Handoff-aware Adaptive Media Streaming in Mobile IP Networks

Dongwook Lee[§,‡], JongWon Kim[†], and Prasun Sinha[‡]

[§]Ultrafast Fiber Optic Research Center and [†]Networked Media Lab.
Dep. of Information and Communications, Gwangju Institute of Science and Technology (K-JIST), Gwangju, Korea,
[‡] Dept. of Computer Science and Technology, Ohio State University, Ohio, USA.

Abstract. Streaming media applications under the mobile IP networks suffer from playback disruptions resulting from handoff blackout period as well as fluctuation in the available bandwidth. To overcome possible shortage of buffer, pre-buffering techniques can be adopted where the client stores sufficient part of the stream in advance. However, under the mobile IP handoff situation that may take up to several seconds, it is extremely difficult to sustain seamless playback. Inaccurate and conservative choice on the required buffering margin can waste limited latency budget, resulting in overall quality degradation. Thus, in this paper, we introduce a scheme that helps estimate the required pre-buffering size more accurately by considering both handoff duration and transient packet losses. Experiments and network simulation results show that the proposed scheme can provide an appropriate guidelines on buffer parameters and thus facilitate seamless streaming over the mobile IP network.

1 Introduction

Mobile and wireless technologies have accelerated wide-spread adoption of multimedia services to mobile computers. In streaming applications, media streams have to be transmitted continuously, overcoming the fluctuations of network resources. The delay, jitter, and busty packet losses are usually addressed by adopting sufficient buffering at the clients prior to playout [1]. This "pre-buffering" smoothes network variations and gives a retransmission opportunity for lost packets. In mobile networks, available bandwidth is scarce and fluctuating. In addition, transmission itself is paused when a handoff occurs. In fact, several mechanisms are proposed to reduce the blackout period due to handoff [2]. However, these schemes are limited since they require special arrangements such as MAC bridge, additional resources, and corresponding signaling. Even with the fast handoff [5] and the smooth handoff [6], in which packets are buffered and forwarded to new foreign agent (FA) during handoff, packet losses are still present due to the weak signal strength and handoff latency is still long upto few seconds because of the latency of link layer handoff. Thus, we need to overcome possible shortage of buffer by adopting pre-buffering techniques. However, inaccurate and conservative choice on the required buffering margin can waste limited latency budget. A large buffer leads playback delay.

In this paper, we introduce a seamless streaming framework by estimating the accurate pre-buffer size to compensate the handoff latency and by adapting rate-shaping scheme to overcome the shortage of bandwidth immediately following the handoff. We calculate the handoff latency by extending our previous work [10]. The handoff latency depends on the link delay between FAs and HA, queue state of FAs, streaming traffic flow, and link layer handoff latency. We implement a preliminary version of the proposed framework by extending the basic mobile IPv4 system. Network simulations as well as media streaming experiments over mobile IP-enabled Wireless LAN show that the proposed scheme can provide appropriate guidelines on buffer parameters.

2 Seamless Media Streaming Framework

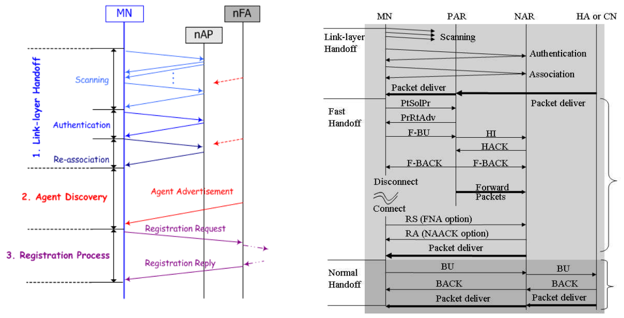
2.1 Problem statement

Mobile IPv4: In the mobile IPv4-enabled WLAN environment, when a mobile node (MN) moves to another foreign network, it experiences handoff latency due to following handoff processes[1]. First, the mobile node can communicate with only one AP at each time and it thus cannot communicate with an old foreign agent (FA) during link-layer (L2) handoff. This L2 handoff includes probe, authentication, and reassociation delays. In addition, the registration process of IP-layer (L3) handoff can begin only after the L2 handoff. This L3 handoff includes agent discovery time, registration message propagation, and message processing time. As stated above, total handoff latency is presented by time line for packet flows in Fig. 1 (a). Clearly, a streaming client starves for media data during the handoff period. Thus, playback of the streaming client can be interrupted.

