SCTP Based Framework for Mobile Web Agent

Yong-Jin Lee[1], Ho-Sang Ham[2], and M. Atiquzzaman[3]

[1] Department of Technology Education,
Korea National University of Education, 363-791, Korea
yjlee1026@daum.net
[2] Electronics and Telecommunication Research Institute,
161 Kajong-Dong, Yusong-Ku, Taejon 305-700, Korea
hsham@etri.re.kr
[3] School of Computer Science, University of Oklahoma,
200 Felgar Street, Norman, OK 73019, USA
atiq@ou.edu

Abstract. The stream control transmission protocol (SCTP) is a new transport protocol, which provides multi-streaming and multi-homing features. Especially, recent SCTP extensions with dynamic address reconfiguration supports transport layer mobility. We address web agent framework supporting seamless transport layer mobility in the wired and wireless environment. Our proposed framework for mobile web agent deploys SCTP with dynamic address reconfiguration. Mean response time is an important performance measure of web agents. Our simulation results show that our SCTP based framework reduces the mean response time of a typical transmission control protocol (TCP) based framework remarkably.

1 Introduction

Hyper text transfer protocol (HTTP) is an application protocol used by the web agent to retrieve objects in the Internet. Since HTTP is a connection-oriented protocol, it currently uses transmission control protocol (TCP) as the transport protocol. HTTP/1.0 does not provide the means to request multiple objects, thus, we must establish a new TCP connection for retrieving each object from the server. This protocol is particularly inefficient because it requires two extra round trip times (RTT) in setting up a new TCP connection between the client and the server. HTTP/1.1 can reduce the extra setup time with persistent connections. Furthermore, it allows the client to send all requests simultaneously by using pipelining. However, even though we use any other enhanced HTTP versions including HTTP/1.1, there is a mismatch between the requirements of HTTP and the functionality provided by TCP. When multiple embedded objects are being transferred using HTTP, it is desired that each object should be reliably transferred in the aspect of TCP. However, ordered delivery of these objects is not a requirement in HTTP. Instead, it is more important to reduce the perceived latency of user. In fact, most users are only concerned about the quick response time.

The Stream Control Transmission Protocol (SCTP) [1] has been proposed by IETF to overcome deficiencies of TCP such as performance degradation, head-of-line (HOL) blocking, and unsupported mobility. The performance degradation problem is alleviated in SCTP by incorporating the enhanced features of TCP congestion control

P. Yolum et al.(Eds.): ISCIS 2005, LNCS 3733, pp. 13 – 22, 2005.

schemes. For example, fast retransmit algorithm, based on selective acknowledgement (SACK), is deployed. This scheme speeds up loss detection and increases the bandwidth utilization [2]. The use of SACK is mandatory in SCTP as compared with TCP. To overcome the HOL blocking problem of TCP, we can make use of SCTP's multi-streaming feature to speed up the transfer of web objects. If one object is lost during the transfer, the others can be delivered to the web agent at the upper layer while the lost object is being retransmitted from the web server. This results in a better response time to users with only one SCTP association for a particular HTML page. Finally, to deal with the unsupported mobility problem of TCP, we utilize the extended SCTP multi-homing feature. SCTP multi-homing allows a single SCTP endpoint to support multiple IP addresses. In its current form, multi-homing support of SCTP is only for redundancy. The extended SCTP multi-homing feature (called dynamic IP address reconfiguration [3]) is capable of supporting transport layer mobility. This feature provides a mechanism that allows an SCTP endpoint to dynamically add and delete IP addresses during the lifetime of an SCTP association. Mobile SCTP [4] and transport layer seamless handover (SIGMA) [5,6] are schemes to utilize the dynamic IP address reconfiguration. These schemes do not require any modification to the IP structure.

While SCTP can solve several deficiencies of TCP, it does not provide the location management function that Mobile IP supports intrinsically. Hence, location management in SCTP is performed with the help of the domain name server (DNS) in the application layer [5] or the mobile IP in the network layer [4]. However, the scheme to use DNS can cause scalability problem; on the other hand, the scheme used in mobile IP can result in complexity and inefficiency in the network. In this paper, we consider a web agent that always initiates the connection setup to web server. Thus, we do not consider the location management problem.

We propose a web agent framework using SCTP with the dynamic IP address reconfiguration. The most important performance measure in web environment is the mean response time between HTTP requests and replies. To compare the performance of the proposed framework and a typical TCP based framework, we have carried out the simulation on our experimental testbed. Results show that our framework's mean response time is less than the TCP based framework by more than ten percents.

