An efficient TCP/IP hand-over control scheme for next-generation mobile communication networks

Yuko Onoe † Yukio Atsumi †
Fumiaki Sato ‡ Tadanori Mizuno ‡

† Multimedia Laboratories, NTT DoCoMo
3-5, Hikarinooka, Yokosuka, Kanagawa, 239-8536, Japan
Tel: [+81]468-40-3327; fax: [+81]468-40-3364
E-mail: yuko, atsumi@mml.yrp.nttdocomo.co.jp

‡ Faculty of Information, Shizuoka University
3-5-1, Jouhoku, Hamamatsu, Shizuoka, 432-8011, Japan
Tel:[+81]053-478-1460;fax:[+81]053-478-1597
E-mail: sato, mizuno@cs.inf.shizuoka.ac.jp

We describe a control scheme for wireless-link layers and the TCP/IP layers in which wireless link states, such as signal strength and transmission rate, are transparent to the upper network layers and govern the behavior of the upper layers. Monitoring and notifying functions for wireless link states are incorporated into mobile hosts, and prefetching functions for the mobility agent lists are added to the present MobileIP functionalities of the network layers. In addition, we give the mobile hosts functions for controlling the TCP advertised windows in the transport layers and give the base-stations buffering functions to deal with the variable signal strength of the wireless links. A simulation (using ns-2) of this control scheme shows that mobile agent can be switched at the network layers soon after base-station hand-over and that errors in packet routing, packet loss and communication throughput reduction can be avoided. Moreover, communications can be interrupted without having to shrink the size of the congestion windows of the TCP senders, which improves overall throughput.

1. Introduction

The growth of the Internet and the availability of smaller and cheaper computers are fueling demand for portable wireless terminals capable of Internet access. Internet connectivities for mobile communication will be needed in the near future. To achieve transparency new Internet services are introduced into mobile telecommunications. In particular, it will be necessary to implement new IP control technologies based on next-generation high-speed mobile communication networks to enable the flexible construction of new network architectures. Many IP-based cellular network architectures targeting micro-mobility have been proposed (Table 1). They include, for example, MobileIP whose specifications are standardized by IETF (Internet Engineering Task Force) are for hierarchical structured networks. Moreover, there are some approaches in which hierarchical mobility management and communications between previous and current base-stations have achieved fast IP-based hand-offs[6].

As yet, however, the TCP/IP protocol layers of these architectures have no control schemes based on wireless-link characteristics, and as a result, their wireless-link layers and TCP/IP protocol layers operate independently of each other. This deficiency causes routing errors and packet loss and also reduces throughput. This study, therefore, is an attempt to develop a control for wireless links and TCP/IP protocol layers in which the wireless link states are transparent to the TCP/IP protocol layers. Current issues are examined in Section 2, proposed methods and their designs are discussed in Section 3, and the methods are evaluated and compared in Section 4. Finally future work to be done in this field is discussed in the concluding Section 5.

2. Current Issues

The next-generation of mobile communication networks will have higher transmission rates and radio frequencies, smaller cell sizes and a greater number of cells. When switching cells, mobile terminals (MH: mobile hosts) must have an efficient

Table.1: Comparison of current IP-based cellular network

Architecture	Advantages	Disadvantages
MobileIPv4 [1]	Extensions to general IP functionalities, communication continuity during movement	Redundant routing through home agents, overhead of tunneling functions
HMIP[6]	Reduced overhead for mobile registration	Many packet tunneling operations required
CellularIP [3]	Short routing paths, no foreign addresses, semi-handoff functions, private addresses	Many routing path entries at routers, redundant paths at intra-domain communications
HAWAII [4]	Short routing path, MobileIPv4 messages, possibility of communicating with several base-stations at a time, smooth handoff functions, securities for routing caches	Many routing entries in routers, redundant routing paths in intra-domain communications, unavailability of private addresses
MobileIPv6 [2]	Next-generation IP resolves address shortages, stateless auto address configuration, extension headers, no foreign agents	Large IP header sizes, packet loss during hand-overs, overhead for mobile registration
HMIPv6 [7]	Reduced overhead for mobile registration, smooth hand-over functions	Many packet tunneling operations required

IP-based hand-over scheme that can handle switching between cells of different characteristics. However, current wireless links and TCP/IP protocols work independently, and this causes problems. For example, the switching of MobileIP agents depends on advertising messages that are periodically issued by the agents. If there are no IP-based mobility controls and if agents aren't switched fast enough, packets may be lost. Another problem is that when packets are lost due to a poor wireless link state caused by say the host moving, fading or radio interference, TCP protocols regard the losses as network congestion and reduce shrink the size of the congestion windows. On wireless links, which are slow and possess long delays, the shrinking of the congestion window delays recovery of TCP communication throughput even if signal strength recovers comparatively quickly. An example of this phenomenon is shown in Fig. 1.

It shows the results of simulations in which mobile hosts use ftp communications during hand-over on a hierarchical network structure that is in accordance with IEEE802.11.

