

An Efficient TCP with Explicit Handover Notification for Mobile Networks

Haruki Izumikawa¹, Ichiro Yamaguchi² and Jiro Katto
Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan.
E-mail: {haruki, yamaguchi, katto}@katto.comm.waseda.ac.jp

Abstract

TCP is a popular internet protocol for reliable end-to-end data delivery, but it cannot be directly applied to wireless networks in which packet loss may be induced by higher BER or handover than congestion. TCP assumes that such packet loss is caused by network congestion and initiates congestion control procedures. In this paper, we present a novel protocol using Explicit Handover Notification to improve TCP performance over wireless links. Additionally, we execute computer simulations using network simulator and compare with other various protocols.

1. Introduction

Future wireless networks are expected to move to end-to-end IP connectivity because of its simplicity and reduction in costs. Mobile IP [1] [2] is a promising approach that will be deployed over next generation wireless networks. It provides mobility support at the IP layer, allows mobile nodes to change its point of attachment without changing its IP address, and maintains their communications consistently. In Mobile IP, packets addressed to a mobile node are delivered using regular IP routing to a temporary address assigned to the mobile node at its actual point of attachment.

The Transmission Control Protocol (TCP), used by popular Internet applications such as File Transfer Protocol (FTP), Web browsing, E-mail and Telnet, is a popular internet protocol for reliable end-to-end data delivery. However, TCP cannot be directly applied to wireless networks in which packet loss may be induced by higher Bit Error Rate (BER) or handover than congestion; it assumes that such packet loss is caused by network congestion and initiates congestion control procedures (e.g. reduction of its congestion window (cwnd)). This incorrect assumption causes TCP to perform poorly in wireless environments.

Recently, many researchers had studied the performance of TCP over wireless networks [3] [4] and

several schemes have been proposed for improving TCP performance over wireless links. The main ideas used to improve TCP performance on transport layer can be classified into three major compartments mentioned below.

- i) Split Connection Approaches
(e.g. Indirect TCP [5], M-TCP [6], etc.)
- ii) Proxy Approaches
(e.g. Snoop TCP [7], WTCP [8], etc.)
- iii) End-to-end Approaches
(e.g. Freeze TCP [9], ELFN [10], etc.)

Note that i) and ii) are approaches that deal with packet losses caused by higher BER on wireless links at the border gateway (base station). On the other hand, iii) is the approach that deals with packet losses caused by disconnection such as handover without assuming gateway intervention. In addition, there are a lot of TCP versions, such as Reno, NewReno, SACK and Westwood (TCPW) [11], which were mainly developed for improving congestion control performance. Some of them were tuned to have robustness against bursty packet losses, that might be promising for wireless networks.

In this paper, we focus our attention on performance improvement of TCP during handover which causes serious TCP degradation [4]. As the explosive growth of the mobile Internet, especially in urban area, cell size may become small because of higher frequency by adopting high bandwidth and high density of mobile users. Small cell size leads to frequent user handovers though BER might be reduced. Therefore, the performance degradation due to handover tends to be more serious to mobile users moving frequently than due to high BER. In case of packet losses due to random errors, the sender can perform a retransmission of what appears to be missing segments using the "fast retransmit" algorithm without waiting for the retransmission timer to expire after receiving 3 duplicate ACKs. On the other hand, in case of packet losses due to handover, the sender may have to wait for a retransmission timer to expire and perform "slow start" since in-flight packets are forwarded to the previous access point to which the mobile node attached before movement and result in packet losses. Therefore, to improve TCP performance over wireless networks, we present a novel protocol using Explicit Handover Notification (EHN) that maintains TCP end-to-end

¹ Haruki Izumikawa is currently with KDDI R&D Laboratories, Inc., Japan.

² Ichiro Yamaguchi is currently with NEC Corporation, Japan.

semantics and deals with packet losses caused by handover.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 presents our proposed scheme in detail. In Section 4, we describe the simulation scenario and compare the simulated protocols. Finally, conclusions are provided in Section 5.