The main contribution of this paper are: (i) proposing an architecture for the mobile web agent framework, (ii) description of functions of the managers in the web agent framework, and (iii) demonstration of performance enhancement of the suggested SCTP based framework over the typical TCP based framework.

The rest of the paper is organized as follows. We begin by describing the suggested framework for web agent in Section 2. Section 3 compares mean response time of TCP based framework with that of SCTP based framework. Section 4 discusses the performance evaluation, and Section 5 concludes the paper.

2 Mobile Web Agent Framework Based on SCTP

We present the framework for the mobile web agent in Fig. 1. The framework is mainly composed of the mobility support manager and the data manager. Components of the mobility support manager are movement detector and handover manager. Data

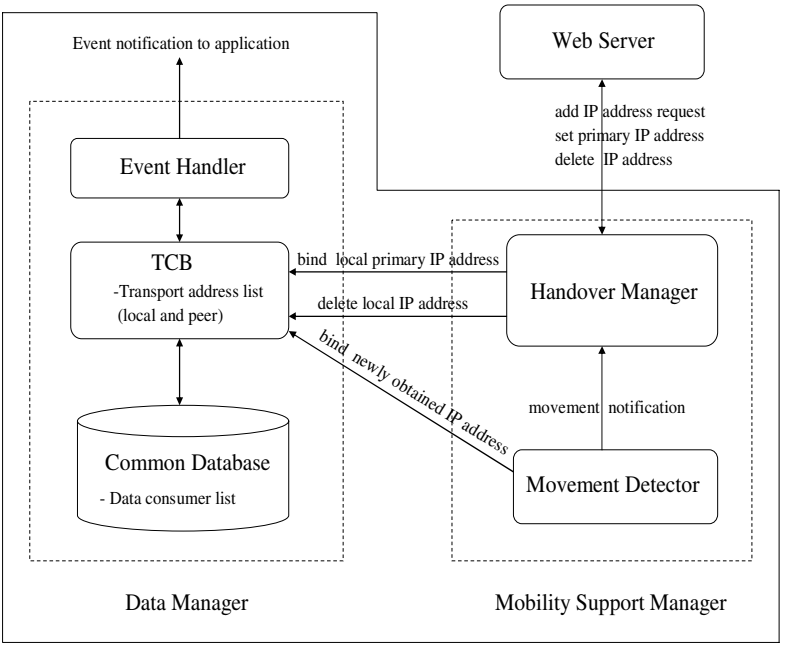


Fig. 1. Mobile web agent framework

manager includes event handler, transmission control block (TCB), and common database. Event handler passes events occurred in mobility support manager to application via TCB. Common database and TCB maintain information related with the current association and mobility support.

2.1 Data Manager

Data manager defines necessary parameters for the protocol implementation. Common database is necessary for SCTP instance as follows: (i) Data consumer list related with the currently connected association. Data consumer means process identification information, such as file descriptor, pipe pointer, and table pointer. (ii) Secret key for the security of end-user. (iii) Address list indicating end points. (iv) Port number indicating the bound port number of end point.

Some important parameters are stored in TCB per association as follows: (i) Peer verification tag indicating the authentication value of the corresponding node (ii) My verification tag indicating the authentication value of local node (iii) State indicating the current status of SCTP such as the connection complete, shutdown, and stop (iv) Peer transport address list indicating the transport address list of the corresponding node (v) Local transport address list indicating the local IP address list (vi) Primary path indicating the primary destination address of the corresponding node.

2.2 Mobility Support Manager

Mobility support manager deals with the seamless transport mobility of web agent using the SCTP address configuration change (*ASCONF*) extension [6]. ASCONF extension defines the new IP address insertion (*add_ip_address*), the old IP address deletion (*delete_ip_address*), and primary IP address change (*set_primary_address*) chunks. According to the receipt of the above chunks, data manager changes the peer transport address list, local transport address list, and primary address stored in the TCB dynamically.

Generally, the mobility support functions in the wireless mobile network include movement detection, handover management, and location management. Movement detection is a function of mobile node (MN) to identify and trace its own location change. Location management is a function of correspondent node (CN) to trace the current location of MN in order to initiate the connection establishment with MN. Handover management is function of both MN and CN to provide the roaming MN with the seamless handover.