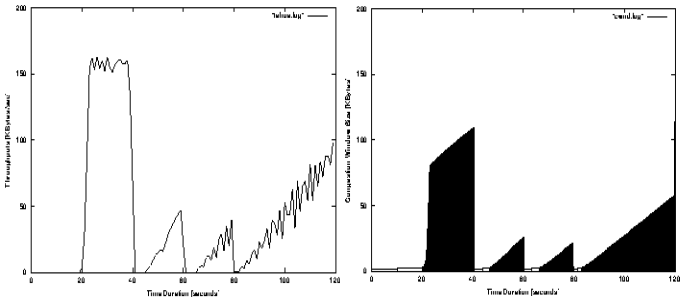

Fig1: Throughput (Kbytes/sec) (left) and congestion window size (right) at hand-over of TCP(Tahoe) implementation.

One can see that at base-station hand-over at 40, 60, 80 sec in Fig. 1, both the throughput and congestion window size are reduced and that after base-station hand-over, throughput is slow to recover. In this simulation, the normal communication rate was 160 Kbytes/sec on average and the ssthresh value for start of the congestion avoidance phase was set at the minimum value during the first hand-over. In this experiments, the end-to-end delay was set at 500 msec on average, because results of experimental measurement show RTTs for current PDC-P communication networks are about 500 msec. For this delay, it took a long time for the congestion window size to rise from its minimum value. A simple calculation showed that when the RTTs are 400 msec, it takes 24.0 sec for throughputs to recover to 160 Kbytes. When the RTTs are 500 msec, it takes 38.0 sec, and when they are 600 msec after the last hand-over, it takes 54.6 sec. In this way, in low-speed, long-delay network environments like wireless links, if congestion window size has once decreased, it takes much time for throughputs to recover.

3. Proposed TCP/IP mobility control scheme

The basic procedure of notifying the upper TCP/IP protocol layers of the wireless link state is shown in Fig. 2. By beacons transmitted by base-stations, signal strength of each wireless link is monitored. Agent solicitation messages are then transmitted by the mobile host to the base-stations that have still weak signal strengths. The agent list of the mobile host is updated with newly assigned addresses and new foreign agents (at an agent list collection). The IP layers are first notified of base-station hand-over (at a wireless link-states monitoring/notifying part), and new addresses and foreign MobileIP agent are allocated from a list of agents (agent list switching part). This procedure enables registration requests to be issued as soon as base-station hand-over is completed. When the signal strength of the base-station that has been selected for hand-over is too weak, the mobile host sends an acknowledgement to the sender indicating that the TCP advertised window size, which determines receiving capacity, is to be set to zero (TCP advertised window control). A detailed description of the procedure is given in the following sub-sections.

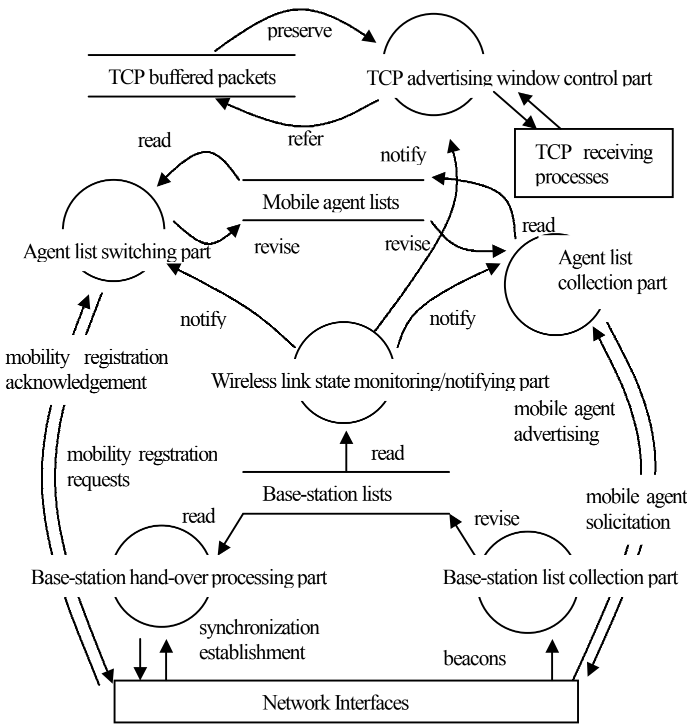

Fig. 2: Structure of a MobileIP control layer and the wireless datalink layer part of a mobile host

3.1. Prefetching functions of mobile agent lists

Base-station IDs, base-station list lifetimes, transmission rates and signal strengths are maintained on base-station lists (Table 2). A base-station sends out a beacon with its ID to any mobile host within its area. When the beacon's signal strength exceeds a certain threshold, the host transmits a solicitation message if the base-station ID is not on its mobile agent list (Table 3). The mobile agent list contains all base-station IDs that have been received within their lifetime. The MAC addresses of hosts can be maintained, even if the routing paths of the up-streams are different from those of the down-streams. A mobile host tends to change base-stations frequently when it is at the edge of a cell. This prefetching function ensures that the mobile host updates the agent list without having to request that the agent list be sent every time it switches base-stations.