2. Related Work

As we described above, a number of studies have been conducted to improve TCP performance in wireless links. In this section, we briefly outline some typical schemes. We here refer to transport layer approaches and Mobile IP extensions though we omit various link layer approaches.

2.1. Split Connection

These schemes split the end-to-end TCP connection between a sender and a receiver at a base station that separates wired/wireless connection. For example, with Indirect TCP (I-TCP) [5], regular TCP is used from a fixed host to a base station whereas a modified TCP protocol that suits to wireless environment is used from the base station to a mobile node (MN). So, transmission errors on wireless link are not propagated to wired network. It attempts to separate loss recovery on wireless link from wired link. However, there are some problems such as high processing load and limited buffering space available at the base station, which buffers transferred TCP segments and serves as the MNs from the fixed side of the network. Another disadvantage of this scheme is loss of TCP end-to-end semantics. Furthermore, special changes at the base station are required and encrypted traffic (e.g. in case IPsec is used, which is currently used in VPN and supported by IPv6) cannot be handled.

2.2. Proxy

These schemes require the base station to buffer all packets destined for the MN and retransmit lost packets. If the base station receives duplicate ACKs from wireless side, it suppresses the ACKs and performs local retransmission. The main advantage of these protocols is avoiding unnecessary fast retransmissions and congestion control invoked at the sender by suppressing duplicate ACKs and retransmitting locally. In addition, these schemes preserve end-to-end semantics of TCP. However, timeout at the sender might occur due to retransmissions on wireless TCP. Another disadvantage of these schemes is that snooping and caching may fail if end-to-end encrypted schemes are used. Furthermore, buffering overhead for potentially thousands of connections is not negligible.

2.3. End-to-end

Freeze TCP is one of the end-to-end protocols to improve TCP over wireless link by letting the sender enter "persist mode" prior to a disconnection through signal strength measurements at the wireless antenna [9]. In this protocol, when an MN detects impending handover, it sends zero window advertisement (ZWA) to the sender. When the sender receives the ZWA, it freezes all retransmit times and enters persist mode without shrinking its cwnd. After the connection is reestablished, the MN sends three successive acknowledgements called TR-ACKs (Triplicate Re-connection ACKs). When the sender receives them, it resumes transmission. This scheme does not require any help from a base station and emphasized that only MN's TCP code needs to be changed. Furthermore, it can be used with encrypted data.

However, with Freeze TCP, the time before which actual disconnection happens, in other words "warning period", is quite a critical issue to be predicted. In [9], a reasonable warning period is estimated as the current round-trip-time (RTT). If the warning period is any longer, the sender will go into persist mode too early, which leads to longer idle time and throughput degradation. On the other hand, if the warning period is too short, there may not be enough time for an MN to send the ZWA to the sender, which causes the sender's cwnd size to be reduced in response to dropped packets during disconnection. In detail, even if the TR-ACKs are received successfully by the sender, there exist in-flight packets to be dropped and the "fast retransmit" algorithm is invoked. In fact, performance improvement of this scheme is totally dependent on accurate prediction of disconnection by the MN [12]. But according to [13], since RTT is often measured very coarsely (the granularity might be 500msec) by the sender instead of the MN, assumption that the MN has the knowledge of RTT may not be practical. Another disadvantage of this scheme, which is much more serious, is that it is quite difficult for MAC layers below to detect precise future disconnections.

Another related paper to ours is [4], in which explicit handoff notification was referred to. However, their approach assumed classical TCP Tahoe and it simply initiates normal fast retransmission after handover followed by slow start (i.e. the cwnd value is reduced to one), which results in drastic throughput degradation when handovers occur consecutively. On the other hand, our proposal recovers the cwnd value without drastic reduction so as to keep high throughput to MN users as possible as we can, similar to the Freeze TCP argument. We also compare various state-of-art TCP versions running on Mobile IP with careful observations on their possible robustness against handovers, and quantitatively prove the efficiency of our proposal.

