

TCP over Micro Mobility Protocols: A Systematic Ripple Effect Analysis

Ganesh Bhaskara, Ahmed Helmy

Department of Electrical Engineering, University of Southern California
3740 McClintock Ave, EEB 232, Los Angeles, CA 90089-2562

{bhaskara@usc.edu, helmy@ceng.usc.edu}, Tel: (213) 740-2685, Fax: (213) 740-7290

Abstract-- With the increasing popularity of powerful handheld / mobile computing devices and ubiquitous availability of wireless connectivity, protocols that support mobility are becoming increasingly important. It is critically important to explore and understand the design space, so that appropriate mechanisms are incorporated into the design to provide the required performance under varied environments and services provided by the lower layers. The mechanisms incorporated into the design may interact in subtle ways with the mechanisms of the higher layer to produce unintended effects. In this paper, we study how the different mechanisms of the lower layers (IP and MAC) that support mobility affect the performance of Transmission Control Protocol (TCP). We use the building block framework to capture and study the wide variety in the IP layer handover optimization mechanisms. We use detailed 802.11 models to study the effect of MAC layer mobility support on TCP. We show by simulation how different mechanisms affect TCP differently in different scenarios. We also show results that very counter intuitive, like buffering packets does not always improve TCP throughput and TCP-Tahoe performs better than its more sophisticated counterparts.

Index Terms—Micro Mobility, Building Block, Handover Performance, Ripple Effect, Inter layer interaction

I. INTRODUCTION

With the advent of the web technology and its associated services, TCP [5] has become the most popular transport protocol in the Internet today. All the transport protocols, including TCP need to work seamlessly in scenarios where one or both the end-hosts are mobile. To overcome the acute problem of frequent and large handover delays of Mobile IP (MIP)[1], micro mobility protocols like Cellular IP (CIP)[2], Handoff Aware Wireless Architecture for Internet Infrastructure (HAWAII) [3],[4] and Multicast based Micro-mobility Protocol (M&M)[7] were designed. In this paper, we analyze the effects of the variation of parameters of the mechanisms of the lower layers like MAC and IP, that support user mobility, (ripple effects, Fig.1) on higher layer protocols, TCP [11] in particular. In section II we motivate and differentiate our work from the existing work. Also, Section II touches upon the various handover optimization mechanisms. In section III, we present our simulation results, followed by inferences and conclusion.

This work was supported by NSF HDCCSR, NSF Career, Intel and Pratt&Whitney PWICE Institute.

II. MOTIVATION AND RELATED WORK

A. Motivation

Depending on the mechanism the micro mobility protocols use, each version of the protocol introduces packet duplication, loss, reordering, delay and jitter into the packet delivery stream. Also, the capabilities (make before break and vice versa) of the underlying radio layer plays a significant role in determining packet delivery properties. Transport protocols like TCP are adversely affected when such variations are introduced into the packet delivery stream. This study focuses on the interaction of TCP with the mechanisms of micro mobility protocols and MAC/Radio layers.

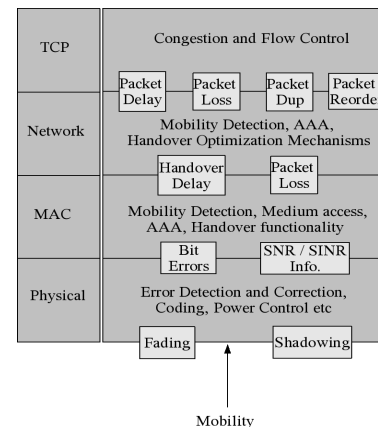


Fig. 1. Ripple Effects

Earlier study [6] [7] and [8] analyzes in detail the effects of adjusting the parameters of the micro mobility protocols in the building block framework (in terms of packet loss, duplication and delays) on the UDP traffic. It has been shown that buffering packets decreases packet loss in scenarios where handover delay is large, and proactive handover schemes perform better than reactive handover schemes in terms of packet delay, jitter and loss characteristics. However, the effect of such variations in performance at the micro-mobility layer on TCP has not been studied systematically before, and is not very well understood. Hence, we believe it is important to systematically analyze the ripple effects (Fig. 1) of the mechanisms that support mobility at the lower layers on the performance of transport protocols like TCP so that we can get guidelines to modify / add new mechanisms to the transport protocols or the handover optimization mechanisms or both.