Table 2: Base-station lists

BSID	Life -time	Receiving time	Transmission rate	Signal Stren -gth (SNR)	Exist flags
BS001	1h	03.01.10:00	2Mbps	8	On
BS010	1h	03.01.12:01	10Mbps	5	Off
BS023	1h	03.01.10:02	2Mbps	3	Off

Table 3: Agent lists

Agent ID	Life -time	Receiving time	Agent addresses	BSID
A001	1h	03.01.10:01	133.201.129.101	BS001
A003	1h	03.01.12:05	133.201.139.110	BS010
A010	1h	03.01.10:03	133.201.139.123	BS023

3.2. TCP advertised window control functions

The advertised window size is used to interrupt TCP communications temporarily when the signal strength becomes weak. This enables the TCP congestion window size to be shrunk without packet loss as well as prompt recovery of throughput. Moreover, we propose a cooperative control scheme of mobile hosts and base-stations where mobile agents locate to improve the ability to compensate sender's for end-to-end delays. Practically speaking, the base-station buffers TCP data temporarily, if the signal strength becomes weak. If it recovers, base-stations start to transmit the buffering data. At hand-over, the buffers are tunneled to the new mobile agents with MobileIP binding update functions.

Processing done at mobile host

The TCP advertised window control has TCP receiving and buffering functions and timer processing functions. The signal strength of all wireless links is monitored by the mobile host, and when the strength falls below a pre-determined threshold, ACK packets with the closed advertised window size are transmitted to the senders. The receiving TCP flow is then interrupted once, after which, the signal strength at the receiver is monitored periodically by timer processing. If it is found to exceed the threshold value, information such as received sequence number is acquired from the received packets, advertising window size is calculated, and ACK packets are transmitted.

Processing done at base-station

The station's TCP advertised window control procedure is shown in Fig. 3.

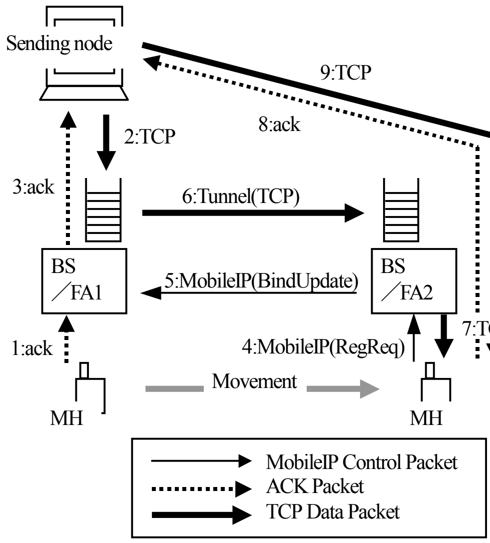

Fig3: Procedure of a TCP advertising window control scheme at base-station.

1. The mobile host transmits ACK packets specified with the closed TCP advertised window size to the sending nodes because of a lowering of signal strength.
2. The base-station buffers TCP data destined to the mobile hosts.
3. The base-station transmits ACK packets to the sending nodes. The sending nodes transmit TCP data until they receive the ACK packets specified closed advertised

window.

4. After hand-over, the mobile host sends MobileIP registration requests to any new foreign agents (FA2) along with notifying extensions of the previous foreign agents (FA1).
5. FA2 forwards MobileIP binding update messages to FA1.
6. FA1 tunnels buffering TCP packets to FA2.
7. FA2 decapsulates the packets and transmits them to the mobile host.
8. The mobile host transmits ACK packets specifying ordinary advertising window sizes to the sending nodes.
9. Sending nodes start to transmit pending packets.

The state transitions of the base-station's TCP advertised window controls are shown in Fig. 4 and the actions performed in each state are listed in Table 4.

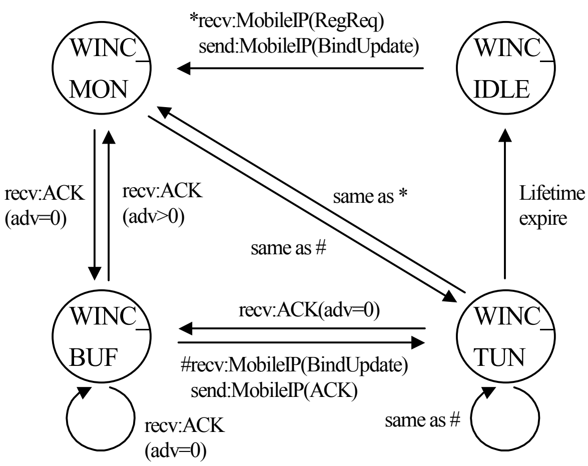

Fig4: State transitions of the base-station's TCP advertised window control.

