End-to-End Mobility and Robust IP Soft Handover

Hosei Matsuoka   Takeshi Yoshimura Tomoyuki Ohya
Multimedia Laboratories, NTT DoCoMo
3-5, Hikari-no-oka, Yokosuka, Kanagawa, 239-8536, Japan
E-mail: matsuoka@nttdocomo.co.jp

Abstract- We describe the design and implementation of a new transport layer protocol MMSP (Mobile Multimedia Streaming Protocol) that realizes end-to-end mobility and robust IP soft handover; its target is to greatly increase the quality of multimedia streaming applications over wireless networks. We address two main causes of quality degradation in the mobile environment. One is the frequent movement of mobile terminals between radio access points. The other is the inherently high bit error rates of wireless links. MMSP supports multihoming and bicasting in combination with FEC. These mechanisms enable mobile terminals to move without any data loss and also improve error resiliency against wireless errors. Implementing the architecture requires no changes to the existing IP networks so it allows for easy deployment. Our performance experiments show that while FEC processing causes some slight delay, such delay is allowable and bicasting with FEC can provide high quality multimedia streaming even under high bit error rates.

1. INTRODUCTION

The mobile Internet allows most of the mass-market services available on fixed Internet terminals to be accessed through mobile terminals. Multimedia streaming is a key goal of 3G and future wireless networks, so streaming clients will soon be deployed in advanced mobile terminals. Multimedia streaming over wireless networks, often called mobile multimedia streaming, allows music, movie and news services to be accessed regardless of location or time. Consumers can access multimedia content at any time.

Current mobile terminals, however, fail to well support mobile multimedia communication due to the high packet loss rates of the wireless networks. Packets passing over wireless links are more susceptible to loss than those passing over wired links. There are two main causes of packet loss in wireless networks. One is the frequent movement of mobile terminals between radio access points. When a mobile terminal moves from one radio access point to another, handover occurs. If the new access point is associated with a new subnet, a change in routing reachability may occur. Mobile IP [1] tackles this problem and offers IP handover in the network layer level, but since communication may be interrupted during handover, packets can dropped. The other is that most wireless links suffer severe fading, noise and other interference factors, and so have relatively high bit error rates.

We address these two problems. To eliminate packet loss during handover, we employ a packet path diversity scheme [2] and develop an end-to-end bicasting mechanism that enables IP soft handover. To offset wireless errors, we employ an FEC (Forward Error Correction) scheme [3] and embed it in the bicasting mechanism.

We propose a new transport layer protocol, MMSP (Mobile Multimedia Streaming Protocol), that supports multihoming and bicasting in combination with FEC. Our main design concept is that the existing IP network should not be changed. Changing the existing network hinders wide and rapid deployment. For example, IPv6 has technically much value and potential, but its deployment is rather slow. Therefore, we decided not to require alteration of the existing IPv4 network or the forthcoming IPv6 network. Adding this new protocol to the current transport layer will allow future networks to support a much wider variety of applications.

2. RELATED WORKS

This section describes the basic technologies for reliability and related works on mobility.

2.1 Reliability

Wireless links have high and time-varying bit error rates, so some error control mechanism is needed if we are to provide high quality mobile multimedia streaming. ARQ (Automatic Repeat reQuest) [4] is the most common solution. Missing packets are retransmitted upon timeout or receipt of an explicit request from the receiver. As this retransmission incurs delays, it is not always useful for streaming applications because the delay hinders interactive VCR-like functionalities such as fast forward and rewind. Although the delay of link-by-link ARQ may be acceptable, that yielded by end-to-end ARQ provided by TCP or SCTP [5] can be excessive. On the other hand, FEC techniques have advantages in terms of delay. The sender prevents losses by transmitting some redundant information, which allows the missing data to be reconstructed at the receiver. Besides recovering the missing packets without increasing latency, this approach generally simplifies both the sender and the receiver since a feedback channel is not necessary.

In the general literature, FEC refers to the ability to overcome both erasures and bit-level corruption. However, in the case of the IP network, the network layers will detect corrupted packets and discard them. Therefore the primary application of FEC codes to the IP network is as an erasure code. There are some very simple codes that are effective for recovering from packet loss. For example, one simple way to provide protection from a single loss is to partition the data set into fixed size source symbols and then add a redundant symbol that is the parity of all source symbols. Reed-Solomon codes for packet level [6] provide protection from multiple packet losses. They partition the data set into fixed size source symbols in the same way and create multiple redundant symbols that are calculated in a Galois field.

