Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Coping with handover effects in video streaming over cellular networks

BOUAZIZI Imed, HANNUKSELA Miska M., RAUF Usama
(*Nokia Research Center, Tampere 33720, Finland*)
E-mail: {imed.bouazizi; miska.hannuksela; usama.rauf}@nokia.com

Received Dec. 4, 2005; revision accepted Mar. 3, 2006

Abstract: The 3rd generation partnership project (3GPP) has defined the protocols and codecs for implementing media streaming services over packet-switched 3G mobile networks. The specification is based on IETF RFCs on audio/video transport. It also adds new features to achieve better adaptation to the mobile network environment. In this paper, we propose an algorithm for handover detection and fast buffer refill that is based on the existing feedback and signaling mechanisms. The proposed algorithm refills the receiver buffer at a faster pace during a limited time frame after a hard handover is detected in order to achieve higher video quality.

Key words: Video streaming, 3GPP packet-switched streaming service, Handover detection, Hard handover, Fast buffer refill, Mobile networks
doi:10.1631/jzus.2006.AS0137 **Document code:** A **CLC number:** TN919.8

VIDEO STREAMING OVER CELLULAR NETWORKS

Video streaming over mobile networks is attracting a steadily growing user community and is more and more being perceived as a substantial service for mobile entertainment. However, the development of this trend is distorted by technical difficulties such as low channel bitrates and service disruption caused by cell handovers and poor network coverage. The former technical difficulty can be circumvented by deploying highly efficient compression algorithms such as H.264/AVC video coding that achieves reasonable video quality at relatively low bitrates. Connection interruptions due to poor network coverage can be detected and fixed by correct network planning measures performed by the mobile network operator. However, cell handovers are more difficult to address. In this paper, we study the effects of handovers on video streaming sessions and propose an algorithm to cope with handovers while achieving fairly constant video quality.

During the lifetime of a streaming session, the mobile device may need to perform several handover operations. Fig.1 depicts a scenario where a user is moving between two cells, which results in a handover. A handover can be initiated for several reasons, such as low signal quality in the active cell in uplink or downlink channel, detection of a better signal from a neighboring cell, inability to meet Quality of Service requirements anymore by the current cell, or user mobility.

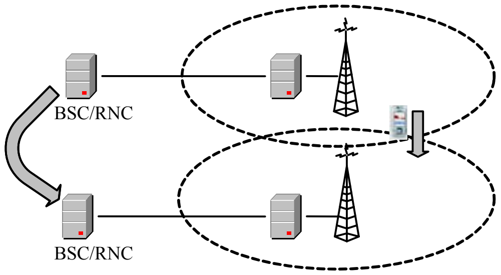


Fig.1 Example of an inter-system handover (between radio network controllers/base station controllers) initiated by mobility

A handover can be either seamless (soft or softer) or hard. Seamless handovers are transparent to the application. In other words, the connection interruption and the resulting packet delays and packet losses are insignificant to the application. Hard handovers usually take place when there is need to switch between frequencies and/or systems (e.g. when moving from 3G to 2G networks or vice versa). During a hard handover, the previous link layer connection is torn down and a new link layer connection is established in the target cell, although the application layer connection is maintained. This usually will result in a significantly long connection interruption, which in turn will incur increased transmission delays and packet loss. The handover may last for several seconds, during which no data is delivered to the receiver. The handover problem has been studied thoroughly at the data link and network layers (Banerjee *et al*., 2003; Wisely *et al*., 2002). However, little work has studied the effects of handover at the application layer.

Coping with hard handover effects at the application layer has been addressed by (Kampmann and Baldo, 2004; Schierl *et al*., 2005; Schierl and Wiegand, 2004). The proposed solutions use an estimation of the receiver buffer duration and a threshold value to detect handovers. They then use a priority-based scheduling algorithm to achieve pre-configured levels for the different significance classes (independent decoding refresh, reference, and non-reference pictures) at the receiver buffer. Pictures are sent in significance class order until the pre-configured buffer occupancy level of that significance class is reached, after which the next significance class is considered.