The inclusion of buffering functions that depend on the signal strength of wireless link means that the signal strength threshold for interrupting communications can be made smaller than the case when only the mobile host specifies the closed advertised window. This raises overall throughput and allows communications to be temporarily interrupted between the mobile host and base-station.

Table4: Actions performed by base-station's TCP advertised window control

States	Action(s) performed
WINC_IDLE	Initial state before FAs receive MobileIP registration request from mobile host.
WINC_MON	When the current FA receives a MobileIP registration request, it sends binding updates to the previous FA and move to this state. When the previous FA receives these binding updates, it changes from WINC_BUF to WINC_TUN and transmits capsulated packets. Moreover, the current FAs begin to monitor
WINC_BUF	When the current FA receives an ACK specified closed advertised window, it starts buffering TCP packets for the addresses and ports of each sender and receiver. When it receives ACK packets specifying values other than zero, it transmits buffering packets to mobile hosts and returns to the WINC_MON states.
WINC_TUN	When the previous FA receives binding updates, it tunnels buffering packets destined for mobile hosts to the current FA. It sometimes transits from WINC_MON to WINC_TUN directly without receiving ACKs specified zero, and it will continue tunneling until the binding caches become invalid. Then it changes to WINC_IDLE.

4. Evaluation of the proposed control scheme

4.1. Evaluation methods

The extensions to MobileIP proposed in this study were implemented on ns-2 simulators (ns-2.1b6) developed in UCB/LBL/VINT projects [8] (Fig. 5 and 6). Node-0 was specified as an ftp sender, node-1 as a gateway between the Internet and mobile networks, node-4, 5, 6, and 7 as base-stations and foreign agents and node-8 as a mobile host and an ftp receiver. A hierarchical network structure was constructed. We assumed that the mobile host would be moving while it handed over to four base-stations one-by-one. Wireless links were simulated as IEEE 802.11.

4.2. Wireless simulators

The wireless simulators were developed by using the revised wireless mobility extensions developed in CMU Monarch projects (Fig. 6). Histories of actual mobile hosts were used to simulate statuses close to those of actual network environments. As the mobile host moved, its real-time throughput was plotted on a graph, and another graph that was recorded in advance was synchronously displayed to compare the throughputs of different methods.

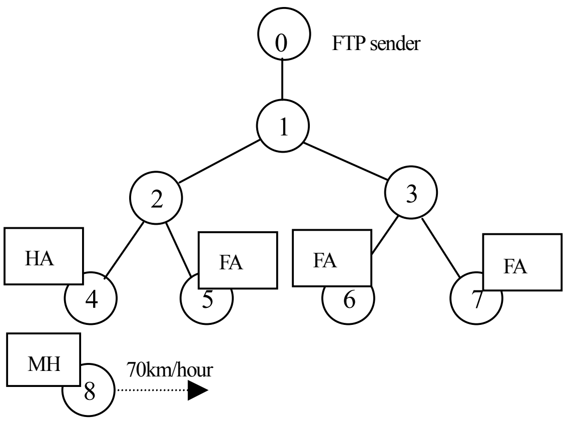

Fig. 5: Network structures of MobileIP simulation

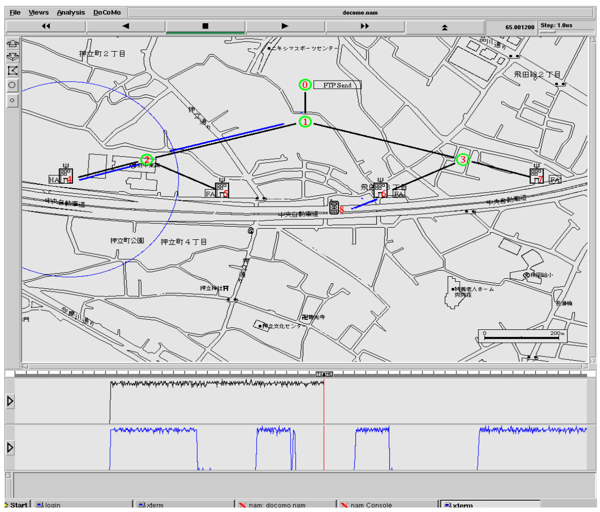

Fig. 6: Displays of MobileIP simulation

4.3. Results and discussion

We tested the following distributed implementations of TCP agents: 4.3BSD Tahoe, Reno, NewReno, Vegas and Sack. We tested them 'as is' (Figs. (a)) and with our prefetching functions of mobility agent lists (Figs. (b)) and TCP advertised window controls added (Figs. (c) for only mobile hosts' control and (d) for both mobile hosts and base-stations). Changes in throughput over time of each TCP implementation at the mobile host are shown in Fig. 7, 8, 9, 10, 11.