We use a novel Building Block based design and analysis [6] methodology to systematically study the performance of TCP during handover. This is a generic framework that enables the investigation of multiple variants of micro mobility protocols (CIP, HAWAII and M&M) by changing the building blocks and its parameters. In this paper we take the approach of analyzing the performance of TCP over various handover optimization mechanisms rather than TCP over micro-mobility protocols.

B. Related work

There exists some literature [9] [10] that deals with the performance of TCP over micro-mobility protocols. Our work differs from the existing work in the approach we take i.e. we analyze the performance of TCP using the building block approach. This allows us to isolate the effects of the building blocks of the protocols on the performance of TCP. Secondly, no literature analyzes the performance of TCP over M&M and its associated handover optimization mechanisms. Here, we address these issues. Further, unlike the existing work, we use more detailed simulation scenarios like MAC layer handover delay and gaps in radio coverage to evaluate the performance of TCP. We show that this significantly alters the performance of the handover optimization mechanisms. Significant work has been done on the topic of TCP over wireless medium [13] [14] [15]. We do not address this topic. We also do not assume or use any sort of split TCP protocol. Snoop type of protocols [16] may not work very well during handover, especially when the MN is receiving packets from more than one BS where as the acks are being sent through only one BS.

C. Handover Mechanisms

When the MN moves from the coverage of a Base Station (BS) / Access Point (AP) in one sub-domain (through an Access Router (AR) in that sub-domain) to a BS in another sub-domain, handover needs to take place so that the incoming packets are directed towards the new AR. The handover process involves mobility detection at the MAC layer and IP layer (IPMD), Authentication Authorization and Accounting (AAA) functions with the new BS and AR, route repair and

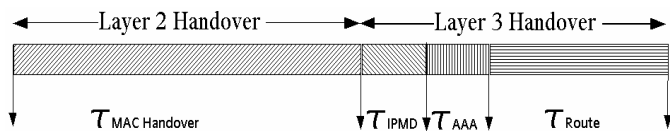


Fig. 2 Traditional Handover Timing Diagram

handover optimization to minimize one or more of packet loss, packet duplication, delay and jitter.

The Layer2 handovers can be classified into hard handover and soft handover. In hard handover, the MN loses the connection to its old BS as it initiates the handover to the new BS. This results in long handover delays. Layer 3 handover occurs after

the Layer 2 handover. In soft handover, the MN continues to receive packets from the old base station as it initiates and establishes a new connection with another BS. The MN is connected to both the old and the new BS [18] for a brief duration. Layer 3 handover mechanisms can be classified into reactive and proactive schemes. In the reactive handover scenario, Layer 3 handover is initiated in response to a Layer 2 handover. Reactive handover usually occurs in scenarios where the MN can communicate with only one BS at a given time or when gaps in radio coverage / fading and other channel conditions render the MN / BS unreachable for a brief duration. The MN or the old BS / old AR does not have a priori information about the new BS / AR to which the MN hands over to. If there exists no Layer 3 handover optimization mechanisms, heavy packet loss occurs. MSF and SSF of HAWAII are reactive handover optimization schemes that buffer packets at the old BS / AR once they notice that the MN is out of its range (of course with the assumption that the MAC/radio layers have mechanisms to track the MN connectivity). Bi-cast handover optimization is reactive in nature and works well with soft handover at Layer 2 (scenarios where the MN can communicate with more than one BS at the same time).