2.2 Mobility

Mobile IP is the IETF-proposed solution for realizing terminal mobility among IP subnets, and it was designed to allow a host to change its point of attachment to an IP network transparently. For IPv6, mobility support has been on the list of required features from the beginning. The Mobile IPv6 specification is on its way to becoming a standard. In both IPv4 and IPv6 networks, handover causes packet loss over some duration. Until the change in a mobile node's address (care-of-address) is notified to the correspondent terminal, traffic for the mobile node is sent to the old address and so is dropped. If the mobile node is some distance from the correspondent node, the amount of time involved in sending the binding updates may be upwards of a hundred milliseconds. This latency in routing update may cause many packets for the mobile node to be dropped at the old address router.

Hierarchical Mobile IP [7] has been proposed to solve this problem. With mobility agents in foreign networks, a change in routing within a domain is managed by the mobility agent. Correspondent nodes contain a regional or hierarchical address maintained by the mobility agent rather than the address of the mobile node. This solution reduces the duration of packet loss. Fast handover [8] also minimizes the duration of packet loss. The mobile node obtains a new address for the new access router when it still has with connectivity with the old access router. When the mobile node sends the binding update to the old access router, which then redirects the packets to the new care-of-address. When the mobile node reaches the new link, and establishes Layer 2 connectivity, it can restart the process of receiving packets. No extra delay is necessary for establishing Layer 3 connectivity because the old access router is already sending the packets to the new address. This mechanism can reduces the packet loss duration to the time taken by Layer 2 handover if the redirection and the actual movement of the mobile node are synchronized. Fast handover with simultaneous bindings and bicasting [9] works well and does not need synchronization. However, these solutions require the considerable changes to the existing IP networks.

Several end-to-end approaches [10] have been proposed as solutions. These methods require no change to the IP substrate, but instead modify the transport layer and applications at the end hosts. DNS already provides a host location service, and its ability to support secure dynamic updates [11] is normally used to locate mobile hosts as they change their network and attachment point. The IP address serves as a routing locator, reflecting the addressee's point of attachment in the network topology. The DNS provides a mechanism by which name resolvers can cache name mappings for some period of time, specified in the time-to-live field. To avoid a stale mapping from being extracted from the name cache, the time-to-live field for the mapping of the name of the mobile host is set to zero, which prevents this information from being cached. In this architecture, there is no need for an additional third-party agent. However, the packet loss duration during handover becomes considerable when the mobile node is some distance from the correspondent node.

As shown in Fig.1, current approaches create trade-off between the cost needed for changing the network and the packet loss duration during handover. Mobile IP with Fast Handover eliminates handover loss, but it requires additional functions in all access routers in the network, so network costs are increased by its deployment. The end-to-end approaches need no change in the existing network, but they suffer considerable packet loss during handover. We propose another approach to eliminate both packet loss during handover and changes to the network at the expense of doubling the connection's bandwidth during handover.

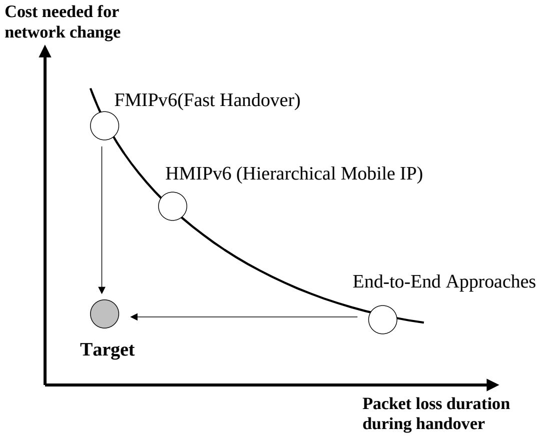
Cost needed for network change
FMIPv6(Fast Handover)
HMIPv6 (Hierarchical Mobile IP)
End-to-End Approaches
Target
Packet loss duration during handover

Fig.1. Research Target.

3. MOBILITY ARCHITECTURE

Our mobility architecture is basically an end-to-end approach and provides multihoming and bicasting mechanisms. Our bicasting mechanism is combined with FEC, but it will be useful, to begin with, to identify the goal of bicasting. The combined effect is then described.

3.1 Multihoming

Our mobility architecture supports multihoming in which more than one IP network interface can be assigned to a single endpoint. Each interface can obtain its own IP address using any address allocation mechanism, such as DHCP (Dynamic Host Configuration Protocol) and RA (Router Advertisement) on IPv6. A single connection may involve the IP addresses of multiple interfaces. The advantages of multihoming are load sharing, connection redundancy and performance improvement. Our main target of multihoming is the soft handover between two different IP addresses at the same end point.

