Vertical Handover Strategy for Multi-Layered Real-Time Video Traffics

Hyun-Ho CHOI†, Hyunggon PARK††, Nonmembers, and Jung-Ryun LEE†††(a), Member

SUMMARY In this letter, we present a new method of alleviating the deterioration in the quality of real-time video service during vertical handover (VHO). The proposed method stochastically delays the starting time of the service disruption of VHO in order to reduce the number of lost frames caused by the inter-frame dependency of multi-layered video traffic. The results show that the proposed method significantly decreases the average frame loss time at the sacrifice of an increased handover execution time by one half of the group of picture (GOP) interval of the video traffic.

key words: vertical handover, multi-layered traffic, inter-frame dependency

1. Introduction

One of the essential technologies to provide seamless service in heterogeneous networks is the vertical handover (VHO). In the VHO, a mobile device has to change its physical air interface (e.g., from 3GPP UMTS to WLAN) and the frame transmission/reception is interrupted during that time. This time duration is called Service Disruption Period (SDP). In general, the SDP of the VHO is longer than that of the horizontal handover (HHO) because of the additional procedures of the VHO, such as the authentication and registration procedures. Therefore, a VHO user experiences a more degraded quality of service (QoS) compared to an HHO user, especially for delay-constraint services such as real-time video service rather than non-real-time web-browsing [1].

In real-time video services that support multi-layer encoding, such as the H.264 codec, the frames can be encoded in a sequence called the Group of Pictures (GOP) that consists of I-, P-, and B-frames, thereby reducing the spatial and temporal redundancies of the video frames [4]. These frames have hierarchical dependencies that contribute to different distortion reduction. More specifically, the I-frame is encoded by itself without any motion compensation, the P-frame is encoded in reference to the preceding I- or P-frame, and the B-frame is encoded with reference to both the preceding and succeeding I- or P-frames. Therefore, the loss of an I-frame results in the corruption of all of the following frames until the next I-frame occurs, and the loss of a P-frame causes the loss of all of the frames that are referenced by that P-frame. On the other hand, the loss of a B-frame does not corrupt any other frames.

Efficient algorithms to overcome the QoS deterioration of real-time video services during the VHO have been studied. In [2], a cross-layer TCP-friendly rate control (TFRC) scheme and the Context Transfer Protocol to minimize the handover signaling procedures are discussed. Another method [3] dynamically controls the GOP structure of real-time video traffic by adjusting the frame interval, so as to provide a seamless real-time video service during the VHO. However, these algorithms in [2] and [3] require the real-time information about network status in the encoders. Since there is no mechanism for the encoder to be aware of network status, the data encoded in the encoder should be transcoded before the transmission, yielding the increase of complexity of the encoder. In order to overcome these limitations, we propose a new approach to alleviating the QoS deterioration of real-time video services during VHO by adjusting the start of SDP in VHO in an opportunistic way. This can alleviate the effect of SDP on the frame loss of video traffic using the inter-frame dependency of multi-layered video traffic without any modification of the GOP structure of video traffic.

2. Modeling of Service Disruption Period

The SDP is determined by the sum of the transmission times of several signaling messages for VHO. These messages are delivered via wireless links and multiple hops in a wired network. For the modeling of the SDP during VHO, the handover delay model proposed in [5] is deployed, which shows the typical signaling flow for vertical handover in the UMTS-WLAN interworking environment (see Fig. 1). In [5], the handover delay consists of five components: the latency of the wireless links ($D_{wl}$), the latency of wired links ($D_w$), the delay at the wireless interfaces ($L_{wl}$), the delay at the wired interfaces ($L_w$) and an additional application latency due to the lookup process in each router node ($L$). $L_{wl}$ and $L_w$ consist of fixed delays such as the processing and propagation delays and are denoted as $d_{wl}$ and $d_w$, respectively. $D_{wl}$ and $D_w$ consist of variable transmission and queuing delays in the wireless and wired interfaces, respectively. $D_{wl}$ is expressed as the sum of the exponential random variable $X$, with rate $\mu C_l - \lambda_{wl}$ where $1/\mu$, $C_l$, and $\lambda_{wl}$ are the mean message length, the capacity of commu-

---

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers
The errors in a frame may propagate over the other related frames in a GOP, resulting in performance degradation.

The frame loss time is defined as time interval during when the user cannot receive any frames or receives corrupted frames. Therefore, the frame loss time of multi-layer video traffic can be derived from both the frames lost directly by the SDP and the frames lost indirectly by being referred by the directly lost frames. Figure 2 shows that total frame loss time is not fixed but varies even for the same length of the SDP, since the number of indirectly lost frames caused by the inter-frame dependency changes depending on the position of the SDP. In detail, if an I-frame of a GOP is lost, then all of the following P- and B-frames in the GOP are lost. The loss of a P-frame causes the loss of all of the following B- and P-frames, so the loss of a preceding P-frame causes more indirectly lost frames than that of a succeeding P-frame. In this context, the frame loss time is minimized when the SDP ends right before the next I-frame starts, so that the SDP is more likely not to include the I-frame and preceding P-frames.

