Proceedings of Asia Pacific Conference ObComAPC- 2004:
Parallel and Distributed Computing Technologies, December 13-15, 2004:
Vellore Institute of Technology, Vellore, TN, India:

PERFORMANCE ANALYSIS AND COMPARISON OF TCP IMPLEMENTATIONS OVER CELLULAR IP ACCESS NETWORKS

Shrikant K. Bodhe
Department of Electronics and Tele-communication Engineering ,
Terna Engineering College,
Navi-Mumbai-400 021, India.
Email: skbodhe@indiatimes.com
Phone: 0091-20-4011145
Mobile: 0091-9822052975

Fekri M. A. Abduljalil
Department of Computer Science,
University of Pune ,Ganeshkhind,
Pune 411 007, India
Email: fekri75@yahoo.com.
Phone: 0091-20-26526091.
Mobile: 0091-9890293316

ABSTRACT

TCP has been designed for reliable connection-oriented networks in which most packets losses occur primarily due to network congestion. With the introduce of IP based wireless networks TCP suffers from other type of loss and delay due to frequent handoff between access points. Cellular IP is an IP based micro-mobility protocol designed primarily to provide fast handoff and minimize the packet loss during frequent handoff. However the comparison of performance of TCP implementations over Cellular IP network has not yet extensively investigated. This paper includes investigation the performance of different TCP implementations (e.g, TCP Tahoe, TCP Reno, TCP Newreno , TCP Vegas, TCP SACK, and TCP FACK) over cellular IP access networks during the handoffs. It is found that TCP versions performance is hardly degraded due packets loss and handoff delay. Also, It is found that all TCP versions present same behavior during the handoff except TCP Vegas, which it is the worst of all.

Keywords: *Cellular IP, hard Handoff, semisoft handoff, TCP versions, TCP implementations.*

1. INTRODUCTION

TCP (Transmission Control Protocol) [1][2] was specifically designed to provide a reliable connection –oriented end-to-end byte stream over an unreliable internetwork. TCP is used by those applications needing reliable, connection-oriented transport service, e.g., web (HTTP), mail (SMTP), file transfer (FTP), and virtual terminal service (Telnet), USENET news (NNTP).

TCP is a collection of many algorithms and enhancements. These enhancements lead to different TCP versions (e.g, TCP Tahoe, TCP Reno, TCP Newreno, TCP Vegas, TCP Sack, and TCP Fack). The major difference between these versions is in the way of controlling the congestion and react for packet loss during the congestion. Many comparisons have been done between some of these versions in wire environment as in [14][15][16]. But to our knowledge that there is not any comparison between these versions during handoff in Micromobility environment in wireless networks.

Copyright© ObComAPC-2004

Cellular IP is one of the prominent solutions for micro-mobility, which support fast handoff and paging techniques. The protocol intended to provide local mobility. In Cellular IP, packet losses occur during the handoffs , which reduce the TCP performance.

Some analysis has been done for the Cellular IP during the handoff , and the performance of TCP and UDP have been evaluated as in [10][11].

To this end, in this work, we will clarify how these TCP variants behave in IP Micromobility environment in Cellular IP access networks by means of simulations and compare their performance.

The rest of this paper is organized as follows. Section 2 describes the TCP versions. Section 3 describes Cellualr IP and its handoff schemes. Section 4 describes Simulation Model (simulation environment and simulation results). Section 6 details our conclusions and future works.

2. TCP OVERVIEW

In this section the essential TCP versions ,which are used in analysis and comparison, are given.

2.1 TCP CONGESTION CONTROL ALGORITHMS

There are two indications of packet loss in TCP versions, which it is either timeout of the sent packets or the receipt of duplicate ACKs[2]. Most of the TCP versions differed in the reaction for the congestion.

TCP has four main different algorithms to handle the packet loss and control the congestion in the network. The four TCP algorithms are slow start, Congestion Avoidance, Fast Retransmit, and Fast Recovery [17]. The packet loss sign in these algorithms is either packet timeout or receiving duplicate ACKs[2].