When a MN is visiting in a foreign network, its HA can intercept packets as proxy that are addressed to a home address of the mobile node. It then sends those packets through IPinIP tunneling to the care-of address (CoA) that indicates the termination point of a tunnel toward the mobile node side. As the MN performs the link-layer/IP-layer handoffs to move to another foreign network, it cannot receive any packets toward the MN, since these packets are destined to old CoA and the mobile node cannot communicate with the old AP and the old FA during the handoff period. This leads to busty packet losses during handoffs effecting the quality of the steaming media.

Mobile IPv6 fast handoff in 802.11 networks: The fast handoff protocol of Mobile IPv6 supports handoff without packet losses by forwarding packets to new access router in advance [5]. In the Fig. 1 (b), the message flow for the fast handoff protocol is described. However, the fast handoff protocol of Mobile IPv6 has following problems in the 802.11 WLAN. First, the fast handoff protocol of Mobile IPv6 still suffers packet losses in the 802.11 WLAN networks. When

[1] The handoff latency varies upto 2.274s in our experiment.



(a) Mobile IPv4 (b) Fast Handoff of mobile IPv6

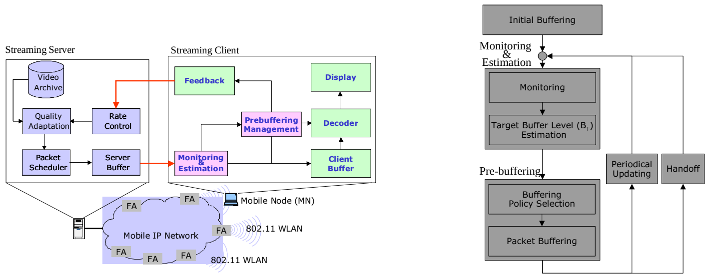
Fig. 1. Handoff latency of (a) mobile IPv4 and (b) Fast handoff of mobile IPv6.

a MN changes its communication channel during the scan period, it cannot communicate with its PAR. Since the PAR does not know MN's scan, it tries to transmit the packets to the MN and they will be dropped. Second, the MN can executes the scan sometime prior to the handoff. It enables to send the F-BU without a new channel scan during handoff procedure. However, this pre-scan may be performed far in advance of the handoff and the pre-selected AR may become incorrect because of changed channel condition. Also, during the scan the MN cannot receive the packets from its AR. Finally, in many commercial 802.11 WLAN devices, the channel scan and AP association cannot be separated and MN may associate with the new AP after the scan. Thus, to come back to the previous AP connected to the previous AR, the old association should be maintained.

2.2 Proposed Framework

Fig. 2 (a) shows the seamless media streaming framework for mobile IP networks [3], where IEEE 802.11 devices are configured in a wireless LAN infrastructure. A streaming application on a MN receives packets from a media server, while maintaining client buffer to overcome fluctuations in network resources such as available bandwidth, delay, and loss. The streaming server reacts to the feedback from the streaming client and performs quality adaptation, and packet scheduling. The streaming client sends its status information to the server, which includes parameters such as current buffer occupancy, receiving rate, and error rate. The relationship between feedback and reaction has been studied in [7].