Multihoming forces the host to choose the source and the destination address. TCP makes this choice when the connection is instantiated; SCTP may make similar choices through the life-time of the connection; UDP may make this choice either for each packet, or at the beginning of an association. We assign a priority to each source and destination address. The priority can be dynamically changed based on the layer-2 information, such as the received signal intensity and the available bandwidth. As an example, when the host has addresses associated with W-CDMA and wireless LAN links, the address associated with the wireless LAN link has higher priority because its wireless bandwidth is wider. When the host has two addresses associated with different wireless LAN links, the address associated with the link offering higher signal strength has higher priority. Hosts choose the source and destination addresses that offer the highest priority.

3.2 Bicasting

When the mobile node moves to another network attachment point during a connection, the correspondent node is directly informed of the new IP address of the mobile node. We assume that the mobile node passes through a region where two different cells overlap and two different access points can be accessed simultaneously. When the mobile node enters the overlap region and gets a new IP address, the mobile node sends a packet that requests the correspondent node to add the new IP address of the mobile node to its destination address list. When the mobile node leaves the overlap region and the old IP address becomes inactive, the mobile node sends a packet that requests the correspondent node to delete the stale IP address of the mobile node from the destination address list.

Bicasting can offer IP soft handover, since more than one access point is involved in the communication. When the mobile node is in a cell-overlap region, the mobile node may have two different IP addresses, and if the signal strengths of both links are weak, both addresses have the same low priority. In this situation, the correspondent node copies the packets and sends copies to each destination address. On receipt, the duplicated packets are discarded. If the signal strength of one link exceeds a certain threshold, the priority of the address associated with the link becomes high. The correspondent node is informed of the changes in the priority of the address and stops copying and sending packets to the other destination address. The mobile node can move from one cell to another without any interruption.

3.3 Combination with FEC

Sending exactly the same packets through different paths is not efficient in terms of increasing error resistance. Our architecture supports a simple way to improve robustness against wireless errors. When the mobile node is in an overlap region, the radio waves from both radio access points are usually weak and bit error rates may be high. If the same packet is lost on both wireless links due to bit errors, this packet cannot be received.

To improve error resiliency, our architecture Reed-Solomon codes the original packet stream and then splits the encoded stream as described below. Reed-Solomon requires byte-organized data. By extending each message from k symbols to n symbols through the addition of $(n\text{-}k)$ redundant symbols, we can detect and recover up to $(n\text{-}k)$ corrupted symbols within the extended message. Fig.2 shows the fragmentation and FEC encoding process. The transport layer fragments an application message into several segments of the same size. Padding is added to the last segment to align the segment size. The transport layer creates the same number of redundant symbols as data symbols, that is to say $n = 2k$. A transport layer header and an IP header are added to each data symbol and redundant symbol. Half of them are transmitted to one destination address and the other half are transmitted to the other destination address.

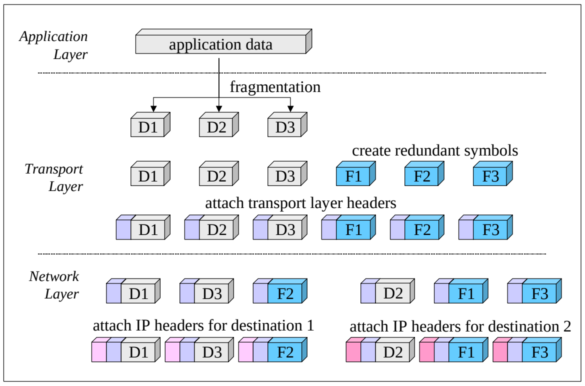

Fig.2. Fragmentation and FEC process.

In this case, any missing packets up to totally *k* on both network paths can be recovered. This means that the probability of data loss would be much lower than it would be with just copying.

4. IMPLEMENTATION

We implemented a new transport layer protocol stack MMSP in the FreeBSD 4.5 kernel. MMSP is a datagram-oriented protocol, as is UDP, and provides our mobility architecture.