2.2 TCP IMPLEMENTATIONS

TCP Tahoe [2]: TCP Tahoe uses Slow Start algorithm and Congestion Avoidance in case Retransmission time out. When The TCP Tahoe sender receives 3 duplicate ACKs ,it use the fast retransmit algorithm to detect and repair loss. Here the congestion widow will be set to one maximum segment size and then it increases exponentially.

TCP Reno [2]: TCP Reno uses Slow Start algorithm and Congestion Avoidance in case Retransmission time out, as in TCP Tahoe. When The TCP Reno sender receives 3 duplicate ACKs, it uses the fast retransmit and fast recovery algorithm. Here the congestion window will be set to half and then it increased linearly.

TCP NewReno[13]: TCP NewReno uses same TCP Reno algorithms except that the Fast Recovery algorithms in NewReno continue until all the packets of the first loss are acknowledged. In TCP Reno receiving partial acknowledgments of the lost packets in the fast recovery phase will make TCP sender wait for Retransmission time and then go in Slow start phase.

TCP Vegas [3]: TCP Vegas is a new TCP version proposed by “Lowrence S. Brakmo”[3] with a different congestion control algorithms from that of other TCP versions. TCP Vegas uses three algorithms to handle the congestion control problem. The first algorithm is an extension for Reno’s retransmission mechanisms, which result in a more timely decision to retransmit a dropped segment. The second algorithm enables TCP to anticipate congestion and adjust its transmission rate accordingly. the last algorithm uses a modified TCP’s slow-start algorithm. So as to avoid packet losses while trying to find the available bandwidth.

TCP SACK1[12]: TCP Sack work differently than TCP Tahoe, TCP Reno, TCP NewReno. TCP Sack is a solution for the problem of multiple packets losses. In the previous TCP versions, the TCP receiver inform the TCP sender about a single lost packet over round trip time, because of the limited information available from cumulative acknowledgment. The TCP Sack enables the receiver to sends back SACK packet to the sender telling the sender of data that has been received. Then, the sender can retransmit only the lost packets, so, the sender can handle multiple packet loss in one round trip time. In another word, TCP Sack survives multiple segment losses within a single window. TCP Sack's congestion control is similar to TCP NewReno's congestion control algorithm with a few different variables to modify the congestion window size.

TCP Fack[4]: TCP Fack congestion control algorithms uses the similar basics of congestion control used in TCP Tahoe and Reno and it works in conjunction with the proposed TCP Sack.

3. CELLULAR IP ACCESS NETWORKS

The cellular IP protocol proposal [7] [8] [10] [11] from Columbia university and Ericsson allows routing IP Datagram to a mobile host. The protocol supports fast handoff and paging techniques. The protocol intended to provide local mobility and handoff support. Cellular IP can interwork with mobile IP [6] to provide macro mobility support.

Cellular IP base stations periodically emit beacon signals. Mobile hosts use these beacons signals to detect the nearest base by the mobile station. Cellular IP network route all IP packets transmitted host from the base station to the Gateway by hop-by-hop shortest path routing.

Cellular IP [8] maintain tow type of distributed cache for location management and routing purposes. Packets transmitted by the mobile host create and update entries in each node's cache. A mobile host also maintains its routing cache mappings even though it is not regularly transmitting data packets, through transmit rout-update packets on the uplink at regular interval called rout-update time when the mobile host move to another access point, the chain of mapping entries always points to its current location because its route-update and uplink packets create new and change old mapping. The mobile host connected to a cellular IP network is always in either Idle state or Active state. The Idle mobile host transmit paging-update packet when the paging time expires. It is used for location management. The paging update packet routed from Base station to the Gateway using hop-by-hop shortest path routing.

3.1 Handoff In Cellular IP

Handoff in cellular IP is a moving from one access point to another access point during an ongoing data transfer. Cellular IP support tow type of handoff to reduce the loss of downlink packets during migration between access points.