Table1: Characteristics of some wireless TCP protocols

	I-TCP [5]	Snoop TCP [7]	Freeze TCP [9]	TCP with EHN
Handover support	no	No	yes	Yes
High BER robustness	yes	Yes	no	No
Encrypted traffic support	no	No	yes	Yes
Need intermediate node	yes	Yes	no	No
End-to-end semantics	no	no	yes	yes
Long disconnection robustness	no	no	yes	yes
ACKs can be routed along different paths from data	no	no	yes	yes
Need impending disconnection prediction	no	no	yes	no

2.4. Mobile IP Extensions

Different from the previous three approaches, Layer 3 approaches extending Mobile IP are also proposed. Examples are Hierarchical MIP (HMIP) [14], Fast MIP (FMIP) [15], Simultaneous Binding (SB) [16] and Localized Mobility Management in a Distributed manner (LMMD) [17]. HMIP and LMMD try to reduce handover delays by localizing signaling procedure of Mobile IP. FMIP and SB do so by buffering (and tunneling) or packet replication, both contributing to packet loss reduction. We had also proposed our own approaches; smooth handover using small group multicast [18] and LMMD extension using geographical information support [19].

A disadvantage of these approaches is their computational complexity forced to access routers and domain gateways. However, they are orthogonal to the transport layer approaches, as well as to the link layer approaches. When these MIP extensions are deployed, further performance improvement of TCP is expected.

3. TCP with Explicit Handover Notification

In this section, we propose an efficient mechanism, where the MN sends an Explicit Handover Notification (EHN) to the sender when the MN performs handover. We assume that the mobility protocol is Mobile IP although other protocols can be applied.

The mechanism of this scheme is very simple. The sender stores previous cwnd size and slow start threshold (ssthresh) whenever they are changed. The MN also stores an ACK that it last sent, i.e. ACK for the last data segment it received. Base stations send advertisement messages (e.g. unsolicited multicast Router Advisements in Mobile IPv6) at a regular interval to help the MN to

detect link changes. With our scheme, as soon as the MN receives this advertisement after handover, it sends an EHN packet to the sender as well as sends a registration packet to MN's Home Agent yielding to Mobile IP procedure. The EHN packet is a special ACK packet which extends a TCP header with 'Handover bit' (or using TCP optional fields). Other values in the TCP header are same as the ACK stored by the MN.

If the sender receives an ACK with this EHN indication, it resets the retransmission timer and sets *snd.next* (next sequence number to be transmitted in TCP sliding window) to *snd.una* (unacknowledged) after it adjusts its send window in response to the sequence number of this ACK. The sender does not reduce its *cwnd* size since it can distinguish handover losses from congestion losses as long as the retransmission timer is not expired. In case *cwnd* size is reduced after the retransmission timer has expired, the sender restores the *cwnd* and *ssthresh* values to previous ones stored by the sender. Furthermore, it can estimate RTT as double the time of one-way delay from the MN to the sender if TCP timestamp option is used. The main advantage of this scheme is to enable the sender to explicitly distinguish packet losses due to handover from others. In addition, it requires no prediction of impending disconnection at the MN. We summarize the characteristics handled by the protocols discussed in this paper in Table 1.

An example is shown in Figure 1. In this figure, S, R and BS represent a sender, a router and a base station respectively. At first, the MN attaches to the BS1. At this moment, the status of send window at the sender is SW1 in Figure 1. The sender sends TCP segments 1, 2, 3 and 4. The MN receives segment 1 and 2 and returns ACK1 and ACK2 for these segments (①). It assumes that the MN moves to BS2 from BS1 (i.e. handover occurred) almost as soon as it sends ACK2 and that TCP segments 3 and 4 as well as ACK2 are dropped during this handover (②).