Packet losses at hand-over for conventional versions of Tahoe, Reno, NewReno and Sack (Figs. 7(a), 8(a), 9(a), 11(a))

In all four conventional implementations, even though the mobile host and the new base-station are connected to each other at the MAC level, registration requests are processed according to periodically issued agent advertising messages, and thus, delays in restarting communication occur. It took an average of 4.768 sec to restart communication with the new base-station. For this, the sender executed congestion window control, and the congestion window size and the threshold for the congestion avoidance phase were reduced to the minimum value.

Window control of the conventional version of Vegas and RTTs of MobileIP (Fig. 10(a))

The conventional version of Vegas, had the lowest throughput because it performs congestion window control on the basis of differences between the expected throughput values and the actual throughput calculated per packet [10]. When the mobile hosts are aparting from base-station and the delays of the wireless links were large because of increased bit error rate, the congestion avoidance phase started sooner than in other implementations. As a result throughputs increased slowly. Furthermore, the average throughput was lower after hand-over because the expected throughput values are specified at the minimum values of RTT for overall communications. For MobileIP, the routing paths and RTT depend on the host's mobility. Therefore, as RTTs increase with number of base-station hand-overs, the difference between expected and actual throughput increases. Vegas regards this situation as congestion, so the window size remains small and, as a result, throughputs do not recover.

Throughputs of link-state notifying and prefetching functions (Figs. 7(b), 8(b), 9(b), 10(b), 11(b))

When these functionalities were added to each TCP implementation, the throughput reduction after hand-over became shorter and throughput recovered faster. This is because the host transmits registration requests using agent information acquired just after base-station hand-over. Therefore, registration requests arrive at the home agent 1.537 sec sooner

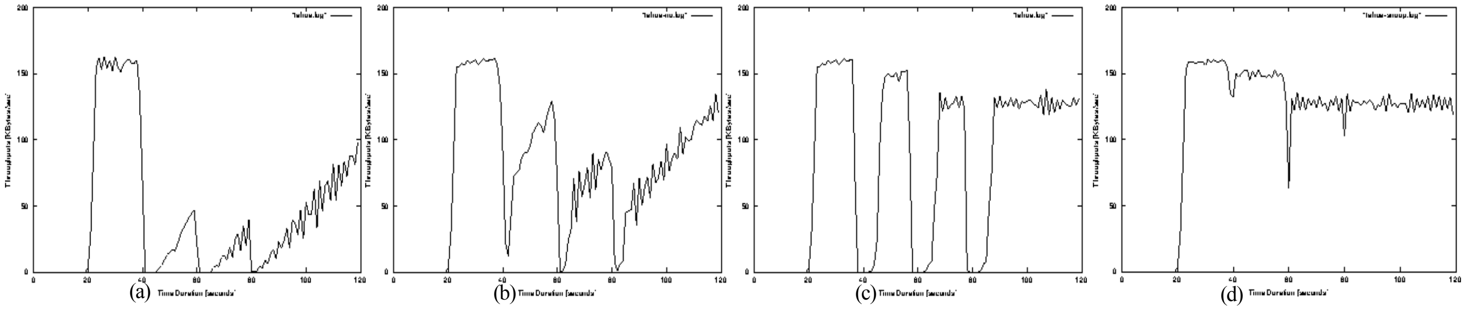

Fig. 7: Throughputs of conventional (a), link state notifying (b) and TCP advertised window control versions (c) and (d) of Tahoe impl (Kbytes/sec)

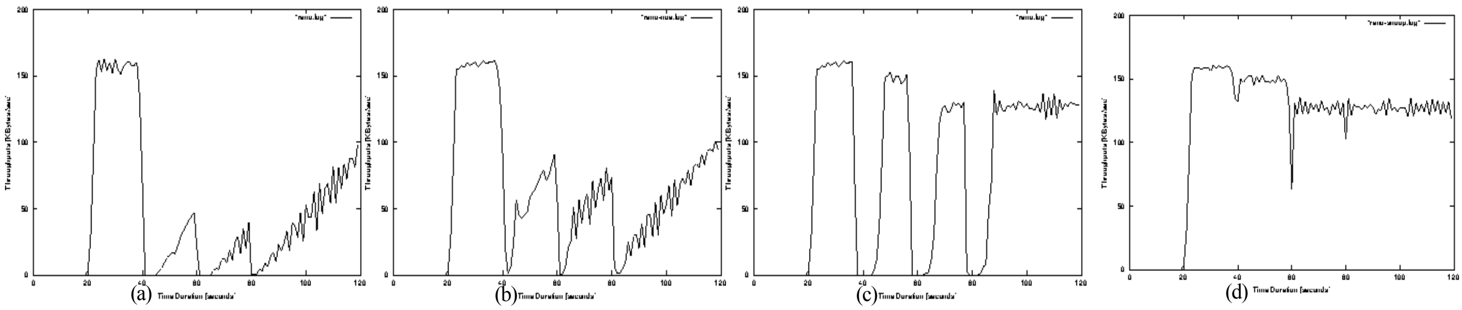

Fig. 8: Throughputs of conventional (a), link state notifying (b) and TCP advertised window control versions (c) and (d) of Reno impl (Kbytes/sec)