In the proactive handover optimization scheme, the identity of the AR or the set of potential ARs to which the MN is handing over to, is known before the connection between the old AR and the MN goes down (as with soft Layer 2 handover). This a priori knowledge can be used to initiate handover optimization mechanisms (setting up routes to the new AR, tunneling packet from old AR to new AR etc.) to reduce packet loss and delay. Another way of achieving proactive handover in scenarios where Layer 2 only supports hard handover (where the MN can communicate with only one BS / AR at a given time), is using the Candidate Access Router set (CAR-set) mechanism as described in [7]. This involves predicting the new AR / potential set of ARs (CAR-Set) to which the MN may handover to. By virtue of being multicast, M&M can deliver packets to multiple (2 or more) ARs simultaneously. The time at which the packet delivery to the CAR-Set is initiated is governed by the type of triggers that are used to initiate packet delivery to the CAR-Set. *Source Trigger* is generated at the old AR prior to the disconnection of the MN (assuming Layer 2 can detect the disconnection). Here the source trigger contains the identification of the AR to which the MN is handing over to. A *Path Trigger* is generated by the serving AR prior to the disconnection of the MN. It contains a list of potential ARs in the CAR-Set to which the MN may handover to. An extreme case of the path trigger is the *no-path trigger* in which the potential list of AR is the entire CAR-Set. The delivered packets are cached at the ARs and delivered to the MN once it joins to the multicast group at that AR. This results in the wireless bandwidth being conserved, while at the same time reduces packet loss and packet delay at the cost of increased usage of bandwidth on the wired links.

III. PERFORMANCE EVALUATION AND ANALYSIS

We evaluate the performance of TCP-Tahoe, TCP-Reno, TCP-NewReno and TCP-SACK over both reactive and proactive handover mechanisms in both hard and soft handover scenarios. We use TCP throughput and the number of timeouts TCP suffers, as metrics to evaluate the ripple effects. By using the building block framework and by varying the CAR-set parameters like buffer size (zero for no buffering and some finite number for buffering), the trigger set (zero, one or many), we can emulate almost all the handover optimization mechanisms of CIP (hard and semi-soft handover), HAWAII (UNF and MNF) and M&M (bi-cast and CAR-set). In the following sections, we introduce the simulation setup followed by the simulation results and its interpretation.

A. Simulation Setup

A packet level simulator of the micro-mobility framework was implemented in ns-2.26 [17]. Simulations of length of 100s averaged over 10 runs were conducted on a tree topology with 21 base stations and 12 wired nodes, with a maximum depth of 5. The MN incurred 20 handovers per simulation run. The Layer 2 handover delay (not including Layer 3 mobility detection and handover delay) was varied from 500ms to 0ms [19] in the case of hard handover. The duration of MN receiving packets from more than one BS was varied from 0ms to 500ms in the case of soft handover.

B. Performance Results and Analysis

B.1 Layer 2 Hard Handover

In this section we present the simulation results of TCP-NewReno, TCP-Reno, TCP-SACK and TCP-Tahoe. Following are the notations used in the figures:

- “-x”ms represents the Layer 2 hard handover delay
- “+x”ms represents the duration for which the MN receives