A. Hard Handoff

Cellular IP base stations periodically emit a beacon signals. The mobile hosts listen to this beacons signal and then initiate handoff based on signal strength measurements. Cellular IP uses ECS technique for handoff detection. The Mobile host perform handoff procedure by tunes its radio to a new base station and then send route-update packet. The route-update packet creates or modifies routing cache entries in Cellular IP nodes to the Gateway. The routing cache entries constitute a reverse

Copyright© ObComAPC-2004

path for downlink packet to new base station. Handoff time can be defined as the time between initiate handoff till reception of the first packet through the new access point. Also the handoff time has been defined as the round-trip time between the mobile host and the crossover node.

B. Semisoft Handoff

Semisoft handoff is optimized for networks where the mobile host is able to listen/transmit to two or more base stations simultaneously for a short duration, as in CDMA network.

When mobile host receive a beacon from new base station, it sends a semisoft packet to the new base station and immediately returns to listening to the old Base station. The semisoft packet will create new routing cache mappings from new base station to crossover node. The mobile host makes final handoff decision after some delay called semisoft delay. The semisoft delay can be an arbitrary value that is proportional to the mobile -to-gateway round-trip delay.

Cellular IP introduce delay at crossover node to synchronize the delay difference between the old route and new route from the crossover node in case the new route is shorter than the old route. The crossover node notified that a semisoft handoff is in progress from the semisoft packet received from a mobile host that has mapping to another interface. Cellular IP use flag in semisoft packet to indicate that downlinks packets must keep in delay buffers before being forwarded for transmission along the new path. After handoff is complete, the mobile host sends data or route-update packet through new path. Cellular IP use the route-update packets or data packets to clear the flag causing all packets in delay buffers to be forwarded to new path. Cellular IP use small delay buffers at base station for the same purpose. Cellular IP does not make any restriction on packet that cannot sustain additional delay at delay buffers and delaying handoff is prohibited. These packets should be forwarded to new path with out any delay.

4. SIMULATION MODEL

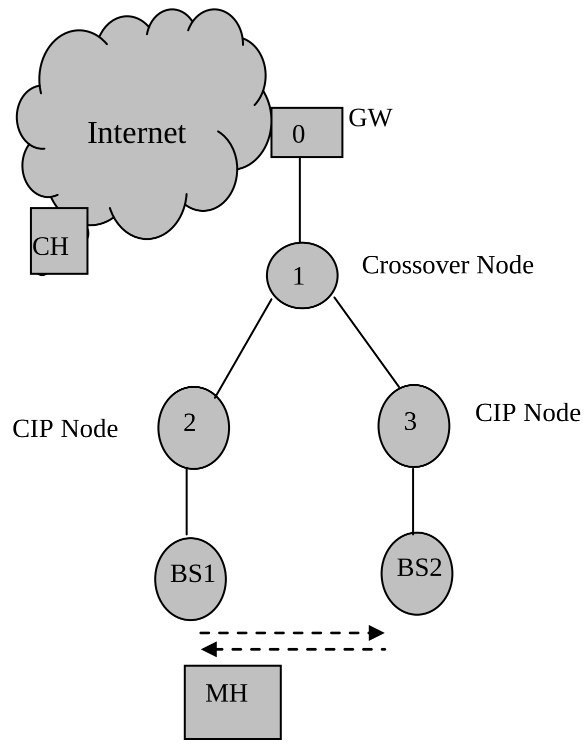

Figure 1: Network topology used for simulation.

4.1 SIMULATION ENVIRONMENT

We used in our simulation study the Columbia IP micromobility software (CIMS) [3], which it is a micromobility extension for the ns-2 based on version 2.1b6. The network topology used in this simulation is as shown in Figure-1.

Under this simulation we assume that the base stations and CIP Nodes are the wireless access point and router of IP packets while performing all mobility functions. The gateway node is a router, which connects the Cellular IP network to Internet. The mobile host connects to the corresponding host using the IP address of the Gateway as the care-of address.

The packet size used in the simulation is 512Byte (typically packet size 536 or 512 byte)[RFC2001].The Mobile Host moves from BS1 to BS2 and from BS2 to BS1. The speed of the Mobile Host is 30m/s and 50m/s. the link delay is 2ms. The simulation time is 10 seconds. the window size is 8 segment.