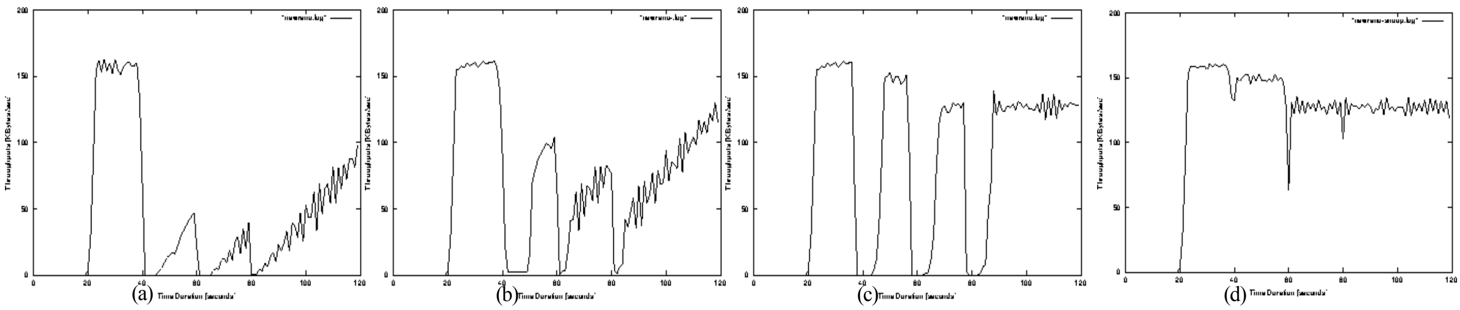

Fig. 9: Throughputs of conventional (a), link state notifying (b) and TCP advertised window control versions (c) and (d) of NewReno impl (Kbytes/sec)

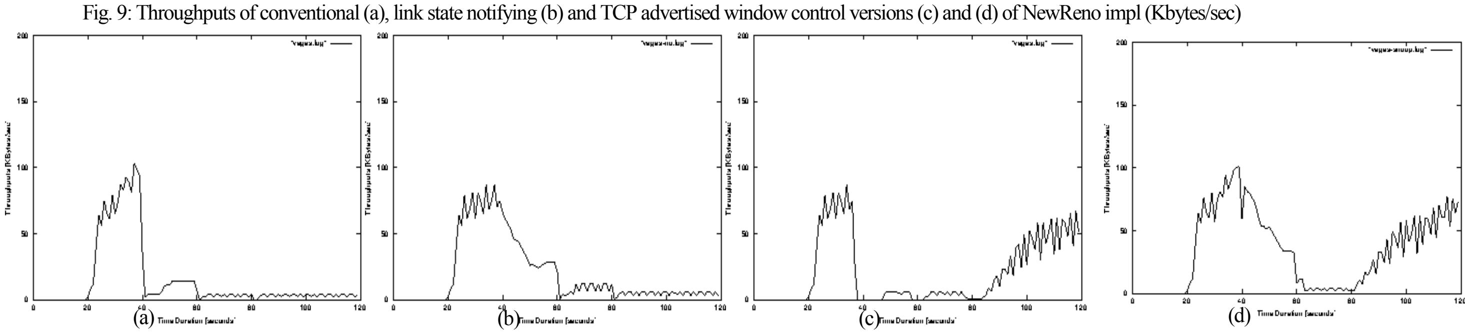

Fig. 10: Throughputs of conventional (a), link state notifying (b) and TCP advertised window control versions (c) and (d) of Vegas impl (Kbytes/sec)

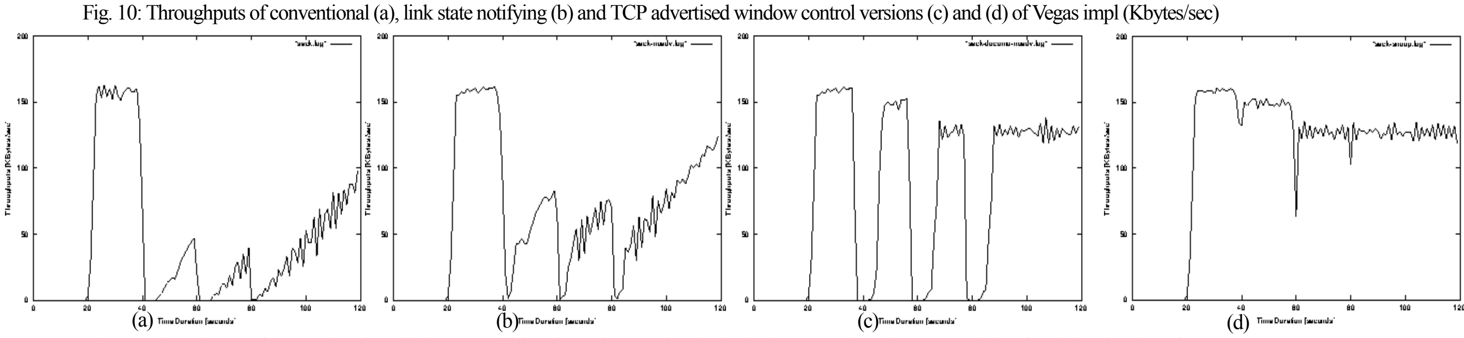

Fig. 11: Throughputs of conventional (a), link state notifying (b) and TCP advertised window control versions (c) and (d) of Sack impl (Kbytes/sec)