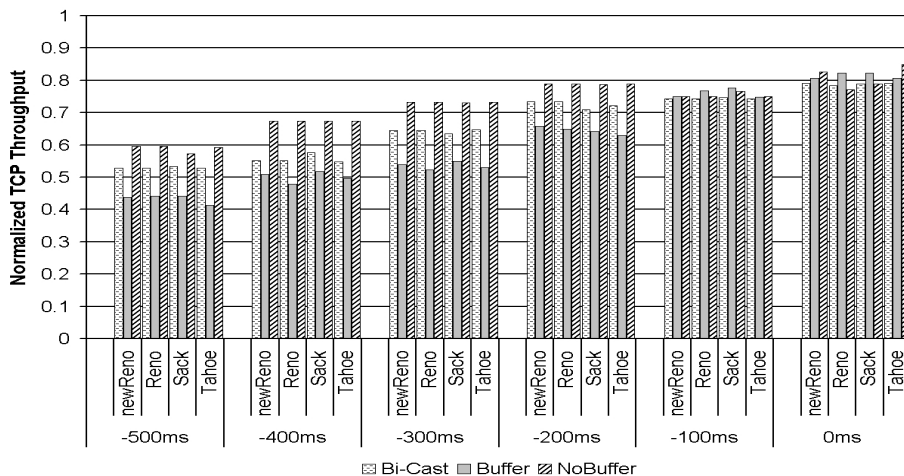


Fig. 3. TCP Throughput (Layer 2 - Hard Handover)

packets from more than one BS (old and new) as in the case of Layer 2 soft handover.

- “*Buffer*” refers to proactive handover optimization mechanism with buffering
- “*noBuffer*” refers to proactive handover optimization mechanism without buffering
- “*biCast*” refers to bicast handover optimization mechanism

Fig. 3 and Fig. 4 shows the normalized throughput and timeouts of the different versions of TCP in a hard handover scenario. We can observe the following from Fig.3 and Fig.4

- High Layer 2 handover delay
 - All versions of TCP have higher throughput with “*noBuffer*” and “*buffer*” performs consistently worse than the other two schemes.
 - The number of timeouts TCP suffers remains almost constant with “*noBuffer*”.
 - The number of timeouts TCP suffers decreases with decrease in Layer 2 handover delay with *Buffer* and *BiCast* handover optimization.
- Low Layer 2 handover delays
 - Most versions of TCP suffer substantially less number of timeouts (Fig 5, 100ms and 0ms) “*Buffer*” than with the other two (*biCast* and *noBuffer*) handover mechanisms.
 - In spite of the substantial differences in timeouts, the differences in throughput of TCP versions with the different handover mechanisms are not very pronounced when Layer 2 handover delays are small(-100ms and 0ms).

The primary reason for TCP with “*noBuffer*” to perform better than TCP with “*Buffer*” and “*biCast*” is that “*noBuffer*” minimizes the total handover delay and packet duplication at the cost of some packet loss. Once the TCP sender timesout, it can send packets and open its window without any interruptions from buffered duplicate packets. All versions of TCP with “*Buffer*” perform poorly when Layer 2 handover delay is large. Once the MN hands over to the new BS, it receives packets that were buffered at that BS / AR. But the sender would have timed out and resent the packets. The MN ends up receiving the old buffered packets and the newly retransmitted packets

leading to large number of duplicate Acks. This causes the sender to either timeout again or enter fast retransmit after cutting its congestion window. Thus TCP with “*Buffer*” performs badly when Layer 2 handover delays are large.

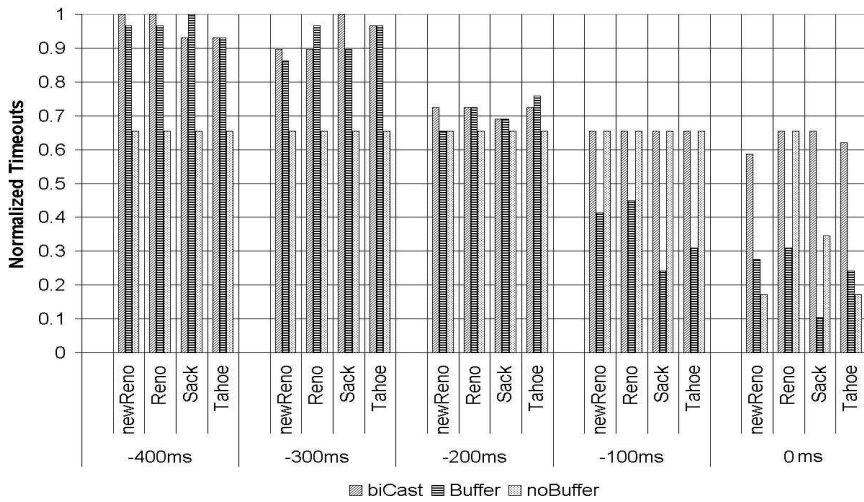


Fig. 4 Normalized TCP-Timeouts (Layer 2 - Hard Handover)