4.2 SIMULATION RESULTS

A. Analysis of TCP versions Behavior during Hard handoff.

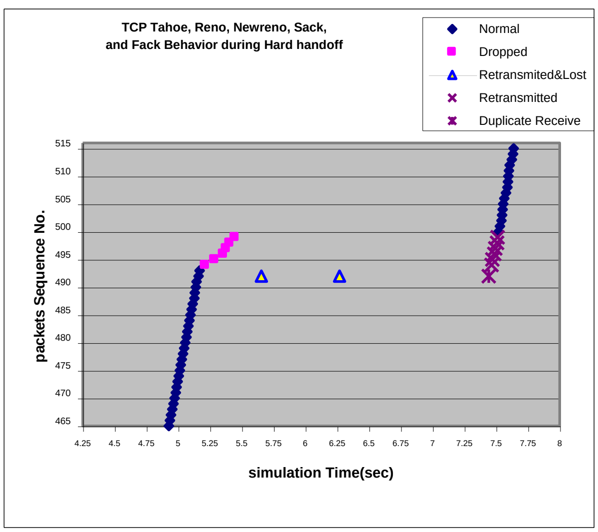

Figure 2: TCP Tahoe,Reno, Newreno, Sack,and Fack during Hard Handoff

Figure 2 shows the TCP sequnce numbers vs. simulation time during hard handoff. During the simulation, it is observed that TCP Tahoe, TCP Reno, TCP Newreno, TCP Sack, and TCP Fack behave the same. Also, Figure 2 shows the dropped packets during the hard handoff. Packets 0 – 493 are received normally. The acknowledgment packet of packet 492 is dropped. All packets (e.g, 494,495,496, 497, 498, 499) that came after handoff are dropped. It can be seen from the figure that the packets loss caused by the handoff results in a TCP timeout, so that no packets are transmitted during the timeout period, and the performance of TCP is seriously degraded. It can be observed that TCP sender did not receive any duplicate ACK. So, TCP sender wait until timeout of packet 492 and then TCP invoke Slow Start Algorithm, and retransmitted packet 492. After 2nd timeout TCP sender retransmitted packet 492. After 3rd timeout TCP sender retransmitted packet 492, which it was received by TCP receiver. It can be observed that the TCP timeout interval is doubled with every successive timeout [1]. This functionality is called TCP exponential backoff[1]. It can be seen that during a Cellular IP handoff all TCP versions experience several successive timeout that increase the timeout interval beyond the duration of the handoff. TCP receiver sent the ACK of packet 493, which it is the last packet received normally at the receiver.

It can be observed that All TCP versions commence the communication with the slow start algorithm after timeout [1][5].

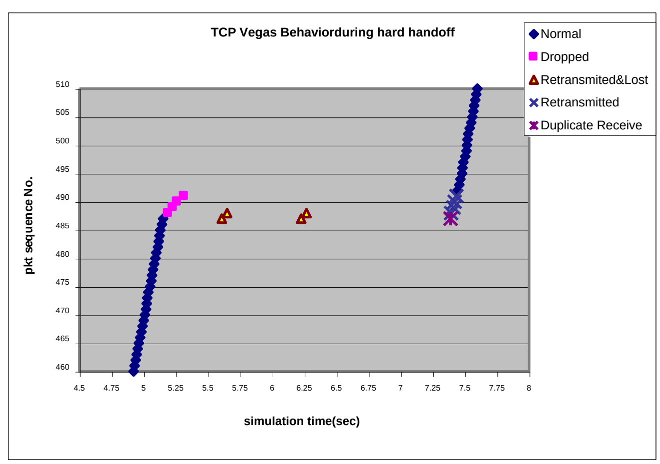

Figure 3: TCP Vegas during Hard Handoff

Copyright© ObComAPC-2004

Figure 3 shows the TCP Vegas sequence numbers vs. simulation time during hard handoff. During the simulation, it is observed that TCP Vegas behave a bit different than other TCP versions. Figure 3 shows TCP Vegas with dropped packets during the hard handoff over cellular IP access network. Packets 0 – 487 are received normally. The acknowledgment packet of packet 487 is dropped. All packets (e.g, 488,489,490, 491) that came after handoff are dropped.