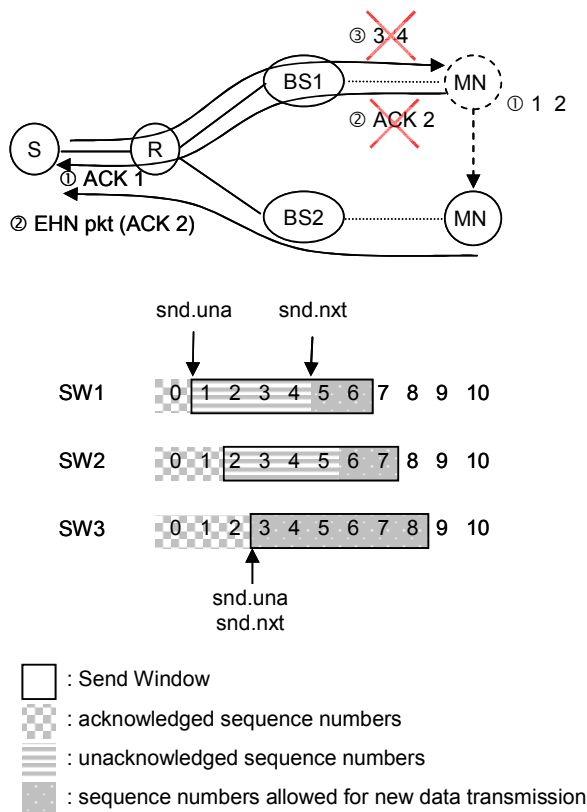


Figure 1: Overview of TCP with EHN

When the sender receives ACK1, the status of send window turns from SW1 to SW2. As soon as the MN receives the advertisement from the BS2, it sends the EHN packet to the sender (③). When the sender receives this EHN packet, the status of send window turns from SW2 to SW3.

To compare the performance of TCP with EHN to those of Freeze TCP [9] and TR-ACKs in [4] which are the end-to-end approaches mentioned above, we execute simulations using network simulator ns-2 (version 2.1b8) [20] in next section. There, we also compare various TCP versions which have different rate control algorithms, known as Reno, NewReno, SACK and TCPW.

4. Performance Evaluations

4.1. Simulation Models

We use network topology shown in Figure 2. A wired link of backbone network has a bandwidth of 100Mbps and a delay of 50msec. Each wired link of user access network has a bandwidth of 10Mbps and a delay of 3msec, while the wireless link has a bandwidth of 2Mbps and a delay of 5msec.

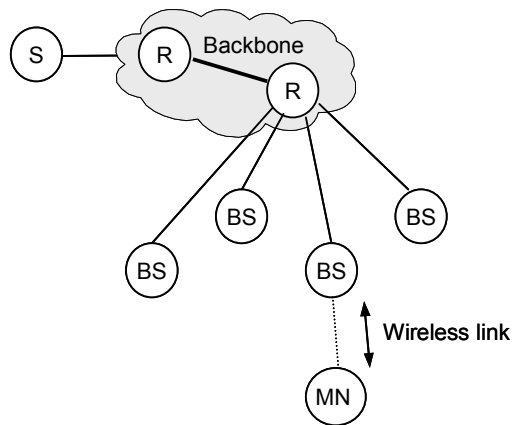


Figure 2: Network topology used in simulations

The wireless link assumes to be lossy. The wireless error models we use in this simulation are a random error model and a burst error model. In the random error model, packets over wireless links may be dropped with probability of 1% uniformly, i.e. packet error rate (PER) is 1%. Note that 1% PER corresponds to 1.25×10^{-6} BER assuming packet size is 1000byte. On the other hand, in the burst error model, each burst error which comprises three packets may occur with probability of 0.5 %, i.e. 1.5% PER though the interval at which burst errors occur is twice as much as the interval at which random errors do.