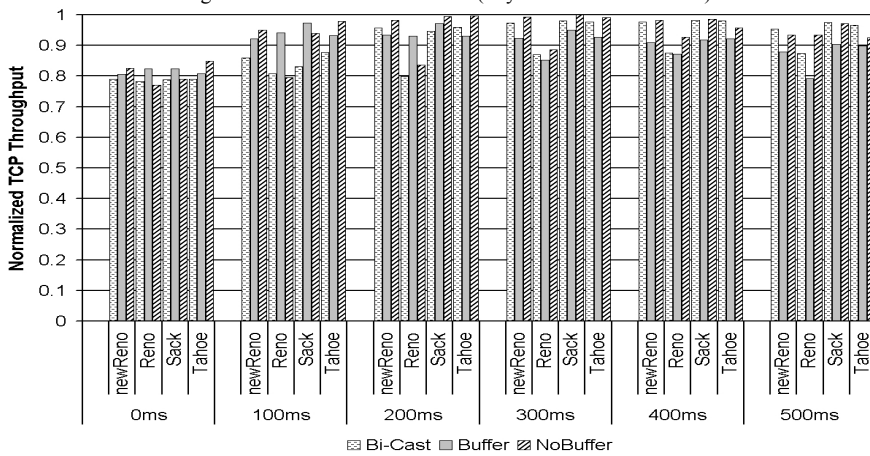


Fig. 5. TCP Throughput (Layer 2 - Soft handover)

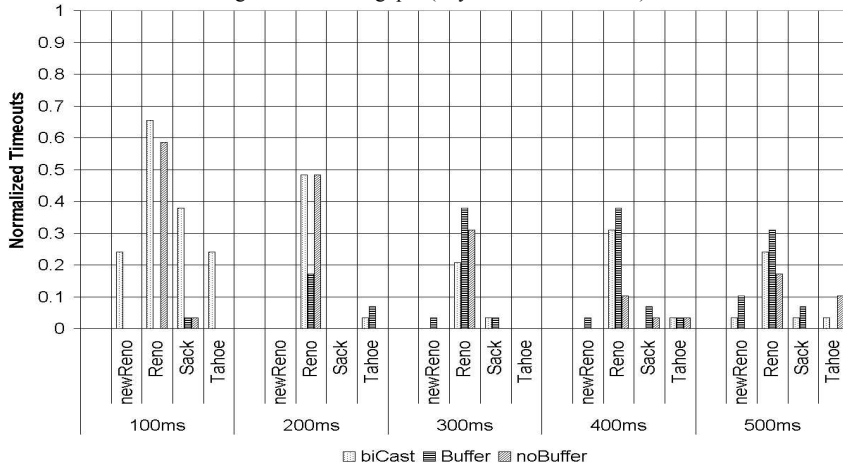


Fig. 6. Normalized TCP Timeouts (Layer 2 - Soft handover)

When the Layer 2 hard handover delay is small, none of the TCP versions over any of the handover optimization mechanisms seem to have an edge over the other. Also, all TCP versions suffer almost the same number of timeouts with “*biCast*” and “*noBuffer*”. In the case of “*noBuffer*”, the Layer 2 handover delay is not long enough to cause a timeout, but the packet loss is severe enough that the TCP timesout after receiving the packets from the new BS. Analyzing the congestion window traces of TCP-Reno at -100ms, we

observed that the TCP timeout delay adds to the Layer 2 handover delay, causing TCP to take longer to timeout. This results in what looks like an anomalous decrease in throughput of TCP with “*noBuffer*” at the -100ms point. Since “*Buffer*” reduces packet loss the decrease in throughput is not seen.