In our framework, the handoff transient time is estimated before the handoff occurs. To estimate the handoff transient time, the mobile client monitors network conditions such as link delay, flow rate, and queue status of the neighbor FAs. The handoff protocol and related signaling procedure are analyzed to get the handoff transient time. After estimating the handoff transient time, the streaming client tries to prepare enough data to compensate for the estimated interruption time during the handoff. To acquire the estimated buffer level, tar-



(a) Seamless media streaming framework. (b) Pre-buffering procedure

Fig. 2. Seamless media streaming framework in mobile IP networks.

get buffer level, the streaming client tries to collect packets while playing back the received media packets. There are two choices to boost required target buffer level: increasing sending rate at the streaming server or decreasing the playback speed. The choice depends on the policy of pre-buffering management module.

The target buffer level can be varied with time. Every FA will have different required buffer level for handling handoff transient time. Even in the same FA, network traffic condition can change the target buffer level. Thus, it is needed to keep the exact buffer level in order not to stop playback during the handoff. In Fig. 2(b), the estimation and buffer-level adjustments are performed periodically. To summarize, the pre-buffering scenario consists of buffer-level estimation procedure and buffer-level adjustment procedure. In the following section, we will analyze the handoff procedure to get handoff transient time which will be used to estimate target buffer level for pre-buffering.

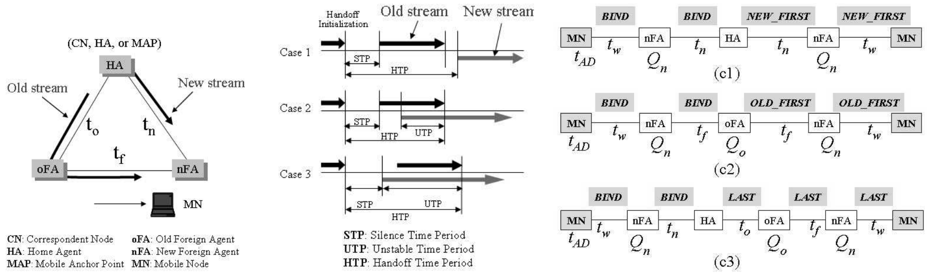
3 Target buffer level estimation for seamless handoff

Handoff latency estimation: In this section, handoff latency is estimated, which is based on the results of previous work [10]. Under the mobile IPv4 networks, a MN has a time period when it can not send and receive packets during handoff. We define this blackout period as the STP (silence time period). Also, we define the UTP (unstable time period) during when the packet sequence could be mis-ordered. Finally, HTP (handoff time period) is total handoff period.

Depending on the location of oFA and nFA, the timing between the old and new streams defined in Fig. 3 (a) can be classified into three cases. As shown in Fig. 3 (b), we can associate the transient time periods (i.e., STP, UDP, and HTP) according to link delays in nFA, oFA, and HA. For example, the 'Case 1' illustrates the situation where a MN moves to the far-away (in network routing sense) nFA from its HA.

Thus, by setting the time when the MN leaves the oFA to zero, the STP and UTP can be denoted by

$$STP = min(T_{f\_old},\ T_{f\_new}), \tag{1}$$



(a) (b) (c)

Fig. 3. Possible packet orderings observed at the MN during the handoff: (a) traffic flows during a handoff, (b) possible packet ordering at the MN, and (c) message flows and associated link delays ((c1) the first packet of the new stream (for T_{f_new}), (b) the first packet of the old stream (for T_{f_old}), and (c) the last packet of the old stream (for T_{l_old})).

$$UTP = max(0, T_{l\_old} - T_{f\_new}), \tag{2}$$

where T_{f_new} is the time when the first packet of new stream arrives to the MN and T_{f_old} and $T_{l_old}$is the time when the first packet and the last packet of the old stream is delivered to the MN, respectively. The link delays between HA - oFA, HA - nFA, and oFA - nFA are denoted by t_o, t_n, and t_f, respectively. A handoff is started when a MN moves from the oFA. After the departure, there is a delay until the moved MN receives a *Router Advertisement* (RA) message from a nFA. This delay consists of handoff latency in a link layer and an agent discovery delay. We denote the L2 handoff delay and the agent discovery delay as t_{L2} and t_d, respectively. The propagation delay of wireless link is denoted by t_w. Queueing delay of a packet in nFA, oFA, and HA is denoted by Q_o, Q_n, and Q_h, respectively. The MN can recognize the change of attachment after L2-handoff completion and RA packet reception. This attachment delay is given by $t_{AD} = t_{L2} + t_d$.