4.1 Protocol Header Format

Fig.3 shows the fields in the MMSP header. Its normal size is 12 bytes, unless options are present. The port numbers identify the sending process and the receiving process. The MMSP length field (16bits) is the length of the MMSP header and the data in bytes. The MMSP checksum field (16bits) covers the MMSP header and the data in the same way as UDP. The packet type field (3bits) specifies the following type of the MMSP message: Data-packet, FEC-packet, Move-packet, or ACK-packet. The block sequence field (5bits) identifies the application message to be reconstructed from the packet. The sequence number is incremented by one for each application message. The packet sequence field (8bits) identifies the position within the block of the data packets and the calculated multiplier in GF(256) for redundant packets. MMSP uses GF(256) as the Reed-Solomon code. The block length field (16bits) identifies the length of the application message. Move-packets contain the option field. The address operation field (8bits) identifies the request types: add-address, delete-address or change-priority. The address family field (8bits) specifies the address family of the contained address. The address priority field (8bits) identifies its priority. ACK-packets are sent in response to Move-packets. Move-packets are retransmitted if they are not acknowledged when the timeout expires.

(General Header) 32bit

source port number			destination port number
MMSP length			MMSP checksum
packet type	block sequence	packet sequence	block length

(Move Option)

address operation	address family	address priority	reserved
Network Address			

Fig.3. MMSP Header Format

4.2 Mobility Control Daemon

To receive layer-2 information, we slightly modified the 802.11b device driver and the IPv6 protocol stack at the end hosts. Although MMSP is currently implemented only on IPv6, IPv4 implementation is possible. Two 802.11b devices are necessary because they are not able to receive the radio waves of multiple access points. We implemented a mobility control daemon that is running at all times and measures the radio wave strength of each 802.11b device every one second. When a new access point becomes accessible, the mobility control daemon sends a router solicitation packet. If the feedback router advertisement packet contains a new IPv6 address, it is added to the MMSP address list. The MMSP stack has its own PCBs (Protocol Control Blocks). These blocks can maintain multiple source and destination IP addresses. When an access point becomes inaccessible, the mobility control daemon removes the stale IPv6 address of the interface. Then the IP stack calls the MMSP *ctlinput* routine and removes the stale address from the PCBs. When the radio wave strength from an access point exceeds or falls below a certain threshold, the mobility control daemon makes the device generate the interruption to the kernel. This interruption calls the MMSP *ctlinput* routine and change the priority of the address. Fig.4 shows the example of the soft handover sequence. When an address is added, deleted, or its priority changes, the mobile node sends Move-packets and informs the correspondent node of the new or stale address and its priority.

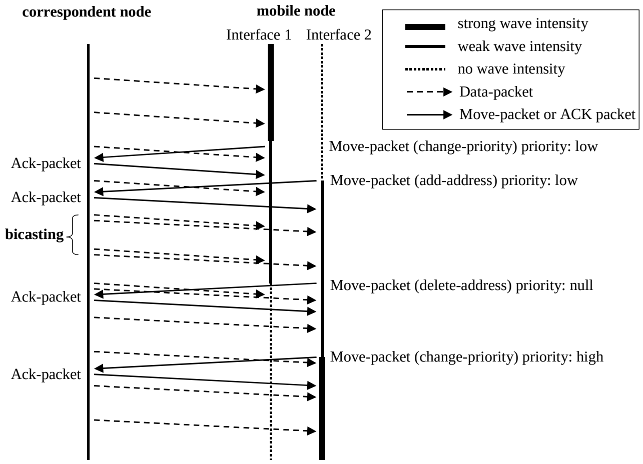
correspondent node
mobile node
Interface 1
Interface 2
strong wave intensity
weak wave intensity
no wave intensity
Data-packet
Move-packet or ACK packet
Ack-packet
Move-packet (change-priority) priority: low
Ack-packet
Move-packet (add-address) priority: low
bicasting
Ack-packet
Move-packet (delete-address) priority: null
Ack-packet
Move-packet (change-priority) priority: high

Fig.4. Soft Handover Sequence

4.3 Routing Table Searches

One of the big problems with bicasting is routing table searches. When the mobile node sends packets to the correspondent node, the mobile node searches its routing table and decides which interface to send a packet out on. The destination address is used as the search key. Therefore, even if the mobile node has two interfaces for bicasting, all the packets from the mobile node to the correspondent node would be sent through the same interface. To distribute them to each interface, the PCB maintains pairs of a source IP address and the next hop router's IP address, which is extracted from the router advertisement message. When the mobile node sends packets, it looks up the next hop router's IP address from the source IP address of the outgoing packets, and uses the next hop router's IP address to conduct routing table search and neighbor discovery [12]. MMSP can decide which interface to send a packet to by filling in the source IP address field of the packet header with the IP address of the selected interface.