Finally, if the buffer level is too low, stream switching is performed. The drawbacks of this approach are:

(1) The handover detection is delayed compared to the method in this paper, as the buffer occupancy for the different significance classes decreases relatively slowly as the consequence of a handover. This is especially the case in the absence of negative acknowledgement messages, possibly due to packet loss induced by the handover.

(2) The priority-based scheduling algorithm may lead to a prolonged period of slide show presentation. This is mainly due to the usage of the interleaved transmission. During the time when a high significance class buffer level is refilled, several lower class media units may have passed their display time and cannot be transmitted anymore.

(3) When a long enough buffering delay is used as is typical in video streaming sessions, the server usually has enough time to retransmit some of the important lost media units. The priority-based scheduling does not make use of that available time to recover the media units that were lost during the handover.

(4) Stream switching may be applied after a handover to achieve a reduction in the server output rate. This, however, may turn out to be unnecessary as the channel bitrate after the handover will usually not undergo significant changes.

In this work, we address all of the previously mentioned drawbacks in a new proposal for a fast buffer refill algorithm. The rest of the paper is structured as follows. Section 2 outlines the proposed fast buffer refill algorithm and describes the scheduling algorithm. Section 3 discusses the experimental results. The paper is concluded in Section 4.

FAST BUFFER REFILL ALGORITHM

While performing a hard handover, the mobile device temporarily loses the connection to the network. During this period, the network will buffer the data packets that are sent to the mobile device. This will ultimately lead to high transmission delays that affect the data packets sent during the handover period. The network buffer may overflow and cause packet loss.

The mobile receiver also maintains a receiver buffer to smooth out transmission delay jitter. Incoming media packets are queued in the receiver buffer until the decoding time of the corresponding media units is reached.

During a long handover period, the receiver buffer drains as no new media units are arriving. Fig.2 depicts the receiver buffer level in bytes, in media units (network abstraction layer units, i.e. NAL units, in the case of H.264/AVC video), and its playout duration, during and after a handover. The buffer playout duration is calculated as the time span between the timestamps of the oldest and most recent media units in the receiver buffer.

The figure shows a drop in the buffer duration and occupancy, which is due to data consumption without data coming in at the receiver. Just after the handover, the buffer starts to fill up in a monotonically increasing manner, which reflects the picture freezing that takes place as a consequence of the data loss. Without taking the necessary measures to refill the buffer, those picture freezing actions may be annoying to the viewer and may also cause the receiver buffer to underflow. This by consequence will lead to a playout interruption for long periods of time and will trigger re-buffering. This is usually perceived as more annoying by the viewer than playout with a reduced frame rate.