on average. For packets that passed through the home agent but did not reach the mobile host, it took 2.847 sec for a new base-station to be registered at the home agent in the conventional versions and only 0.814 sec in our versions. Therefore because the interruption was shorter in our versions, the slow-start phase was longer and during the slow-start phase, throughput and the size of the congestion window increased exponentially. However, when there were repeated hand-overs, the thresholds became reduced and congestion avoidance phase started quickly. During this phase, the throughput recovery time was equal to that of the conventional versions.

Comparison of throughputs of Tahoe, Reno, NewReno, Vegas and Sack with link-state notifying and prefetching functions (Figs. 7(b), 8(b), 9(b), 10(b), 11(b))

After hand-over, the throughputs of Reno (Fig. 8(b)) are lower than those of Tahoe (Fig. 7(b)). This is due to Reno's fast retransmission algorithms. In Tahoe, whenever ACKs for packets to be retransmitted are received, the next packets are consecutively retransmitted without waiting for their timeouts. However in Reno, when ACKs for packets to be retransmitted because of duplicated ACKs are received, no more lost packet is retransmitted before timeout occurs, and thresholds are set at half again. In this way, since the threshold is a half of that of Tahoe, the congestion avoidance phase starts quickly and throughput is slow to recover. The throughput of NewReno [9], after hand-over, stayed at almost zero for a certain amount of time, after which it recovered rapidly (Fig. 9(b)). This is because NewReno's extensions to Reno functioned after multiple packets in one window were lost. When duplicated ACKs were detected, only one packet in a window was retransmitted and ACKs are waited. Since NewReno repeats these actions until all packets within the window are acknowledged, especially after window size is extended as much as possible before hand-over, throughput stays low for a while on long-delay wireless links (40.8 - 50.6 sec in Fig. 9(b)). On the other hand, the throughput of Vegas (Fig. 10(b)) improved compared with that of the conventional version (Fig. 10(a)). At the first hand-over, throughput did not drop because Vegas calculates RTTs for each ACK received and then decides the window size. Just before hand-over, the link state worsened, delays increased and the congestion window size was reduced to restrain for data transmission. In this way, before the hand-over RTTs became long, so Vegas was able to predict a link-down condition and reduced the packet transmission rate, so that packet losses at hand-overs could be avoided. In Sack (Fig. 11(b)), during recovery periods, the number of packets that were transmitted in response to a received ACK is limited, so retransmission does not occur and this results in timeout. Therefore, the congestion avoidance phase started sooner than in the Tahoe implementation.

Comparison of Tahoe, Reno, NewReno, Vegas, Sack implementation of TCP advertised window control versions (Figs. 7(c), 8(c), 9(c), 10(c), 11(c))

The results of implementing link-state notifying, prefetching, and advertised window controls at mobile hosts are shown in Figs. 7(c), 8(c), 9(c), 10(c), 11(c). As the mobile host moves away from a base-station and signal strength becomes weaker, by setting the ACK-specified advertised window size to zero, packet losses caused by hand-overs can be avoided and communications can proceed without a decrease in the congestion window size or ssthresh. Thus, when signal strength is good, throughputs can be kept high. As the evaluation results, the average throughputs changed for repeated hand-over. This is because in this experiment, the RTTs of the routing path changed one-by-one, and the maximum throughput was limited by the initial window size (80 Kbytes) by using TCP window scale extension options. After three hand-overs, the throughputs of Vegas (Fig. 10(c)) increased steadily in contrast to that of the conventional version (Fig. 10(a)). This is because when our advertised window controls were implemented there were no packet on the networks, so the basement RTT was reset and reestimated. As a result, the difference between the expected throughput and measured throughput became small, and the congestion window size and throughput increased accordingly.

Comparison of Tahoe, Reno, NewReno, Vegas, Sack implementation of TCP advertised window control versions at base-stations (Figs. 7(d), 8(d), 9(d), 10(d), 11(d))

Because the buffering functions of the base-stations now depend on the signal strength of the wireless links, delays can be shortened and the duration of throughput loss can be minimized. Even in the worst case, the reduction only occurs between hand-over and the time at which the binding updates sent by the mobile host reaches the previous base-station. At the second hand-over, the distance between the previous base-station (node-5) and the current one (node-6) is large on the network hierarchies and the delays are 200 msec on one way, so throughputs is reduced by almost half. Moreover, the packets are either directly transmitted from the home agents to the current base-station or are redirected through the previous base-stations via tunneling to the current base-station, so sequences of TCP packets reach their destination out of order and for this, duplicate ACKs are transmitted. A simple sequencer could be added to the base-stations to solve this problem. We will evaluate this functionality in the near future.