4.4 User Data Fragmentation

MMSP uses path MTU metric to determine the packet size needed for FEC partitioning. An ICMP message is sent by a router in response to a packet that it cannot forward because the packet is larger than the MTU of the outgoing link. This message contains the MTU of the next-hop link. The information is passed to the MMSP stack and path MTU metric can be discovered.

MMSP allows applications to specify the FEC partitioning size. Large packets are vulnerable to bit error in wireless links thus causing relatively high packet losses. While small packets increase the protocol header overhead, it may provide better quality of multimedia streaming. MMSP provides an additional argument of *setsockopt* system call for specifying the FEC partitioning size.

4.5 Socket Interfaces

A new socket type *SOCK_MMSP* was prepared for MMSP and used in the same way as *SOCK_STREAM* for TCP and *SOCK_DGRAM* for UDP. It is easy to rewrite existing UDP applications as MMSP applications, because the MMSP socket interface is in full conformance with the UDP socket interface. All that the application developer needs to do is change the socket type of each *socket* system call.

One requirement for MMSP is using *connect* system call. Network address changes are managed in the transport layer and not passed to the application. The application fixes the destination at the beginning of an association and uses *send* or *write* system calls to send packets.

5. PERFORMANCE MEASUREMENT

We measured FEC encoding and decoding time of our implemented kernel, and evaluated the impact of FEC on the quality of MPEG-4 streaming applications

5.1 FEC Encoding and Decoding

One of the problems with FEC is the amount of calculation needed for encoding and decoding the redundant symbols. This section presents the results of a detailed performance analysis of FEC overhead. The measurements were made while performing encoding on the correspondent node and decoding on the mobile node. Both nodes were IBM PC-clones with single Intel Pentium III processors running at 1.2GHz; the FreeBSD4.5 system contained our implemented kernel.

In this experiment, application message size ranged from 1000 bytes to 8000 bytes: the fragmentation size was fixed at 500 bytes. The correspondent node created redundant packets and sent them. All the Data packets were discarded at the intermediate router in order to measure the recovery time possible with the redundant packet scheme. For example, for an application message size of 8000 bytes, the application message was split into 16 data packets and 16 redundant packets were encoded. The 16 data packets were discarded at the intermediate router, so only the 16 redundant packets were received at the mobile node. The mobile node recovered the 16 data packets by decoding the 16 redundant packets.

Fig.5 shows the correlation between the application message size and encoding/decoding time. Both encoding time and decoding time were proportional to the square of the application message size. When the application message size was 8000 bytes, FEC encoding and decoding took a total of 44.8 milliseconds. This shows that current CPUs can perform FEC encoding and decoding in a reasonable time

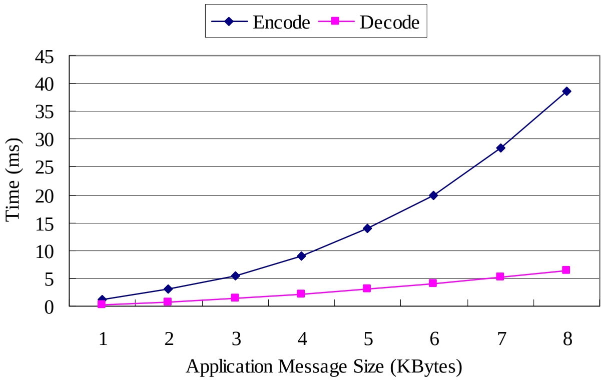

Fig.5. FEC encoding and decoding

5.2 Error Resiliency

We evaluated the effectiveness of combining bicasting with FEC. We compared the error resiliency of copy-bicasting to that of FEC-bicasting using an MPEG-4 video stream. The video content was a 60 second sequence of CIF size (352x288) frames encoded at 384Kbps. The frame rate was 15 frames/sec, and the I-picture interval was 5 seconds.

All the other frames were P-pictures. Each video packet occupied approximately 5000 bytes, in other words, the application message size was 5000 bytes. The fragmentation size was fixed at 500 bytes. For the case of copy-bicasting, the MPEG-4 stream sender sent exactly the same packets through two different network paths to the receiver. For the case of FEC-bicasting, the sender sent data packets through one network path and redundant packets through the other network path. We simulated wireless bit error rates on both network paths. Both bit error rates were the same and ranged from 10E-6 to 10E-4.