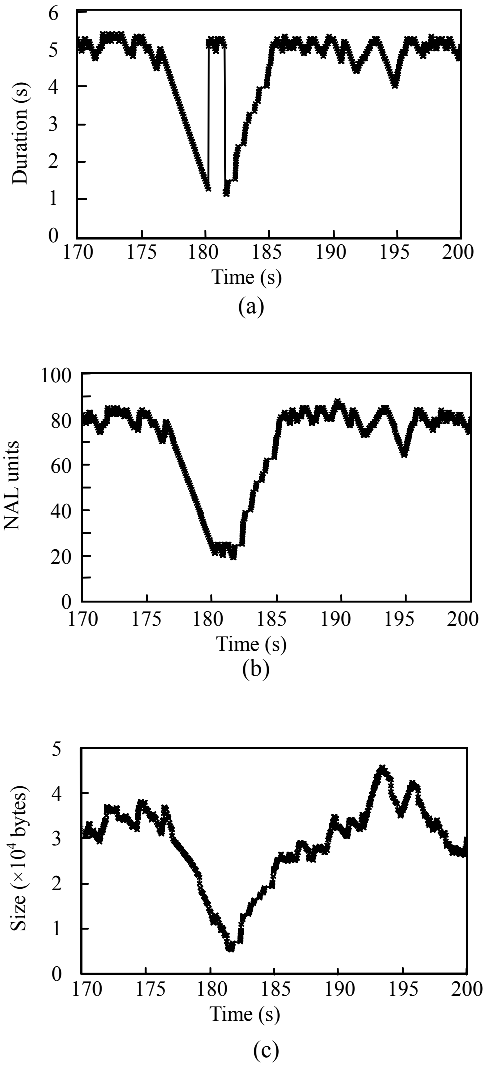


Fig.2 Receiver buffer during and after a hard handover operation. (a) Receiver buffer duration; (b) Amount of NAL units in the receiver buffer; (c) Receiver buffer occupancy

Detection of handover

In this paper, we propose a mechanism for detection of hard handovers without referring to lower layer indications. In (ETSI TS 126 234 V 6.5.0, 2005), the 3GPP packet switched streaming (PSS) mandates the usage of the RTP profile for audio and video conferences with minimal control (Schulzrinne and Casner, 2003). It also recommends the usage of the extended RTP profile for RTCP-based feedback (RTP/AVPF) (Friedman *et al.*, 2003). Furthermore, 3GPP PSS recommends the implementation of the RTP control protocol extended reports (IETF RFC 3611) and also specifies the next application data unit (NADU) RTCP APP packet for client buffer feedback (ETSI TS 126 234 V 6.5.0, 2005). An overview of how to use the different feedback mechanisms defined by 3GPP PSS can be found in (Kampmann and Baldo, 2004).

RTP and the supported RTP profiles allow for periodic reporting sent to the sender by the receiver. The frequency of these reports can be adjusted by appropriately setting the bandwidth reserved for these reports in the session description using the RR bandwidth modifier. The receiver then estimates the average time between two consecutive RTCP reports as follows (where a single receiver is assumed),

$$\Delta_{\mathrm{RR\_avg}}=\frac{S_{\mathrm{RTCP\_avg}}}{RR\times(e-1.5)}, \tag{1}$$

where $S_{\mathrm{RTCP\_avg}}$ is the average RTCP packet size, and *RR* is the bandwidth reserved for reception reports as described before.

We propose to use the RTCP receiver reports as an indication for hard handovers. Since hard handovers may last for up to several seconds, the handover period will usually cause delay and loss of several consecutive receiver reports. Each time the server receives a reception report, it should calculate the time period between the reception of the current and last reception reports. If the time period exceeds a predefined threshold value, the server should assume that a handover took place. The threshold value should be selected appropriately to take into account packet delays and losses due to other factors that could hinder in-time reception of reception reports. The threshold value could e.g. be set as follows:

$$T_{\text{thr}}=m\times\Delta_{\text{RR\_avg}}, \tag{2}$$

where m can be adjusted according to the expected channel conditions in order to cover long enough delay and loss burst periods.

The server may also use the feedback messages in the RTCP compound messages to make a more precise decision to detect the position of a handover within the transmitted packet sequence. This may e.g. be performed using negative acknowledgement (NACK) feedback messages or the loss fraction information. The RTP retransmission payload format (Rey *et al*., 2005) can be used to signal the packet losses to the streaming server using feedback messages within extended receiver report blocks.

It is recommended that enough bandwidth is reserved for receiver reports, so that the time gap between two consecutive reports is fairly short (in the order of a few hundred milliseconds). This will allow for frequent reporting to the server and more accurate estimation of the interruption period.

Rate adaptation after handover

During the setup of a streaming session, the receiver may inform the server about the conditions of its wireless link using the “3GPP-Link-Char” header field, which may be conveyed to the server in SETUP, PLAY, OPTIONS, or SET_PARAMETER RTSP (Schulzrinne *et al*., 1998) messages. The parameters included in the “3GPP-Link-Char” header field include the guaranteed bit-rate and the maximum bit-rate of the link.