In this paper, we consider the web environment, where MN (web agent) always initiates connection setup to CN (web server). Thus, CN does not need location management. Nevertheless, if the location management function is necessary, we can add it into the mobility support manager in Fig. 1. Mobility support procedure is depicted in Fig. 2.

(1) Movement detector
Initially, mobile web agent hears router advertisement (RA) from old access router (AR), and finds the network prefix included in RA (1, Fig. 2). Web agent acquires its own IP address by using stateless auto-configuration of IPv6 based on the network prefix (when using IPv6) or inquiring to dynamic host configuration protocol (DHCPv4/v6) server (when using IPv4/IPv6). Web agent and web server establish the association by exchanging IP address. At this time, each end point specifies the primary IP address on which data is sent (2, Fig. 2). Information related with the association is recorded in each TCB of web agent and web server, followed by exchange of data.

The web agent, while communicating with the web server, moves from the coverage of old AR to the overlapped region which is covered by both old AR and new AR. Web agent hears new router prefix from new AR (3, Fig. 2), and detects its movement into new network by comparing its current network prefix (1, Fig. 2) with new network prefix (3, Fig. 2). If web agent uses IPv6, it can itself configure new IP address using stateless auto-configuration based on the network prefix. Otherwise, it can acquire a new IP address from the DHCPv4/v6 server (4, Fig. 2), which increases the required signaling time. Anyway, newly obtained IP address is bound on the local transport address list in the TCB of web agent (5, Fig. 2). These events are delivered to the application via event handler.

(2) Handover manager
After binding the new IP address on TCB, web agent informs the web server that it will use the new IP address by sending *ASCONF add_ip_address* (6, Fig. 2). Web server modifies its own TCB by adding the received new IP address of web agent and replies to the web agent by an *ASCONF add_ip_ack* (7, Fig. 2). At this time, web

agent becomes multi-homed, and is thus reachable by two different networks. It can receive data on both old and new IP addresses. Consequently, if there is a physical problem with the path related to the primary address, the new IP address can be used as an alternate address.

As web agent leaves the overlapped region and enters the coverage of new AR, it experiences more packet loss on the primary path. If the amount of received packets on new IP address is greater than on the primary IP address, web agent sends out the *ASCONF set_primary* chuck to web server. This makes the web server to use the new IP address as primary address for data communications (8, Fig. 2). Web server replies to the *ASCONF set_primary_ack* chunk to web agent (9, Fig. 2).

As the web agent continues to move into the core coverage of new AR, the previous primary IP address becomes obsolete. Web agent sends out the *ASCONF delete_IP* chuck to web server, which eliminates the previous primary IP address (10, Fig. 2). The reason to delete the obsolete IP address is as the follows: We assume that the newly set primary path is broken in the coverage of new AR. If we did not delete the previous primary IP address in the binding list, it might become an alternate path. Thus, the data from web server may be redirected to the alternate path. However, the previous primary IP address cannot receive any data in the coverage of new AR. As a result, there exists the unnecessary traffic in the network. Handover is completed by the web server when responding by *ASCONF delete_ip_ack* to web agent (11, Fig. 2).