B.2 Layer 2 Soft Handover

Fig 5 and Fig 6 presents the normalized TCP throughput and normalized TCP timeouts with the various handover optimization mechanisms with soft Layer 2 handover. Following are the observations we can make from the figures.

- As expected, TCP tends to suffer fewer timeouts with soft Layer 2 handover than with hard layer 2 handover.
- Considering all versions of TCP, throughput tends to peak with “*Buffer*”, “*noBuffer*” and “*biCast*” in that order.
- When the MN receives packets from old BS and new BS for a short duration
 - TCP throughput tends to increase as the duration for which the MN receives packets from both the old and new BS increases (0ms to +300ms).
 - No handover optimization mechanism stands out in particular except for “*Buffer*” with TCP-Reno in 100ms and 200ms case.
 - TCP-Reno consistently under performs all other TCP versions with most handover optimization mechanisms
- When the MN receives packets from old and new BS for a long duration (> 300ms)
 - As this duration increases TCP throughput tends to decrease.

In the case soft Layer 2 handover, the MN continues to receive packets from the old BS, as it initiates route repair through the new AR. Thus MN rarely incurs packet loss. The major factors that affect the throughput of TCP are packet duplication and packet reordering and its ability to handle these effects. In the soft handover scenario, it is

more interesting to look at the performance of the individual TCP flavors with the different handover optimization mechanisms.

An important observation is the surprisingly competent performance of TCP-Tahoe. TCP-Tahoe is the least sophisticated version of TCP without additional mechanisms to handle multiple packet loss in a window. However, this very nature of TCP-Tahoe helps it to perform reasonably well. The only way it handles multiple duplicated and reordered packet /

window is by a timeout. As soon as sender receives multiple duplicate acks in a window, indicating an apparent loss of more than a packet / window, the sender does not wait for the retransmission timer to expire, but it immediately starts from slow start after setting ssthresh to half the congestion window. Even though TCP-SACK does not suffer as many timeouts as TCP-Tahoe, the difference in their through puts is very marginal.

IV. CONCLUSION

In this paper, we have presented the ripple effect study of mechanisms at the network and MAC layer that support node mobility on TCP. From the performance curves, it is clear that the different mechanisms that support mobility, affect TCP in different and unexpected ways in different scenarios. Following are observations from this study

- TCP with proactive handover mechanisms perform significantly better than reactive handover mechanisms in Layer 2 hard handover scenario
 - TCP performs better with “NoBuffer” when handover delays are expected to be high
 - TCP performs better with “Buffer” when the handover delays are expected to be small.
- In Layer 2 soft handover scenario,
 - TCP with “Buffer” is marginally better than others when the duration for which the MN receives packets from more than one BS is small.
 - When the duration for which the MN receives packets from more than one BS is large, TCP with “NoBuffer” and “BiCast” performs better than with “Buffer”
 - TCP-Reno suffers in this scenario as the repeated duplicate acks cause the sender to reduce its congestion window repeatedly.
 - TCP-Tahoe performs surprisingly better than TCP-reno even though it tends to suffer more time-outs as it is able to maintain a higher average congestion window than TCP-Reno.

However, some of the results are surprisingly counter intuitive, such as

- the dip in performance of TCP with “NoBuffer” at -100ms,
- bad performance of TCP with “Buffer” in scenarios where Layer 2 handover delays are large,
- TCP-Tahoe performing almost as good if not better than its sophisticated counter parts,
- More timeouts not necessarily leading to decrease in performance (TCP-Tahoe in the case of +100ms, +200m)

These results occur due to the subtle interaction of the mechanisms of TCP with the effects produced by the mechanisms of the lower layers designed to support node mobility. This study captures the subtle interaction of the various mechanisms and provides a good starting point for designers looking to design new transport protocols that are more robust to the varied packet delivery patterns (during

handover) or mobility protocols that better serve the needs of the existing transport protocols. As a part of future work, we plan to study the performance of SCTP [12] and TCP over additional handover mechanisms and formally model and quantify the interlayer interaction.