Furthermore, the server can use the RTCP reception reports and the feedback messages from the receiver to estimate the available channel bandwidth. Curcio and Leon (2005) summarized the tools and feedback mechanisms available in 3GPP PSS for rate adaptation. Bouazizi and Wenger (2004) presented a rate estimation algorithm based on RTCP feedback information that can be deployed for rate adaptation.

The information on the channel bandwidth is then used by the server to increase its output rate after a hard handover period in order to refill the receiver buffer at a faster pace. The aim is to restore the buffer duration to its state before the handover took place.

Fig.2a depicts the buffer duration during and after a hard handover. The buffer duration starts decreasing during the handover, as the receiver is only consuming data. After the hard handover is over, the receiver starts receiving new RTP packets, which have a significantly larger playout time than the media units already present in the receiver buffer. In consequence, the buffer duration is suddenly increased. This, however, does not reflect the amount of data in the receiver buffer as shown by figures (Figs.2b and 2c). The buffer duration drops down to a low level after all the media units that were received prior to the handover are consumed. The result is then a long picture freeze period, where the receiver has nothing to playout, as the corresponding media units were lost during the hard handover. If the handover period lasts longer than the receiver buffer duration prior to the handover, the receiver buffer will underflow and the playout will be completely interrupted for a long re-buffering period.

The fast buffer refill algorithm, introduced in this paper, aims at recovering from the handover by reducing the playout interruption and bringing the receiver buffer duration to its original level prior to the handover. This will allow us to cope with future hard handovers while providing a reasonable video quality after the handover period. The fast buffer refill is essentially done by partially retransmitting the identified missing media units at a maximum data rate, as indicated with the maximum bit-rate parameter in the “3GPP-Link-Char” header field or concluded by the server otherwise, during a short time period. The fast buffer refill period can be adjusted to be long enough to achieve reasonable video quality and short enough to end up before the next handover. It should also not be too long so that it does not incur a high traffic load and cause network congestion by sending at higher bitrate for a long period of time.

Optimal scheduling for fast buffer refill

Whenever a hard handover is detected, the server triggers the fast buffer refill algorithm. The server calculates in a first step the output rate as described in Section 2.2. It then selects a set of media units out of the look-ahead period and schedules them for transmission at the calculated output rate.

In this section, we describe the scheduling algorithm used to achieve optimal video quality when deploying the fast buffer refill algorithm. Even though the described algorithm is presented only for video, it can be extended to cope with other real-time media types in the same streaming session.

The presented scheduling algorithm uses the temporal scalability features of H.264/AVC coding, which are introduced in this paragraph.

There are two temporal scalability methods in H.264/AVC, which complement each other. First, non-reference pictures are such that they are not used as prediction reference for any other pictures in the bitstream and hence they can be disposed of without affecting the decoding of the remaining pictures. Non-reference pictures can be of any picture coding type (e.g. inter predicted pictures or bi-predicted pictures).

Second, the sub-sequence technique provides means for hierarchical temporal scalability, in which pictures are arranged in sub-sequence layers and sub-sequences within those layers. A sub-sequence consists of a number of inter-dependent pictures that can be disposed of without affecting the decoding of any other sub-sequence in the same sub-sequence layer or any sub-sequence in any lower sub-sequence layer. Fig.3 presents an example of a sub-sequence consisting of a bi-predicted reference picture and two bi-predicted non-reference pictures. The sub-sequence technique enables easy identification of disposable chains of pictures when processing pre-coded bitstreams. Sub-sequence layers are arranged hierarchically based on their dependency on each other. The base layer (layer 0) is independently decodable. Sub-sequence layer 1 depends on some of the data in layer 0, i.e., correct decoding of all pictures in sub-sequence layer 1 requires decoding of all the previous (in decoding order) pictures in layer 0. In general, correct decoding of sub-sequence layer N requires decoding of layers from 0 to N−1. It is recommended to organize sub-sequences into sub-sequence layers in such a way that discarding of enhanced layers results in a constant or nearly constant picture rate. Picture rate and therefore subjective quality increase along with the number of decoded sub-sequence layers. The sub-sequence technique is presented in more details in (Tian *et al.*, 2005). The paper also concludes that both temporal scalability features are beneficial in terms of compression efficiency when compared to non-scalable bitstreams and that certain sub-sequence structures improve compression efficiency compared to the use of non-reference pictures.