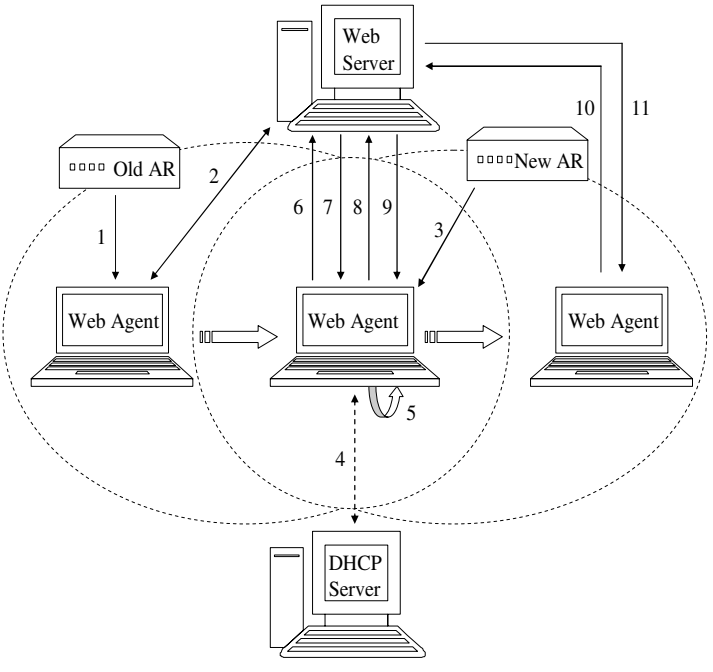


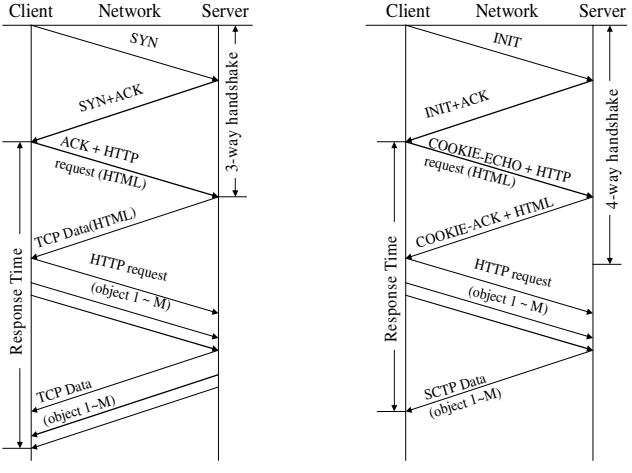
Fig. 2. Mobility support procedure of web agent

3 Mean Response Time for TCP and SCTP Based Framework

In this section, we describe TCP based framework and the SCTP based framework. We use HTTP 1.1 on both frameworks (HTTP hereafter refers to HTTP 1.1).

3.1 Mean Response Time for TCP Based Framework

In the TCP based framework, timeline of web agent using pipelining is depicted in Fig. 3 (a). Response time is defined as the time taken between the initial request from the client to the server, and the completion of the response from the server to the



(a) TCP based framework (b) SCTP based framework

Fig. 3. Timeline of web agent in TCP and SCTP based frameworks

```
Procedure web agent for TCP based framework
Variable M: the number of objects imbedded in the HTML file
         t_start: start time of response time
         t_stop: finish time of response time
         response_time: total response time
Begin
     Create socket interface and open connection to web server;
     Start timer(t_start
     Send HTTP request for HTML file to web server;
      begin
           Wait for all the threads to be created are terminated;
            for all k such that k=1,2,..,M do
                Create thread to send HTTP request for object
                and receive reply to and from web server
            end for
      end
     Stop timer(t_stop);
     Set response_ time = t_stop – t_start;
End
```

Fig. 4. Web agent for TCP based framework

client. After 3-way-handshake, client receives the HTML file with *M* embedded objects. Then, the client that supports persistent connections and pipelines sends its requests. That is, the client sends *M* requests simultaneously using pipelining. A server sends its responses to those requests in the same order that the requests were received.

In order to compare TCP based framework with SCTP based framework fairly, we have to use the same test environment such as operating systems, hardware, and programming language. Thus, we use the same approach for TCP based framework and SCTP based framework. Web agent for TCP based framework is represented in Fig. 4: we simulate the pipelining using *M* threads that send its own request simultaneously. Meanwhile, client receives HTTP reply sequentially from the server.

3.2 Mean Response Time for SCTP Based Framework

In the SCTP based framework, the initialization of a SCTP association is completed after the exchange of four messages. The passive side of the association does not allocate resources for the association until the third of these messages has arrived and been validated. Last two messages of the four-way handshake can already carry user data. With this piggybacking, SCTP has the same connection-establishment delay as TCP, namely one round trip time. Since SCTP has multi-streaming feature, it avoids the head-of-line blocking. Furthermore, SCTP does not limit maximum number of objects for pipelining. We depicted the timeline of web agent for SCTP based framework in the Fig. 3 (b).

```
Procedure web agent for SCTP based framework
Variable  M: the number of objects imbedded in the HTML file
          t_start: start time of response
          t_stop: finish time of response time
          response_time: total response time
Begin
   Create socket interface and open connection to web server;
   Start timer(t_start);
   Send HTTP request for HTML file to web server;
   Fork the current process;
   If the process is parent for sending
      begin
        Wait for all the threads to be created are terminated;
        for all k such that k=1,2,..,M do
            Create thread to send HTTP request for object
            to web server
        end for
      end
    else if the process is client for receiving
      begin
         Wait for all the threads to be created are terminated;
         for all k such that k=1,2,..,M do
             Create thread to receive HTTP reply for object
              from web server;
         end for
      end
   end
   Stop timer(t_stop);
   Set response_ time = t_stop – t_start;
End
```

Fig. 5. Web agent for SCTP based framework

Web agent for SCTP based framework is shown in Fig. 5: we first fork two processes- parent and child. In the parent process, *M* threads simulate the pipelining for sending *M* HTTP requests for objects to web server. In the child process, *M* threads simulate the multi-streaming for receiving *M* HTTP replies from web server. To summarize, main difference between TCP based framework and SCTP based framework is that SCTP can receive multiple objects in parallel by using its own multi-streaming feature.