L2 handoff delay consists of AP scan, authentication, and association procedures. AP scan is performed to get available AP list. Scan is a process in which the MN cycles through all the channels and sends a “Probe” to all APs within its range and waits for a “Beacon” from these APs within a time period. Thus, the scan time depends on the ”Beacon” waiting time and the number of channels. Scan time mainly dominates the L2 handoff delay.

According to the delay of each message flows shown in Fig. 3 (c), the T_{f_new}, T_{f_old},T_{l_old} can be described as follow:

$$T_{f\_new} = t_{AD} + 2t_w + Q_n + Q_c + 2t_n, \tag{3}$$

$$T_{f\_old} = t_{AD} + 2t_w + 2t_f + Q_o + Q_n, \tag{4}$$

$$T_{l\_old} = t_{AD} + 2t_w + 2Q_n + t_n + t_o + Q_o + t_f. \tag{5}$$

Packet loss estimation: Handoff is initiated by monitoring the received signal

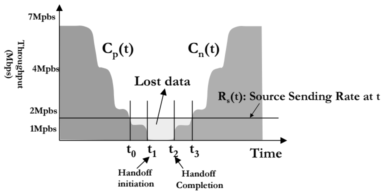


Fig. 4. Handoff scenario based on channel throughput.

strength of received data. Fig. 4 describes a throughput profile during a handoff. When a MN is moving to a nFA from an oFA, the signal strength (and related transmission throughput) from the oFA decreases [4]. It is clear that the transmission throughput of IEEE 802.11b is smallest when the MN is around handoff. The streaming client should consider the throughput degradation around a handoff as wall as the handoff processing delay described in previous section. Streaming application may not guarantee required bandwidth (so called the source sending rate at time t, $R_s(t)$), around a handoff. Accordingly, when the throughput is less than $R_s(t)$, the packets sent by a streaming server will be lost. Suppose that the throughput of the oFA channel becomes lesser than $R_s(t_0)$ at t_0 and the MN starts a handoff at t_1 as shown in Fig. 4. When the handoff is finished at t_2, the channel throughput of the nFA is lesser than $R_s(t_2)$ until t_3. The total packet loss caused by handoff can be divided into three losses: pre-loss, post-loss, and handoff-loss. The pre-loss, L_{pre}, and the post-loss, L_{post}, are packet losses during periods before and after the handoff, respectively. The handoff-loss, $L_{handoff}$, is data losses during handoff duration. Then, the total loss, L_{total}, can be presented by $L_{total} = L_{pre} + L_{handoff} + L_{post}$, where $L_{pre} = \int_{t_0}^{t_1}(R_s(t) - C_p(t))dt$, $L_{post} = \int_{t_2}^{t_3}(R_s(t) - C_n(t))dt$, and $L_{handoff} = \int_{t \in STP} R_s(t)dt$, where $C_p(t)$ and $C_n(t)$ is the throughput function of channel in the oFA and in the nFA at time t, respectively.

Channel adaptation by feedback: The packet losses caused by decreased signal strength can be reduced by adopting a rate shaping method [7]. Using the feedback sent by the client, the streaming server can adjust $R_s(t)$ to the rate constrained by channel condition. For example, the streaming server reduces $R_s(t)$ to $C_p(t_0)$ at t_0 in Fig. 4. Generally, after a handoff, the received signal strength from the nFA is bigger than that from the oFA. Thus, $C_p(t_1) <= C_n(t_2)$. Then, there is no pre-loss and post-loss, since the $R_s(t_0) = C_p(t_0)$ and $C_p(t_0) = C_p(t_1) < C_n(t_2)$. Consequently, the total loss includes only the handoff-loss.