Considering the above-mentioned inter-frame dependency in view of minimizing the impact of SDP on frame loss time, the best strategy to improve the user-perceived QoS may be to defer the starting time of the SDP so that the SDP ends right before the next I-frame starts. Notice that although it is practically impossible to control the occurrence time of VHO, it is possible for network entities to control the starting time of SDP by recessing the transmission of particular VHO message for a while. In the UMTS-WLAN interworking procedure shown in Fig. 1, the BS can handle the starting time of SDP by delaying the transmission time of ACK1 message.

Let \( t_s \) be the starting time of the SDP and let \( I_p \) be the first forthcoming I-frame after \( t_s \). Then, the loss probability of forthcoming I-frame (i.e., the probability that the current SDP includes \( I_p \)), \( p_c \), is calculated as

\[
p_c = \text{Prob}[Z \geq t_{I_p} - t_s] = 1 - F_Z(t_{I_p} - t_s)
\]

where \( t_{I_p} \) denotes the \( I_p \) frame’s occurrence time.

On the other hand, we define the target probability \( p_{tar}(k) \) as

\[
p_{tar}(k) = \text{Prob}[Z \geq T(k)] = 1 - F_Z(T(k))
\]

where \( T(k) \) is a function of \( k \). Here, \( T(k) \) can be arbitrary
Proof 1: Figure 3 shows the schematic representation of the proposed algorithm. Suppose that the SDP starts at time $t_{\text{tar}}$ and $k$ is a non-decreasing function of $k$, whose role is to control the value of $p_{\text{tar}}(k)$.

The target of the proposed algorithm is to make $p_c$ be the same as $p_{\text{tar}}(k)$. That is, the proposed algorithm controls the starting time of SDP in order that the estimated loss probability of I-frame equals the target probability, by adding some delay at between handover occurrence time and handover execution time.

Let $T_w$ be the amount of the delay of SDP starting time. The following Proposition 1 determines the value of $T_w$ according to the value of $p_c$. 

**Proposition 1:** The waiting time in the proposed algorithm to make the probability of forthcoming I-frame loss be the same as the target probability is given by $T_w = t_{\text{tar}} - t_s - T(k)$ or $T_w = t_{\text{tar}} - t_s - T(k)$. 

**Proof 1:** Figure 3 shows the schematic representation of the proposed algorithm. Suppose that the SDP starts at time $t_s$ and $k$ is given. Let $t_e(k)$ be the estimated ending time of SDP with probability of $1 - p_{\text{tar}}(k)$. From (3), $t_e(k) = t_s + T(k)$. Let $t'_e$ and $t'_e(k)$ be the delayed times of $t_e$ and $t_e(k)$ by the amount of $T_w$, respectively. That is, $t'_e = t_e + T_w$ and $t'_e(k) = t_e(k) + T_w$.

**Case 1** If $p_c \leq p_{\text{tar}}(k)$, then $t_e(k) \leq t_e$ from (2) and (3). In order for the delayed ending time of SDP to end at time $t_{\text{tar}}$, we have following equation:

$$t'_e(k) = t_e(k) + T_w = t_s + T(k) + T_w = t_{\text{tar}}. \quad (4)$$

**Case 2** If $p_c > p_{\text{tar}}(k)$, then $t_e(k) > t_e$. So the estimated ending time of SDP with the target probability of $p_{\text{tar}}(k)$ is larger than the occurrence time of forthcoming I-frame. In this case, the delayed ending time of SDP should end at the occurrence time of second forthcoming I-frame, $t_{\text{tar}}$, in order to make the probability of I-frame loss be the target probability. So, we have following equation:

$$t'_e(k) = t_e(k) + T_w = t_s + T(k) + T_w = t_{\text{tar}}, \quad (5)$$

(4) and (5) end the proof trivially. 

### 4. Experimental Results

The used parameters for evaluations are based on [5]. We set $\alpha_e=2 \text{ ms}$, $\alpha_{sd}=0.5 \text{ ms}$, and $\Delta=100 \text{ ms}$. $\lambda_w$ and $\lambda_{sd}$ are set randomly between 90 and 100 $\text{s}^{-1}$ and $C_i$ is set between 2 and 2.5 Mbps. From Fig. 1, $N_1$ and $N_2$ are set to 3 and 7, respectively. Regarding the video codec, we refer to the H.264/AVC codec whose encoding rate is 25 fps and the GOP patterns are $I(BBP)^3BB$ and $I(BBBP)^3BBB$. Regarding the target probability, we set $T(k) = E[Z] + k\sigma_Z$ by using the mean ($E[Z]$) and standard deviation ($\sigma_Z$) of $Z$ as in [8]. We use the Monte Carlo method with 10,000 VHO calls for the simulation runs. The performance of the proposed method is compared with that of the conventional scheme, in which no SDP starting time control is employed, i.e., $T_w=0$.