4 Performance Evaluation

4.1 Experimental Setup

In this section, we compare the performance of web agents for TCP based framework with SCTP based framework in terms of mean response time. For this experiment, we wrote two Linux C server programs which simulate HTTP over TCP and SCTP server, respectively. We also wrote two Linux C client programs according to procedures which are presented in Figs. 4 and 5 to simulate pipelining and multi-streaming, respectively. The main reason for the simulated HTTP server and client using our own program is due to the lack of adequate support of SCTP by current HTTP client/server. To simulate TCP based framework, each web agent sends requests for *M* objects. In response, the TCP server sends the requested *M* objects to the web agent sequentially using one TCP connection. On the other hand, SCTP server sends *M* objects in different streams on a single SCTP association. Table 1 shows the host and network configurations of the testbed. We used the CISCO-7102 router to control the bandwidth between web agent and web server. We measured the response time using *t_start*() and *t_stop*() system calls defined in *timval* structure of Unix system.

Table 1. Host and network configuration of testbed

Node	hardware	software	Operating system	network
Web Server	CPU: Intel Pentium-4/1.7 GHz, RAM: 512 MB NIC: 3 Com PCI 3cDSOHIO 100-TX	TCP /SCTP server program	Fedora Core 2 Linux Kernel 2.6.5-1	210.93.87.0
Web Agent	CPU: Intel Pentium-4/1.7 GHz, RAM: 512 MB NIC: 3 Com PCI 3cDSOHIO 100-TX	TCP /SCTP client program	Fedora Core 2 Linux Kernel 2.6.5-1	210.93.83.0
Router	Cisco-7102		IOS	

Table 2 represents test parameters used for our experiment. For example, to investigate the multi-streaming effect of SCTP over TCP according to the object size (*O*), we used 8 KB, 10 KB, and 13.5 KB. In this case, the number of objects (*M*) and bandwidth (*bw*) are 10 and 2 Mbps, respectively.

Table 2. Test parameters for the experiment

test parameters	object size (KB)	number of objects	bandwidth (Mbps)	round trip time (ms)
object size (*O*)	8, 10, 13.5	10	2	-
number of objects (*M*)	13.5	3, 5, 10	2	-
Bandwidth (*bw*)	13.5	10	0.064, 2, 8	-
round trip time (*RTT*)	13.5	10	2	55, 80, 256

4.2 Experiment Result

Fig. 6 ~ Fig. 9 shows mean response times corresponding to test parameters in Table 2. Mean response times are computed for 50 trials at each test parameter for TCP based framework and SCTP based framework, respectively. Fig. 6 shows that the mean response time is directly proportional to the object size. This is due that the transfer time is increased in proportional to the file size. In Fig. 7, the difference of mean response time between TCP based web agent and SCTP based web agent is increased as the number of objects becomes larger. This means that large number of objects in SCTP based framework can reduce the mean response time of TCP based framework. Fig. 8 shows that the mean response time is inversely proportional to the bandwidth. It is natural that larger bandwidth decreases total transfer time, which reduces the mean response time. In Fig. 9, we found that the increase rate of the mean response time in TCP based framework is about 100 % between 55 ms and 256 ms, meanwhile, which in SCTP based framework is only 30 %. Since TCP can not utilize the multi-streaming feature, all objects experience large *RTT*. On the other hand, multiple objects in SCTP based framework experience only one *RTT*. Thus, as the number of objects and *RTT* are increased, the performance benefit of SCTP over TCP will be increased.