Smooth handoff - no packet losses: In the smooth handoff, during STP, the packets in the old stream are stored in oFA and forwarded to nFA. Hence, the handoff-loss does not exist in the smooth handoff, if FAs have enough buffer space to save the packets in the old stream during the STP.

Minimum buffer size to overcome handoff latency: We now describe the minimum buffer size that the client can tolerate the handoff latency. The video and audio data are first packetized at the server. For simplicity we assume that the media data is compressed such that one video frame and associated audio fit into one service frame. The service frame is segmented into n packets and the packets are transmitted at a constant rate. The packets are stored in the receiver buffer until the buffer reaches the target buffer size B_{Target}. When media playback is started, n packets are collected to reconstruct one service frame. Playback time of each service frame corresponds to inter-frame spacing, t_F, at the sender. Buffered packets in a streaming client, k, can be played out during T_p and it can be denoted by $T_p(k) = \lfloor \frac{k}{n} \rfloor \cdot t_F$, where $\lfloor x \rfloor$ represents the largest integer smaller or equal to x. Once the receiver faces buffer underflow, it stops playing media and stores incoming packet to the receiver buffer until the receiver buffer size reaches B_{Target}. When we apply adaptive playout control to streaming client, T_p can be more increased. The main role of the adaptive playout control is to reduce the discontinuity incurred by buffer over/underflows. For adaptive playout we need to increase or decrease the playout speed of frames. The playout factor, s, is the addition speed ratio: if $s > 1$, the playout speed is slowdown and otherwise, the playout speed becomes fast. The maximum playout time with buffered packets increases or decreases to $T_p^{'} = sT_p$. Without the smooth handoff, packets are lost during handoff. Streaming client sends a retransmission request message to recover packet loss. Obviously, to recover one packet loss, streaming client takes one round trip time (RTT). Thus, to receive lost packets generated during burst loss period, t_{loss}, streaming client should wait $T_{rtx}(t_{loss}) = t_{loss} + RTT$. In our seamless framework described in Sec. 2.2, packet losses caused by decreased signal strength around a handoff can be overcome by rate shaping controlled by client's feedback information. Then, to playback media continuously, streaming application has to keep buffered packets, B_{Target}, and $T_p(B_{Target}) \geq T_{rtx}(STP)$. If we add the condition to above conditions, the target buffer level is given by

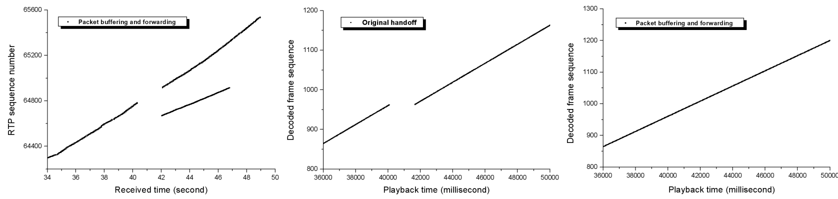
$$\lfloor \frac{B_{Target}}{n} \rfloor = \begin{cases} \frac{T_{rtx}(STP)}{s \cdot t_F}, & \text{without smooth handoff} \\ \frac{STP}{s \cdot t_F}, & \text{with smooth handoff.} \end{cases} \tag{6}$$

4 Experiment and Simulation

In experiment, the received RTP sequence and decoded video frame sequence are monitored to see the seamless handoff. In simulation, the accuracy of prebuffering level estimation is verified by buffer consumption rate during a handoff.

4.1 Experiment results

The HUT version of mobile IP [8] and the MPEG4IP implementations [9] for Linux are adopted for our experiment. On top of the HUT mobile IP, we add the smooth handoff, since the HUT version of mobile IP does not have the smooth handoff. Cisco Aironet 350 series APs and client adapters supporting IEEE 802.11b are configured.



