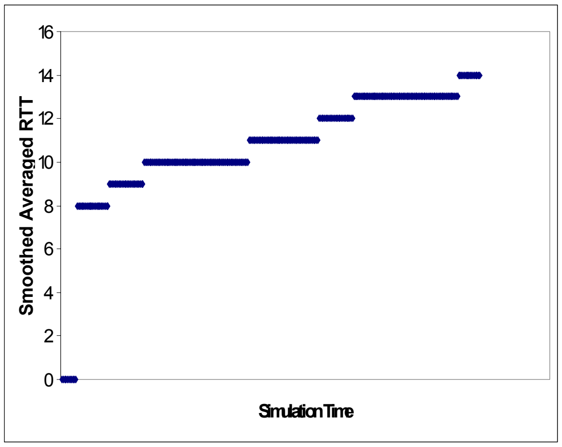
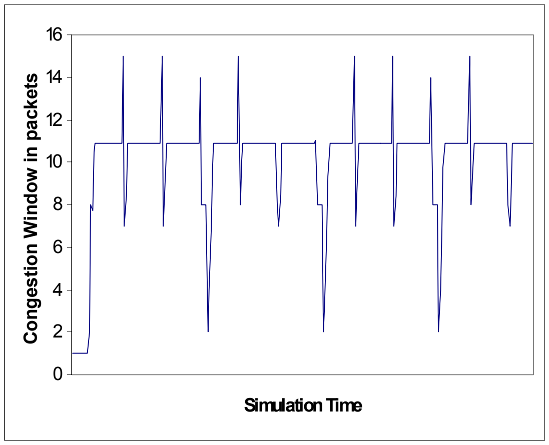


Figure 5: Sequence Number Sent versus Simulation Time for the last seconds of the simulation: Tahoe as defined in [9], Reno as in [4], New Reno as in [5], Forward Acknowledgement as in [7], Selective Acknowledgement as in [6], Tahoe TCP with Configurable Delay per ACK (tahoedelack) as in [11], Sack TCP with Configurable Delay per ACK (delack1) as in [11] and Vegas as in [10].

The desirable behaviour on this graph would be a smooth line of dots extending diagonally from the lower left to the upper right, that is, the sharpest the slope is the better performance is indicated. The flat lines in the middle of the graph show the retransmission of the same packet in different simulation times; this happens when a handoff occurs, packets are lost and it is necessary to retransmit them.

The next round of simulations was focused on the measure of the smoothed averaged Round Trip Time and the size of the congestion window. As we can see in figures 6 and 7, the estimation of the Round Trip Time will be an important factor to





Figures 6 and 7: Smoothed Averaged Round Trip Time and Congestion Window in packets for Vegas TCP.

seize the congestion window. The greater the variation is in the estimation of the Round Trip Time, the greater the variation is in the size of the congestion window. Although this parameters are not directly related, we can assume that an unstable network will change constantly its estimations and therefore it will result in an unstable performance. We have chosen a particular case, the Vegas TCP connection where the congestion window and the Round Trip Time vary resulting in a medium performance with multiple handoffs. Similar results were obtained for other TCP schemes.

Figure 8 will show the differences in throughput for the TCP schemes. It can be concluded that the normal mechanism for congestion avoidance used in Tahoe TCP [9] results in a better performance reaching the best throughput transmission. The rest of the schemes that were posterior modifications of TCP [4,5,6,7,8,9,10] seem to be not very appropriate for a micromobility environment. This is due to the fact that TCP Tahoe attacks the lost of a packet calling a congestion control algorithm that adapts “slowly” to the new situation recovering “quickly” from the lost. The posterior modifications of other algorithms can improve performance in a wired network but they are not appropriate for a wireless micromobility environment.

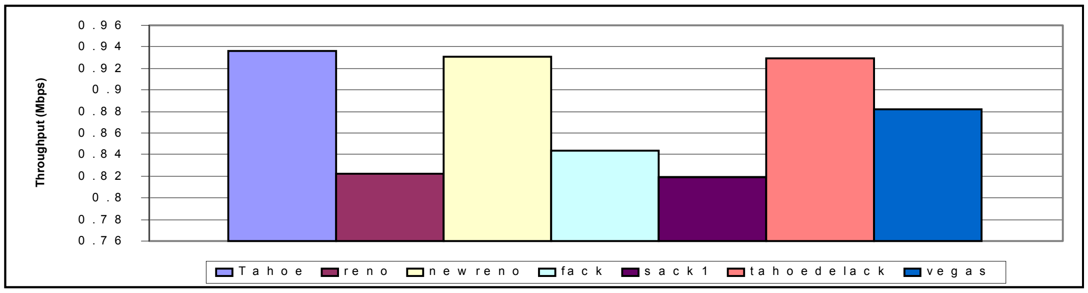


Figure 8: Average Throughput for different TCP connections.

5. Conclusion and future work.

In this paper the simulation results of UDP and different kinds of TCP over MMP (Multicast for Micromobility Protocol) using *ns2* have been presented.

The study has shown that the performance of UDP is much better in a micromobility environment when we get rid of the frequent location updates needed in Mobile IP.

Tahoe TCP with its retransmission policy has been shown to be the protocol that better works in an environment of multiple handoffs whereas other schemes that perform well for a wired network present a poor performance in a micromobility environment.

Our next study will be focused on the fact that some modifications introduced to TCP can result in an improvement in the overall performance in conjunction with the small amount of packets lost when using MMP.

References

[1] A. Mihailovic, M. Shabeer, A. H. Aghvami, "Multicast for Mobility Protocol (MMP) for emerging internet networks"
[2] "User Datagram Protocol", J. Postel. RFC 768. August 1980.
[3] Defense Advanced Research Projects Agency. RFC, RFC 793, Sept. 1981.
[4] W. Stevens. " TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, January 1997.
[5] S. Floyd and T. Henderson. "The NewReno Modification to TCP's Fast Recovery Algorithm". RFC, RFC2582, April 1999.
[6] M. Mathis, J. Mahdavi, S. Floyd and A. Romanov. "TCP Selective Acknowledgement Options". RFC, RFC 2018, October 1996.
[7] M. Mathis and J. Mahdavi. "Forward Acknowledgment: Refining TCP Congestion Control", in ACM SIGCOM, August 1996.
[8] H. Balakrishnan and V. Padmanabhan. "A comparison of Mechanisms for improving TCP performance over Wireless Links", in ACM SIGCOM, August 1996.
[9] V. Jacobson. "Congestion Avoidance and Control", in ACM SIGCOM, CA, USA, August 1988.
[10] L.S. Brakmo, S.W. O'Malley, L.L. Peterson. "TCP Vegas: New Techniques for Congestion Detection and Avoidance", University of Arizona.
[11] K.Fall and K. Varadhan. "ns Notes and Documentation". October 1999.