Fig.6 shows the average PSNR (Peak Signal to Noise Ratio) of all video frames for both bicasting schemes. Missing video packets cause block noise which may be propagated to the next video frame, thus degrading the video quality. For the case of copy-bicasting, the PSNR falls dramatically as the bit error rates exceeds 10E-5. On the other hand, FEC-bicasting kept the quality high.

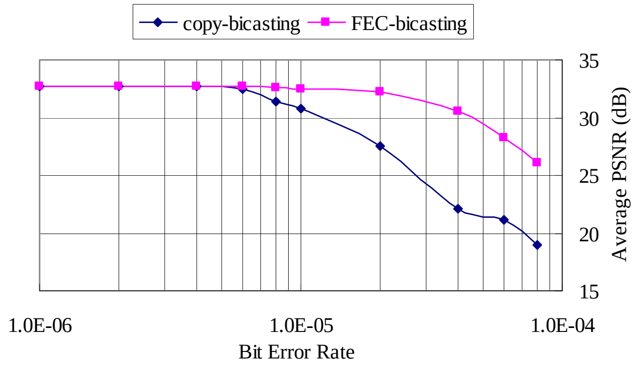

Fig.6. Error Resiliency.

6. CONCLUSIONS

We have introduced a new architecture that provides robust and fault-tolerant media transport and prevents the quality degradation that occurs with traditional architectures through the use of our new Mobile Multimedia Streaming Protocol (MMSP). Adding MMSP to the end-hosts allows mobile terminals to offer high-quality mobile multimedia communications. We actually implemented an MMSP stack in the FreeBSD 4.5 kernel. Multimedia services running on our implemented system can take full advantage of IP soft handover and the improved error resiliency achieved by MMSP without any change to the existing network.

The main disadvantage of our architecture is the doubling of connection bandwidth during handover. Our proposed bicasting mechanism can offset much higher bit error rates than copy-bicasting, thus providing high quality multimedia applications even under high bit error rates. This suggests that MMSP will contribute to reducing the energy-per-bit requirements of wireless access links. While the FEC encoding used in the architecture causes some slight delay, such delay is allowable, and the use of FEC significantly improves multimedia streaming quality.

We are currently exploring an end-to-end security mechanism for MMSP that will prevent connection hijacking

REFERENCES

[1] C.Perkins. “IP Mobility Support for IPv4,” RFC3220, January 2002.

[2] Y. J. Liang, E. G. Steinbach, and B. Girod, "Multi-stream Voice over IP Using Packet Path Diversity", Proceedings IEEE Fourth Workshop on Multimedia Signal Processing, pp. 555-560, Cannes, France, Oct. 2001

[3] J-C.Bolot, S.Fosse-Parisis, and D.Towsle. Adaptive FEC-Based error control for Internet Telephony. In Proceedings of Infocom’99, March 1999.

[4] R.J.Benice, A.H.Frey and Jr, “An Analysis of Retransmission Systems,” In Proceedings of IEEE Transaxtions of Communication Technology, COM-12, pp.135-145, 1964.

[5] R.Stewart, Q.Xie, K.Morneault, C.Sharp, H.Schwarzbauer, T.Taylor, I.Rytina, M.Kalla, L.Zhang and V.Paxson, “Stream Control Transmission Protocol,” RFC2960, 2000.

[6] J.Plank, “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems,” In Software Practice and Expertence, Vol.27, pp.995-1012, May 1990.

[7] H.Soliman, C.Castelluccia, K.El-Malki and L.Bellier, “Hierarchical MIPv6 mobility management (HMIPv6),” draft-ietf-mobileip-hmipv6-05.txt, July 2001.

[8] G.Dommety, A.Yegin, C.Perkins, G.Tsirtsis, K.El-Malki, and M.Khalil, “Fast Handovers for Mobile IPv6,” draft-ietf-mobileip-fast-mipv6-04.txt, March 2002.

[9] K.El-Malki and H.Soliman, “Simultaneous Bindings for Mobile IPv6 Fast Handoffs,” draft-elmalki-mobileip-bicasting-v6-02.txt. June 2002.

[10] Alex C.Snoeren and H.Balakrishnan, “An End-to-End Approach to Host Mobility,” In Proceedings of 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’00), 2000.

[11] D.Eastlake, “Secure domain name system dynamic update,” RFC2137, April 1977.

[12] T. Narten, E.Nordmark, and W.Simpson, “Neighbor Discovery for IP Version 6 (IPv6),” RFC2461, December 1998.