(a) (b) (c)

Fig. 5. (a) Received RTP sequence with smooth handoff, (b) decoded frame sequence with smooth handoff, and (c) decoded frame sequence with proposed framework.

We set the period of agent advertisement broadcasting to $1sec$, which is the lower boundary of the sending interval of agent advertisement messages in mobile IP specification. Handoff latency is computed as a time interval between the last packet transferred from the old FA and the first packet transferred from the new FA (except for the mobile IP signaling). To measure required link delays both of wireless link delay (t_w) and link delay between old FA and new FA (t_f), we added the probing packet and measurement functions in the HUT mobile node and agent. In our testbed, we measured each delays several times and take the maximum values among them: $t_f = 1.84ms$, $t_w = 2.09ms$, $t_{L2} = 1.26sec$, $t_d = 1.007sec$, $t_{AD} = 2.267sec$. According to Eq.(1), we set STP to $2.274sec$[2]. Our experiment does not fully implement the proposed framework shown in Fig. 2 and it is a preliminary work. In our experiment, we simply use a fixed pre-buffering time estimated by maximum measurement result of STP. To evaluate the quality improvement for streaming media, we experiment MPEG-4 video streaming with the sending rate of about 1 Mbps and the frame rate is $25frames/sec$. We measure the sequence number of RTP packet and the decoded frame sequence.

The trace of sequence number of received RTP packets is depicted in Fig. 5 (a). We can observe that the packets from the old stream can be received by the client by the smooth handoff. The trace of decoded frame is depicted in Fig. 5 (b) and (c). From the Fig. 5 (b), we observe that if the pre-buffering time was set at under $2sec$, which is default value of the MPEG4IP streaming client, the playback was interrupted, even thought the smooth handoff is used. However, in the Fig. 5 (c), there does not exist playback disruption during the handoff period because the forwarded packets can be applied to playback by the sufficient pre-buffering in the MPEG-4 client.

4.2 Simulation Results and Discussion

The simulation was based on the Network Simulator (NS-2). Fig. 6 shows the simulation scenario and parameters. Under the rate shaping option, the STP is

[2] In the mobile IPv4, STP can be greatly changed by t_d, which is a random variable from 0 to $1sec$.

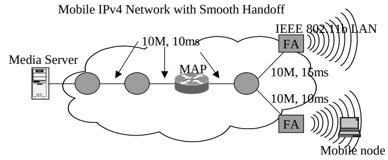


Fig. 6. Simulation scenario ($R(s)_t$=2.4Mbps, n=10packets, packet size=1Kbyte, and t_F=0.03sec).

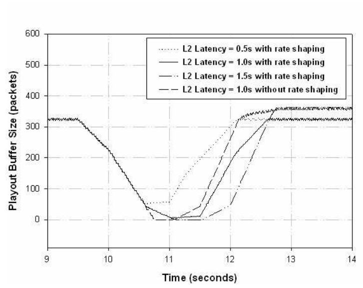
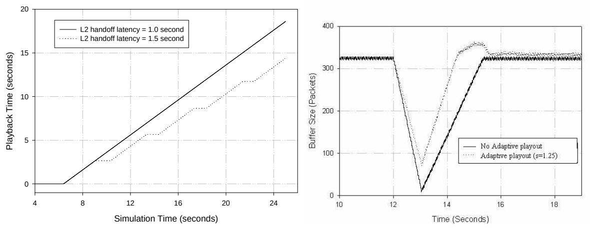


Fig. 7. Buffer consumption rate with respect to L2 latency; in each case B_{Target} is fixed by 330 packets that is for $t_{L2} = 1.0sec$.

main variable to calculate the B_{Target}. And, L2 handoff latency (t_{L2}) is a main control variable to calculate the B_{Target}. The IEEE 802.11b PHY is operated with 11Mbps in normal condition. It changes the mode to 1Mbps in the oFA before 0.5sec of handoff initiation. After the end of handoff, it continues 1Mbps data rate and the mode is changed to 11Mbps after 0.5sec later. In our simulation, the 10% additional margin is added to the B_{Target}. Only the streaming traffic is emitted to the link, and other traffic are not introduced in the simulation, which makes queueing delay at FAs ignorable.