V. REFERENCES

- [1] C. Perkins, IP Mobility Support , RFC 2002, Internet Engineering Task Force, October 1996.
- [2] A.T. Campbell, J. Gomez, C.-Y. Wan, S. Kim, Z. Turanyi and A. Valko, Cellular IP, Internet draft, draft-ietf-mobileip-cellular-00-txt (December 1999).
- [3] Ramachandran Ramjee, Thomas La Porta, Sandy Thuel, Kannan Varadhan, Shie-Yuan Wang, HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks , International Conference on Network Protocols, ICNP'99
- [4] [4]A. T. Campbell and J. Gomez, "IP Micro-Mobility Protocols", ACM SIGMOBILE Mobile Computer and Communication Review (MC2R), 2001.
- [5] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", [RFC2581](#) , April 1999.
- [6] G. Bhaskara, A. Helmy, S. Gupta, "[Micro Mobility Protocol Design and Evaluation: A Parameterized Building Block Approach](#)", IEEE Vehicular Technology Conference (VTC), October 2003.
- [7] A. Helmy, M. Jaseemuddin, G. Bhaskara, "[Multicast-based Mobility: A Novel Architecture for Efficient Micro-Mobility](#)", ALL-IP WIRELESS NETWORKS, IEEE Journal on Selected Areas in Communications (JSAC), 1st Quarter 2004.
- [8] A. Helmy, M. Jaseemuddin, G. Bhaskara, "[Multicast-based Mobility: A Novel Architecture for Efficient Micro-Mobility](#)", ACM SIGCOMM Computer Communication Review (CCR), October 2002.
- [9] Fabio Vena, Llorenç Cerdà and Olga Casals, "Study of the TCP Dynamics over Wireless Networks with Micromobility Support Using the ns Simulator", European Wireless 2002, Florence, Italy, February 2002.
- [10] A. T. Campbell, Gomez, J., Kim, S., Turanyi, Z., Wan, C.-Y., A, Valko, "Comparison of IP Micro-Mobility Protocols", IEEE Wireless Communications Magazine, Vol. 9, No. 1, February 2002.
- [11] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", Computer Communication Review, pp. 5-21, Volume 26 No. 3, July 1996.
- [12] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, "Stream Control Transmission Protocol" RFC 2960, October 2000.
- [13] R. Zorzi, A. Chockalingam, R. R. Rao. "Throughput analysis of TCP on channels with memory", IEEE J-SAC, 18(7):1289-1300, 2000.
- [14] A. Chockalingam, M. Zorzi, R. R. Rao. "Performance of TCP on wireless fading links with memory", Proc. IEEE ICC'98, 595-600, 1998.
- [15] C. Parsa, J.J. Garcia-Luna-Aceves. "Differentiating congestion vs. random loss: a method for improving TCP performance over wireless links", Proc. IEEE WCNC'2000, 1:90-93, 2000
- [16] Sarma Vangala and Miguel A. Labrador. "Performance of TCP over Wireless Networks with the Snoop Protocol", In Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN 2002), November 2002.
- [17] Information Science Institute-University of Southern California, University of California-Berkeley, Xerox PARC, Network Simulator, ns-2 Home Page: <http://www.isi.edu/nsnam/ns> .
- [18] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, [A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks](#), Microsoft Technical Report, MSR-TR-2003-41, June 2003
- [19] A. Mishra, Min-ho Shin and William A. Arbaugh, "[An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process](#)", ACM SIGCOMM Computer Communication Review (ACM CCR), Volume 33, Issue 2, April 2003.