The interval of Mobile IP router advertisements from the base station is 0.1sec. Therefore, the mean time at which a MN may detect its own movement is 50msec. The MN migrates among base stations at 5sec intervals. For TCP parameters, packet sizes are 1000 bytes and ACK sizes are 40 bytes. TCP segments are transferred via FTP from the sender to the MN. We use drop tail queues at all nodes.

To investigate the impact of the warning period for Freeze TCP, the warning period is varied; equal to RTT (ideal value), $RTT+0.5sec$ and $RTT+1.0sec$, respectively. We execute simulations for 300sec and repeat each simulation five times with random values. Simulation results are derived from the average of these five times repetitions.

4.2. Simulation Results and Observations

A. Throughput Comparison of Various TCP Combinations

We evaluate throughputs of various TCP combinations – Original TCP, TR-ACKs in [4], Freeze TCP [9] (three variations of warning period) and TCP with EHN on Reno, NewReno, Sack and TCPW, respectively. Figure 3 and Figure 4 show the average throughputs of four different TCPs assuming random errors and burst errors

on wireless link, respectively. Among them, the throughput of TCP with EHN is approximately the same as Freeze TCP where the warning period is RTT (i.e. ideal case) regardless of wireless error models. In addition, improvement in throughput of TCP with EHN becomes about 50% compared with Original TCP, about 30% compared with that of Freeze TCP (warning period is $RTT+1.0\text{sec}$), and about 15% compared with Freeze TCP (warning period is $RTT + 0.5\text{sec}$) regardless of wireless error models. This means that the choice of time to send ZWA is quite critical in Freeze TCP, of which throughput decreases as the warning period is not equal to RTT. In addition, about 10~15% improvement is usually achieved by TCP with EHN against TR-ACKs. This is mainly due to cwnd recovery procedure after handover.

With regards to TCP versions with different rate control strategies, TCPW, that is based on Reno and improves TCP performance by its own bandwidth estimation strategy, certainly shows higher performance than others under the random error model. However, its performance is lower than those of NewReno and Sack under the burst error model. The reason of this fact is that TCPW is based on Reno which suffers from consecutive packet losses as described in [21]. On the other hand, NewReno and Sack are useful in lossy links, especially for burst errors.

B. Throughput Comparison of Window Management Mechanisms after Handover

We also investigate the impact of window management strategies of a MN after handover on fixed nodes (FNs) in a new domain. We compared three approaches: (1) Resume with slow start, in which a MN always initiates slow start procedure when it moves to a new domain [4]. (2) Resume with previous values, in which last cwnd/ssthresh values of the previous domain are initially applied in a new domain. (3) Resume with adjusted parameters, in which a RTT between a MN and a base station in a new domain is estimated from the EHN packet with TCP timestamp option and is used to determine initial cwnd/ssthresh values by the TCPW window update procedure. Figure 5 shows a simulation model, in which a MN moves around randomly inside four domains. In each domain, there is a base station to which zero to three fixed nodes are attached. When a MN moves into a new domain, it initiates one of three window management procedures as described above.

Table 2 shows average throughputs of the fixed node attached to BS1, in which TR-ACKs, Freeze TCP (with RTT warning period) and TCP with EHN are compared. A MN applies "Resume with slow start" to the first one, and "Resume with previous values" to the last two. This table shows that their throughputs are almost the same, i.e.

effect of the new user arrival onto existing users is small enough, even if the TCP with EHN (and Freeze TCP) does not reduce the cwnd value to one after handover.