Fig. 7 represents the buffer consumption level with respect to the L2 handoff latency variation. The consumed buffer level is increases as the t_{L2} increases. When t_{L2} is 1.5sec, all 330 packets are consumed and the client suffers from buffer underflow. When t_{L2} is 0.5sec, only 268 packets are consumed . When the t_{L2} is 1.0s, 326 packets, are consumed but there is no buffer underflow. The results are meet the result of the Eq. (6) well. When t_{L2} is 1.0sec without rate shaping, we can observe that the buffer becomes underflow quickly. Note that the channel bandwidth reduces to 1Mbps before handoff but streaming traffic is still 2.4Mbps. Thus, the oFA drops or stores the incoming packets in the buffer, which results in the client buffer level lower before handoff. The media playback time of a streaming client is shown in Fig. 8(a) where the target buffer level is 330 packets and maximum buffer limit is 363 packets that is estimation result for 1sec of L2 handoff latency. When the L2 handoff latency is 1sec, the playback is not interrupted but continuously served. However, with the same target buffer level (363 packets) but the L2 handoff latency is 1.5sec, playback



(a) (b)

Fig. 8. (a) Playback time with respect to the L2 handoff latency where the B_H is 330 packets (33 frames) and (b) Buffer consumption of adaptive playout control (s=1.25).

is interrupted whenever a handoff happens. We can also see that the adaptive playback can reduce the required target buffer level as shown in Fig. 8(b). The normal playback keeps 8 packets in the buffer during handoff, but the adaptive playback has 71 packets in Fig. 8(b). This means that the target buffer level of the adaptive control can be reduced as a factor of adaptive playout factor, s.

5 Conclusion

We introduced the seamless streaming framework by estimating the accurate buffer level for pre-buffering to compensate the handoff latency. We calculated the handoff latency in application point of view under the mobile IPv4 environment. The simulation and experiment results show that the handoff-aware streaming has no playback discontinuity while keeps a minimal pre-buffer size.

References

1. E. Steinbach, N. Farber, and B. Girod, "Adaptive playout for low latency video streaming," in *Proc. of IEEE ICIP'01*, 2001.
2. H. Yokota, A. Idoue, T. Hasegawa, and T. Kato, "Link layer assisted mobile IP fast handoff method over wireless LAN networks," in *Proc. of ACM MobiCom '02*, Atlanta, Georgia, Sep. 2002.
3. C. Perkins, "IP Mobility Support in IPv4,", IETF RFC 3344, Aug. 2002.
4. Javier D. Pavon and Sunghyun Choi, "Link adaptation strategy ofr IEEE 802.11 WLAN via received signal strength measurement ," in *Proc. of IEEE ICC'03*, 2003.
5. R. Koodli, "Fast handovers for mobile IPv6," IETF RFC 0468, July 2005.
6. K. Malki, "Low Latency Handoffs in Mobile IPv4," IETF Internet Draft, July 2005.
7. G. J. Conklin, et. al, "Video coding for streaming media delivery on the Internet," in *IEEE Trans. on Circuits and Systems for Video Tech.*, vol.11, no.3,2001.
8. (2001) Dynamics - HUT Mobile IP, [Online] Available: http://dynamics.sourceforge.net
9. Bill May. (2003) MPEG4IP - Open Streaming Video and Audio. [Online]. Available: http://www.mpeg4ip.net
10. D. Lee and J. Kim, "Transient time period analysis of smooth handoffs in mobile IP networks and its application to media streaming," in *Proc. of SPIE Applications of Digital Image Processing XXVI*, Aug. 2003.