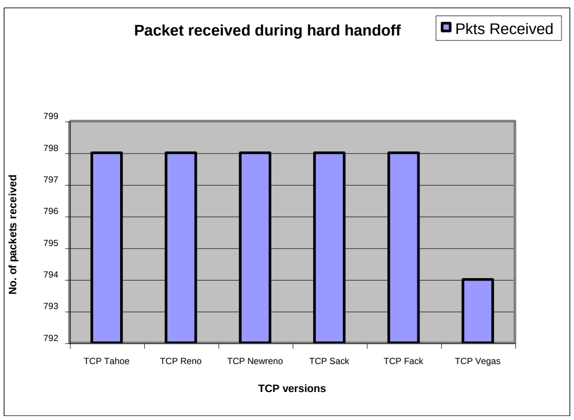

Figure 4: Packets received During Hard Handoff

B. Analysis of TCP versions Behavior during Semisoft handoff

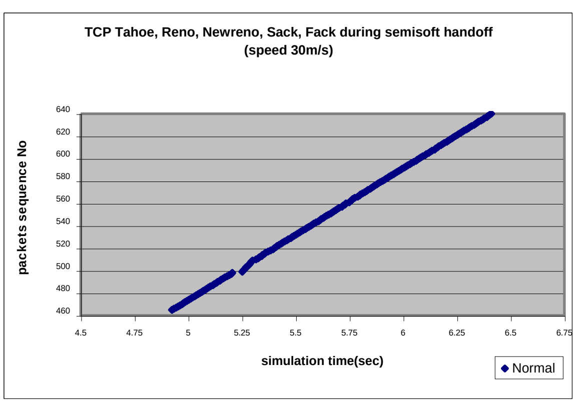

Figure 5: TCP Tahoe,Reno, Newreno, Sack,and Fack during semisoft Handoff with 30m/s Speed

Figure 5 show the down link packet sequence number observed at Mobile host in all TCP versions during semisoft handoff excepts TCP Vegas vc. Simulation time. In this test, the speed of the Mobile host is 30m/s. It can be observed that no packet loss occurs during the handoff, so that the TCP versions throughput is not affected by semisoft handoff.

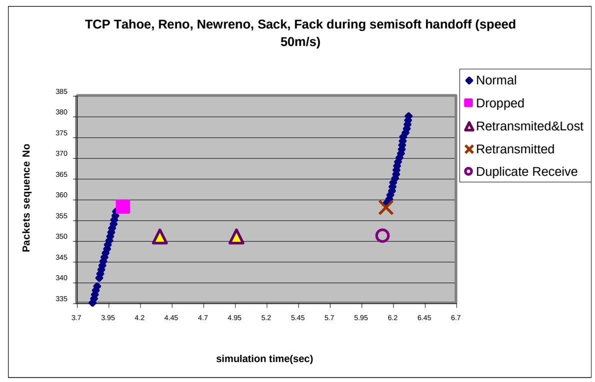

Figure 6: TCP Tahoe,Reno, Newreno, Sack,and Fack during semisoft Handoff with 50m/s speed.

Figure 6 show the packet sequence number in all TCP versions during semisoft handoff in Cellular IP access network excepts TCP Vegas vc. Simulation time. In this test, the speed of the Mobile host is 50m/s. the window size is 8 segment. The packets number 0-357 received normally, but the ACK of the packets 351 to 357 have been dropped during the handoff in addition to the packet 358. so that, the sender invoked slow start algorithm and retransmitted the packet 351 after 1st retransmission timeout. Then TCP sender retransmitted same packet after 2nd retransmission time out. After 3rd retransmission timeout, TCP receiver received the 3rd retransmitted packet and it was a duplicate packet. Then, the TCP receiver sent the ACK for packet number 357.