In this letter, we assume that the network is non-congested and enough link bandwidth is provided for the service, in order to make a frame be lost only due to the SDP in the VHO procedure, and therefore to focus on the effect of the proposed method on the performance of the proposed VHO procedure.

Figures 4 and 5 show the average frame loss time and average waiting time of the mobile station (MS) under both the conventional and proposed methods as a function of $k$, where GOP patterns are $I(BBP)^3BB$ and $I(BBBP)^3BBB$, respectively. As $k$ increases from 0 to 5, $p_{\text{tar}}(k)$ decreases from 0.41 to 0.002. Therefore, the probability that the SDP includes a forthcoming I-frame decreases and consequently, the average frame loss time decreases. However, at the instant when the increasing $T(k)$ includes a P-frame, all of the B-frames on either side of the P-frame are lost. This explains the sudden increase of the frame loss time in Fig. 4. However, when GOP pattern is $I(BBBP)^3BBB$, $T(k)$ cannot include a P-frame until $k$ increases up to 5 because of longer distance between I-frame and P-frame.
Fig. 4 Average frame loss time and average waiting time as a function of $k$ when GOP pattern is $(BBP)^3BB$.

Note that, in Fig. 4, the average frame loss time is minimized when $T(k)$ includes only P-frames while $p_{in}(k)$ decreases as much as possible, which means that $T(k)$ is located between $P_3$ and $I_n$ (or $I_{n+1}$) in Fig. 3. In this result, the average frame loss time is minimized when $k$ is 1.62, which is obtained by solving $E[Z] + k\sigma_Z = 150$ ms where $E[Z]=135.79$ ms and $\sigma_Z=8.73$ ms from the simulation parameters and 150 ms is the time gap between $P_3$ and $I_n$ (or $I_{n+1}$).

In Figs. 4 and 5, the average waiting time increases by an average of 0.32 s and 0.24 s, respectively. These values correspond to one half of a GOP interval. Note that the average waiting time is almost fixed, irrespective of the value of $k$, since the occurrence of VHOs is independent of the value of $k$ and is uniformly distributed over time.

5. Conclusion

The experimental results verify that the proposed method considerably reduces the frame losses while increasing the total handover execution time by approximately a half of a GOP interval. Since the performance of the proposed method is affected by what kinds of inter-frame dependency and vertical handover delay models are employed, we will extend our work by considering various video codecs and VHO delay models in future.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education Science and Technology (2011-0025424 and 2011-0024132), the Human Resources Development program (No.20124030200060) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Trade Industry and Energy, and the MSIP (Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1015) supervised by the NIPA (National ICT Industry Promotion Agency).

References

[1] A.K. Salkintzis, G. Dimitriadis, D. Skyrianoglou, N. Passas, and N. Pavlidou, “Seamless continuity of real-time video across UMTS and WLAN networks: Challenges and performance evaluation,” IEEE Wirel. Commun., vol.12, no.3, pp.8–18, June 2005.
[2] J.Y. Pyun, “Context-aware streaming video system for vertical handover over wireless overlay network,” IEEE Trans. Consumer Electronics, vol.54, no.1, pp.71–79, Feb. 2008.
[3] H.S. Kim, J.W. Yun, and H.C. Jin, “Dynamic GOP structure selection to support QoS in vertical handoff,” Proc. IEEE ISCE Symp., pp.237–241, June 2005.
[4] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” IEEE Trans. Circuits Syst. Video Technol., vol.17, no.9, pp.1103–1120, Sept. 2007.
[5] K.S. Munasinghe and A. Jamaliour, “An analytical evaluation of mobility management in integrated WLAN-UMTS networks,” Computers & Electrical Engineering, vol.36, no.4, pp.735–751, July 2010.
[6] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol.13, no.7, pp.560–576, July 2003.
[7] G.J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol.22, no.12, pp.1649–1668, Dec. 2012.
[8] J.R. Lee and D.H. Cho, “Performance evaluation of energy-saving mechanism based on probabilistic sleep interval decision algorithms in IEEE 802.16e,” IEEE Trans. Veh. Technol., vol.56, no.4, pp.1773–1780, July 2007.
[9] M. van der Schaar and P. Chou (eds.), Multimedia over IP and Wireless Networks: Compression, Networking, and Systems, Academic Press, 2007.