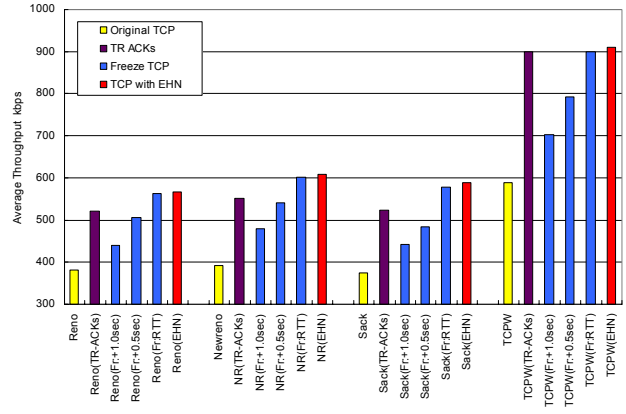


Figure 3: Throughput comparison under the random error model at wireless link

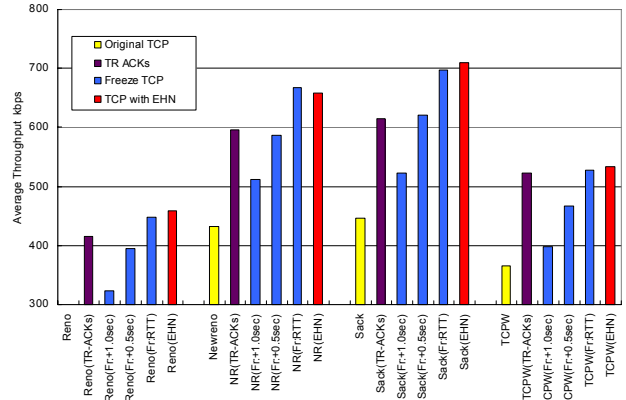


Figure 4: Throughput comparison under the burst error model at wireless link

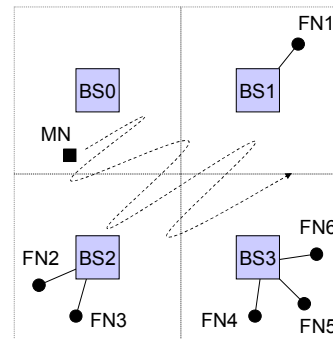


Figure 5: A mobility model used in a simulation

Table 2: Throughput comparison of a fixed node (kpbs)

	Reno	NewReno	Sack	TCPW
TR-ACKs	566.43	611.67	608.70	871.30
Freeze TCP	562.20	612.03	607.71	872.06
TCP with EHN	563.700	610.57	609.05	871.60

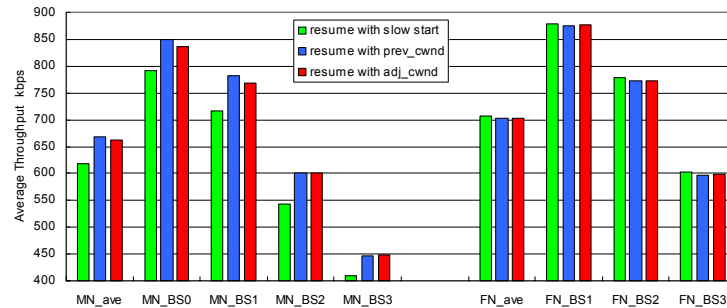


Figure 6: Throughput comparison of different window management mechanisms after handover.

Figure 6 shows a throughput comparison result of a MN and FNs, according to three different window management strategies assuming TCPW. In this figure, MN_ave means an averaged throughput of the mobile node, and MN_BS i denotes that when the mobile node belongs to domain i (i.e. it is attached to the base station i). Similarly, FN_ave represents an averaged throughput of fixed nodes, and FN_BS i does that of fixed nodes belonging to domain i . From this figure, we can notice that the slow start resumption perform worst although the other two approaches show comparable performances. This is because the other two approaches are different only in their initial parameter setting, with the same congestion avoidance procedure followed. We also notice that, fixed node performance is not so drastically affected by the window management procedure as long as the tested appropriate mechanisms are applied.

5. Conclusions

This paper described an efficient transport layer approach using explicit handover notification to improve TCP performance over wireless networks. Split connection and proxy approaches force high processing capability at base station and have difficulty in support of encrypted traffic. Freeze TCP, a previous end-to-end approach, has to predict precise future disconnections and suffers from severe performance degradation when the prediction fails. Our approach, using explicit handover notification, demonstrates efficiency and stability by performing reactively against handovers. Detailed comparisons of various TCP combinations and window management strategies after handover are also provided. As future work, since our adaptation method still needs refinement together with congestion control strategy, more sophisticated window management mechanism has to be considered.