Comparison of the average throughputs of each implementation (Fig. 12)

The average throughputs of the conventional, link-state notifying and TCP advertised window control versions of each implementation (Kbytes/sec) are compared in Fig. 12. Compared with the conventional versions, the addition of link-state notifying functions increased throughputs by 48.6%. Adding the TCP advertised window control to the mobile host increased throughput by 79.1% and adding it to both the mobile host and the base-station improved it by 157.5% on average.

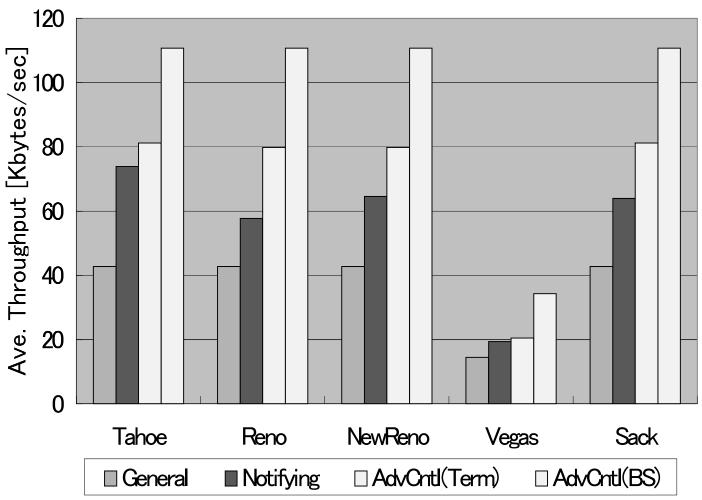

Fig. 12: The average throughputs of conventional(in the left-side), link-state notifying(middle left) and TCP advertised window control versions (middle right for only terminal and right for both terminal and base-station) of each implementation (Kbytes/sec).

5. Summary and future works

In this study, during base-station hand-over, the TCP/IP layers are informed of the link-state and trigger the IP mobility controls. Practically, even if signal strength is still weak, the foreign agent lists are maintained at the mobile hosts in advance. This allows the mobile host to switch foreign agents promptly responding to base-station hand-over. The simulation results show that packet routing errors and packet losses can be avoided. We also introduced TCP advertised window controls to provide functionality for interrupting the data transmission according to the signal strength of wireless link. Communications can be temporarily interrupted at hand-over without having to shrink the size of the congestion window and, as a result, the throughputs of overall communication can be improved. In particular, for continuous data communications while switching base-stations, communication throughputs have been improved by almost 160% with generally used Reno TCP algorithm. In the near future, information about base-station location, network bandwidth and delay will become the topology data for ns-2, utilization statistics and communication duration time the traffic data, and the location and moving history of mobile terminals the scenario data. We plan to evaluate various functions of MobileIP that have been proposed at IETF in situations that are close to those found in actual mobile communication-network environments.

Acknowledgements

We thank the members of Mizuno and Sato Labs of Shizuoka University for useful discussions. We also thank Hirotaka Nakano, Leader of Multimedia Labs of NTTDoCoMo Inc., Osamu Takahashi and the members of the 4th Generation Internet Laboratory for their significant opinions.

References

[1] C.Perkins. IP Mobility Support for IPv4. revised, Draft RFC. Jan.2000.

[2] D.Johnson, C.Perkins. Mobility Support in IPv6. Draft RFC. April.27th,2000.

[3] A. Campbell, J.Gomez, C-Y.Wan, S.Kim, Z.Turanyi, and A.Valko. Design, Implementation and Evaluation of CellularIP. IEEE Personal Communications, August 2000.

[4] R.Ramjee, T.La Porta, S.Thuel, K.Varadhan, and L.Salgarelli. HAWAII: A Domain Based Approach for Supporting Mobility in Wide-area Wireless Networks. ICNP99.

[5] Y.Xu, R.Bhalla, E.Campbell, K.Freter, E.M.Hadwen, and G.Dommety. MobileIP Based Micro Mobility Management Protocol in the The Third Generation Wireless Network, 1999. Draft-ietf-mobileip-3gwireless-ext-00.txt

[6] R.Caceres, V.N.Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. MOBICOM'96.

[7] C.Castelluccia, L.Bellier. Hierarchical Mobile IPv6. ACM Mobile Computing and Communication Review (MC2R) -April 2000 issue.

[8] ns Notes and Documents. VINT Project, Feb.25th, 2000. http://www-mash.CS.Berkeley.EDU/ns/ns-documentation.html.

[9] S.Floyd, T.Henderson. RFC2582. The NewReno Modification to TCP's Fast Recovery Algorithm.

[10] L.Brakmo, S.O'Malley, L.Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. ACM SIGCOMM94.