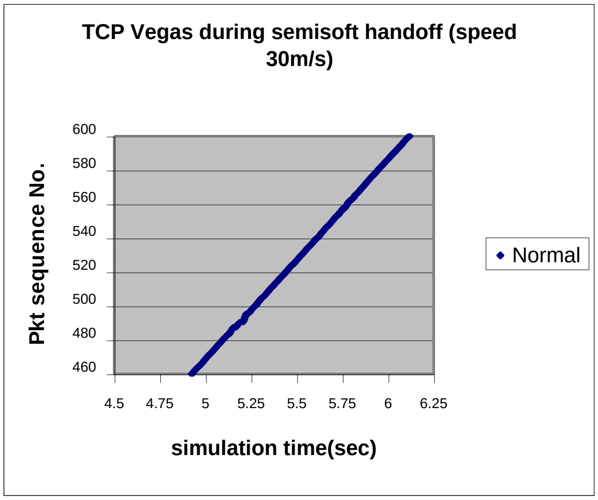

Figure 7: TCP Vegas during semisoft Handoff with 30m/s Speed.

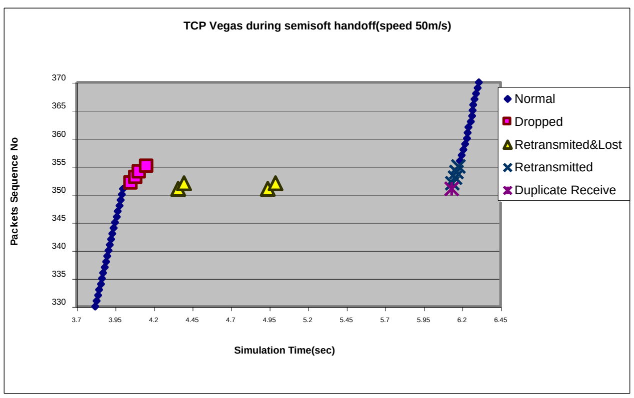

Figure 8: TCP Vegas during Semisoft Handoff with 50m/s Speed.

Figure 7 and Figure 8 show the packet sequence number in TCP Vegas during semisoft handoff in Cellular IP network with mobile host speed 30 m/s and 50 m/s respectively. It can be observed that no packet loss during handoff with 30 m/s mobile host speed, while some packet loss occur during handoff with 50 m/s mobile host speed.

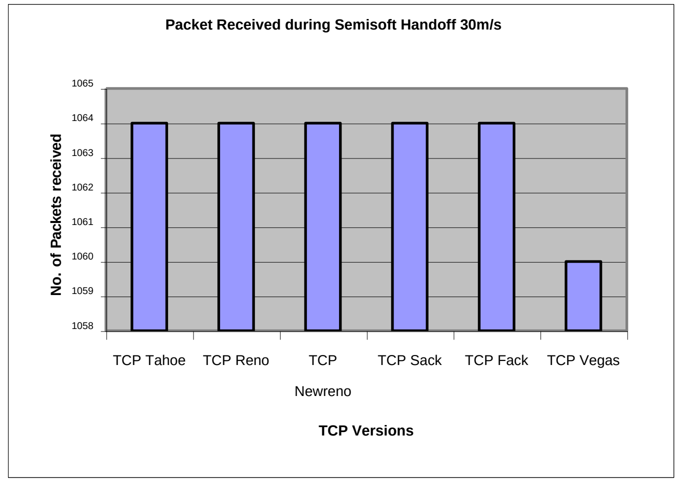

Figure 9: Packets received During Semisoft handoff 30m/s

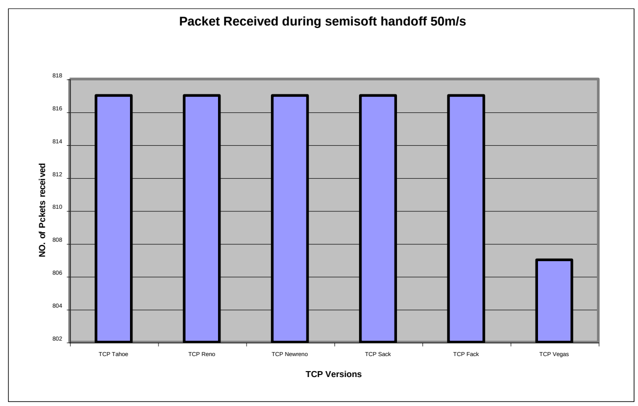

Figure 10: Packets received During Semisoft handoff 50m/s

Figure 9 and 10 show that TCP Vegas is the worst throughput with and without packet loss compared to other TCP versions during semisoft handoff.