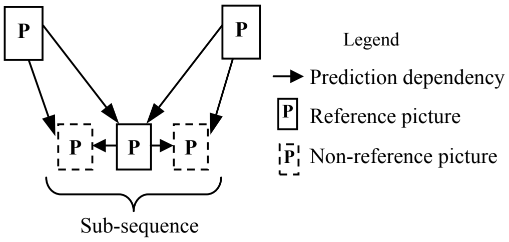


Fig.3 Example of sub-sequence usage

The proposed scheduling algorithm is based on assigning the pictures of the video stream into significance classes according to their sub-sequence layers. If no sub-sequence layering is defined for the stream, all reference pictures are considered to be in sub-sequence layer 0. The significance of a sub-sequence layer is further refined as follows. For the base layer, i.e. sub-sequence layer 0, the following order of significance, from highest to lowest, is used:

(1) Pictures starting a base layer sub-sequence and which are coded as a response to scene or shot change. Various methods have been proposed for detection of scene changes. Wang *et al.*(2003) describes a method to detect scene changes from coded sequences using scene information Supplemental Enhancement Information messages.

(2) All other pictures starting a base-layer sub-sequence.

(3) All other pictures in a base-layer sub-sequence.

Non-reference pictures are always classified in the lowest significance class below the significance class of any reference pictures. This classification coarsely reflects the significance of the pictures to the decoding process and the perceived visual quality.

Based on this classification method, the scheduling algorithm is aimed at achieving a receiver buffer duration D after a fast buffer refill period $\Delta_{\mathrm{refill}}<D$. As discussed in Section 2.2, Δ_{refill} should be selected carefully. A value approximately equal to the length of the handover period is generally a good choice for the length of the refill period.

During Δ_{refill}, the server is sending at an increased rate R_{refill}, as indicated in the previous section, in order to achieve a fast buffer refill. In order to do so, the server adjusts the gap between every two consecutive packets to match the target rate R_{refill}.

The total amount of data sent during the fast refill period can then be calculated as follows:

$$S_{\text{total}}=R_{\text{refill}}\times\Delta_{\text{refill}}. \tag{3}$$

Given this constraint on the total amount of data to send during the fast buffer refill period, the server performs a look-ahead operation of at least $\Delta_{\text{refill}}+D$ seconds, to select a set of pictures (or NAL units) to be transmitted during the buffer refill duration Δ_{refill}, in order to reach the target receiver buffer duration time D and at the same time to maximize the playout quality. For this purpose, the server builds up a set of pictures that belong to the timeframe starting at the last correctly received picture before the handover and ending after $\Delta_{\text{refill}}+D$ seconds. Let us assume that this operation results in n pictures.

This optimization problem can be formulated as follows:

$$\max\sum_{i=1}^{n} c_i\cdot x_i\cdot\prod_{j\in references(i)} x_j, \tag{4}$$

where

$$\sum_{i=1}^{n} s_i\cdot x_i\le S_{\text{total}}, \tag{5}$$

and

$$t_{\max}-t_{\min}\ge D+\Delta_{\text{refill}}, \tag{6}$$

where

$$t_{\max}=\max_{i\in(1,\ldots,n)}\{t_i\cdot x_i\}, \tag{7}$$

$$t_{\min}=\min_{i\in(1,\ldots,n)}\{t_i\mid x_i=1\}. \tag{8}$$

$x_i\in\{0, 1\}$ defines whether the ith picture is scheduled for transmission or not. *references*(i) indicates all (direct and indirect) reference pictures to picture i. c_i indicates the significance class of the ith picture and s_i the size of that picture. The $t_{\min}$ and $t_{\max}$ are the smallest and largest picture display timestamps of pictures selected for transmission respectively.