References

- [1] C.Perkins, Ed.: "IP Mobility Support," RFC2002, Oct.1996.
- [2] D.B.Johnson, C.Perkins and J.Arkko: "Mobility Support in IPv6," draft-ietf-mobileip-ipv6-19.txt, Oct.2002.
- [3] H.Elaarag: "Improving TCP Performance over Mobile Networks", *ACM Computing Surveys*, Vol.34, No.3, pp.357-374, Sep.2002.
- [4] R.Caceres and L.Iftode: "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," *IEEE Journal on Selected Areas in Communications*, Vol.13, No.5, Jun.1995.
- [5] A.Bakre and B.R.Badrinath: "I-TCP: Indirect TCP for Mobile Hosts," *IEEE ICDCS'95*, pp.136-143, 1995.
- [6] K.Brown and S.Singh: "M-TCP: TCP for Mobile Cellular Networks", *ACM Computer Communication Review*, 1997.
- [7] H.Balakrishnan, V.Padmanabhan, S.Seshan and R.H.Katz: "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *ACM SIGCOM'96*, Aug. 1996.
- [8] K.Ratnam and I.Matta: "WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links", *IEEE ISCC'98*, 1998.
- [9] T.Goff, J.Moronski, D.S.Phatak, and V.Gupta: "Freeze-TCP: A True End-to-end TCP Enhancement Mechanism for Mobile Environments," *IEEE INFOCOM 2000*, Mar. 2000.
- [10] G.Holland and N.H.Vaidya: "Analysis of TCP Performance over Mobile Ad Hoc Networks," *ACM/IEEE MOBICOM '99*, Aug. 1999.
- [11] S.Mascolo et al.: "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," *ACM MOBICOM 2001*, Jul, 2001.
- [12] G.M.T.Da.Costa and H.R.Sirisena: "Freeze TCP with Timestamps for Fast Packet Loss Recovery after Disconnections," *Int. Symp. on Performance Evaluation of Computer and Telecommunication Systems*, Jul. 2002.
- [13] F.Hu and N.K.Sharma: "The Quantitative Analysis of TCP Congestion Control Algorithm in Third-Generation Cellular Networks Based on FSMC Loss Model and its Performance Enhancement," *IEEE INFOCOM 2002*, Jun. 2002.
- [14] H.Soliman et al.: "Hierarchical MIPv6 Mobility Management," draft-ietf-mobileip-hmipv6-03.txt, Feb.2001.
- [15] G.Dommety et al.: "Fast Handovers for MIPv6," draft-ietf-mobileip-fast-mipv6-03.txt, Jul.2001.
- [16] K.El Malki et.al.: "Simultaneous Bindings for Mobile IPv6 Fast Handovers," draft-elmalki-mobileip-bicasting-v6-02.txt, Jun. 2002.
- [17] J.Choi et al.: "Localized Mobility Management for Mobile IPv6 in Distributed Manner," draft-jinchoi-mobileip-lmmdv6- 01.txt, Nov.2001.
- [18] H.Izumikawa and J.Katto: "Seamless Handover in Mobile IP using Small Group Multicast," *2001 Commun. Society Conf. of IEICE*, B-7-18, Sep.2001 (in Japanese).
- [19] H.Izumikawa and J.Katto: "Distributed Localized Mobility Management using Geographical Location Information," *IEICE Tech. Report*, IN2002-110, Nov.2002 (in Japanese).
- [20] "The Network Simulator - ns-2," <http://www.isi.edu/nsnam/ns>.
- [21] K.Fall and S.Floyd: "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", *Computer Communication Review*, Vol.26, pp.5-21, Jul.1996.