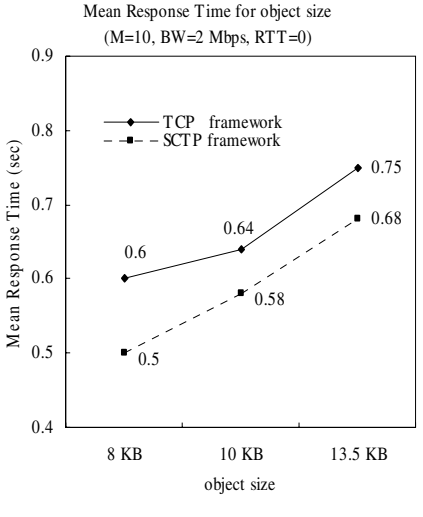


Fig. 6. Mean response time for object size

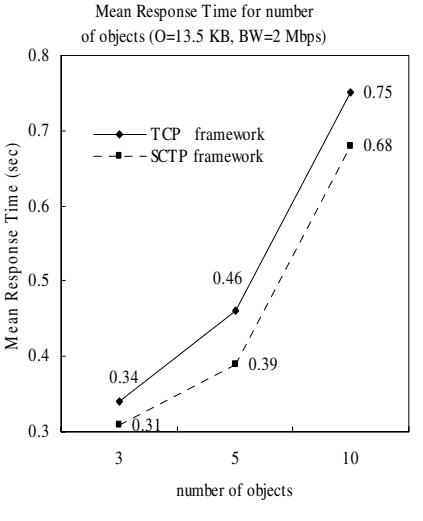


Fig. 7. Mean response time for number of objects

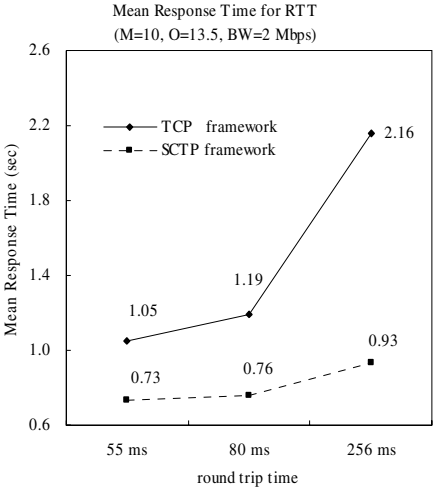


Fig. 8. Mean response time for bandwidth

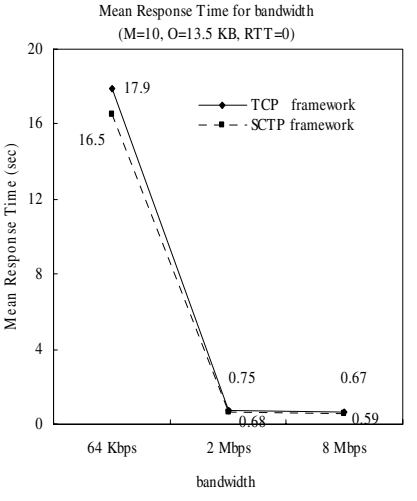


Fig. 9. Mean response time for round trip time

5 Conclusions

In this paper, we have proposed and evaluated a stream control transport protocol based web agent framework. We have also investigated and described important characteristics and functions necessary to implement the web agent to support seamless mobility. To compare the performance of our SCTP based framework with typical TCP based framework, we have carried out experiment in testbed. Results show that our web agent framework can reduce the mean response time over a typical TCP based framework remarkably. Future extension of this work includes performance evaluation in a real wireless mobile environment.

References

1. Caro, A., Iyengar, J., Amer, P., Ladha, S., Heinz, G. and Shah, K.: SCTP: A Proposed Standard for Robust Internet Data Transport. IEEE Computer. (2003) 20-27.
2. Fu, S., Atiquzzaman, M.: SCTP: State of the Art in Research, Products, and Challenges. IEEE Communication Magazine. (2004) 64-76.
3. Stewart, R. et al.: Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. IETF Internet draft, draft-ietf-tsvwg-addip-sctp-06.txt. (2003).
4. Koh, S., Chang, M., Lee, M.: mSCTP for Soft Handover in Transport Layer. IEEE Communication Letters, Vol. 8. (2004) 189-191.
5. Fu, S., Ma, L., Atiquzzaman, M., Lee, Y.: Architecture and Performance of SIGMA: A Seamless Mobility Architecture for Data Networks, IEEE International Conference on Communi- cations (ICC), Seoul, Korea, May 16-20. (2005).
6. Sivagurunathan, S., Jones, J., Atiquzzaman, M., Fu, S., Lee, Y.,: Experimental Comparison of Handoff Performance of SIGMA and Mobile IP, IEEE Workshop on High Performance Switching and Routing, Hong Kong, May 12-14. (2005).