5 CONCLUSIONS AND FUTURE WORK

In this paper the performance of TCP Tahoe, TCP Reno, TCP Newreno, TCP Sack, TCP Fack, and TCP Vegas during Hard and Semisoft handoff in Cellular IP network have been evaluated and compared. The study focuses on the impact of Handoffs on the performance of TCP versions. It is found that

Copyright© ObComAPC-2004

All TCP versions performance is hardly degraded due packets loss and handoff delay. It is identified that all TCP versions behave the same except TCP Vegas during the handoffs in Cellular IP Access Network. It is identified that TCP Vegas was the worst of all TCP versions during the hard handoff and Semisoft handoff.

However the comparison of performance of TCP implementations over Cellular IP network has not yet extensively investigated. This performance analysis and comparison of TCP Versions over Cellular IP access network can provide directions for farther improvement to TCP or Cellular IP.

For future study, it is recommended to analysis and compare the non-traditional TCP[19] like Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/Fast recovery, Transmission/time-out freezing, Selective retransmission, and Transaction oriented TCP over Cellular IP access network.

6 REFERENCES

[1] Tanenbam, Computer Network,4th edition,2003.

[2] W. Richard Stevens, TCP/IP Illustrated, Volume 1, Pearson Education, 2000.

[3] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson,TCP Vegas: New Techniques for Congestion Detection and Avoidance,

[4] Mathias, Matt, and Jamshid Mahdavi.., "Forward Acknowledgement: Refining TCP Congestion Control," Proceedings of Association for Computing Machinery Special Interest Group on Data Communication (SIGCOMM), 1996.

[5] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, Jun 1997.

[6] Ed. C. Perkins. IP Mobility Support for IPv4. Internet RFC, RFC 3220, January 2002.

[7] A. T. Campbell, J. Gomez, S. Kim, A. G. Valk´ o, C-Y. Wan, and Z. Tur´ anyi. Cellular IP. Internet draft, draft-ietf-mobileip-cellularip-00.txt, January 2000. Work in progress.

[8] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility," ACM SIGCOMM Comp. Commun. Rev., vol.29, no. 1,Jan. 1999, pp.50-65

[9] Micromobility home page, http://comet.columbia.edu/micromobility

[10] Andrew T. Campbell, Javier Gomez, Sanghyo Kim, Andras G. Valko, Chieh-Yih Wan, Zoltan R. Turanyi, " Design, Implementation, and Evaluation of Cellualr IP," IEEE Personal Communications, August 2000.

[11] A. T. Cambell, S. Kim, J. Gomez, C-Y. Wan, Z. Turanyi, A. Valko, Cellular IP Performance, draft-gomez-cellularip-perf-00.txt, May 2000.

[12] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP Selective Acknowledgment Options, RFC 2018, October 1996.

[13] S. Floyd, T. Henderson, The NewReno Modification to TCP's Fast Recovery Algorithm, RFC 2582 , April 1999.

[14] K.Fall and S. Floyd, "Simulation-based Comparison of Tahoe, Reno, and SACK TCP," ACM Computer Communication Review, Vol.26, No.3, pp.5-21, July 1996.

[15] Hiroyuki KOGA, Yoshiaki HORI, Yuji OIE, Performance Comparison of TCP Implementations in QoS Provisioning Networks, INET 2000 Proceedings, july 2000.

[16] Kenji KURATA , Go HASEGAWA , Masayuki MURATA, Fairness Comparisons Between TCP Reno and TCP Vegas for Future Deployment of TCP Vegas, INET 2000 Proceedings, july 2000.

[17] M. Allman, V. Paxson, W. Stevens,TCP Congestion Control, RFC 2581, April 1999.

[18] URL:http://www.isi.edu/nsnam/ns/

[19] Jochen Schiller, “Mobile Communcations”, Addison-Wesley longman, Pearson Education Limited , 2000.

Copyright© ObComAPC-2004