The following algorithm is then used to solve this problem:

(1) Set all x_i to 0; Set the current significance class C to the highest significance class; set current picture index i to n; set total size S to 0;

(2) Check if picture i belongs to significance class C and x_i=0 and $S+s_i\le S_{\text{total}}$ and the reference pictures of picture i, if any, are scheduled for transmission;

(3) If true, schedule the picture for transmission by setting x_i=1. Update S to be $S\leftarrow S+s_i$, goto Step 10;

(4) Check whether all pictures of the current significance class are handled;

(5) If not, update i to be the index of the next picture. Go to Step 2;

(6) If all pictures of the current significance class are handled, check whether all significance classes are handled;

(7) If all significance classes are handled, go to Step 10;

(8) Else, set C to the next lower significance class and set the current picture index i to be either n for significance classes containing Independent Decoding Refesh (IDR) pictures and other INTRA coded pictures or 1 for other significance classes. Go to Step 2;

(9) Check that the receiver buffer duration satisfies the constraint. If true then terminate the search algorithm;

(10) If not increase the look-ahead and restart from Step 1;

(11) Else, terminate the search algorithm.

This algorithm selects IDR and INTRA coded pictures in reverse order starting from the end of the look-ahead buffer, in order to maximize the probability of in-time arrival of the transmitted pictures. The NAL units are, however, transmitted to the receiver in the decoding order. This will allow for pictures, whose playout time is earlier, to arrive in time at the receiver.

It then picks up other pictures in the decoding order, to assure that all decoding dependencies are considered correctly. The algorithm can be improved further by trying to achieve equal distribution of the transmitted pictures among the different sub-sequences. This can be easily achieved by modifying the search algorithm for pictures (other than INTRA and IDR pictures) to select pictures sequentially from each sub-sequence (i.o.w. the algorithm would select one picture and jump to the next sub-sequence).

EXPERIMENTAL RESULTS

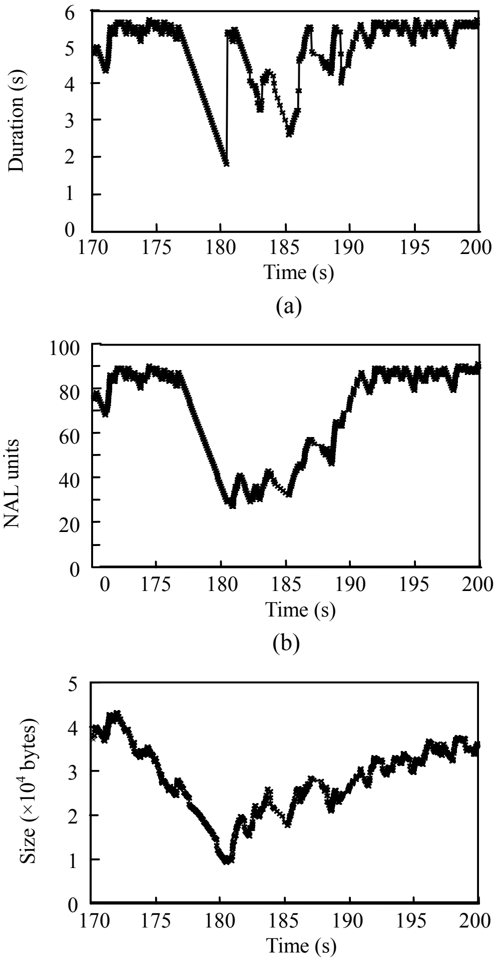
In this section, we discuss the experimental results obtained using a mobile network emulator and running a streaming session. In the used mobility scenario, a mobile device is receiving a streaming session from a fixed streaming server. The mobile device is moving back and forth between 2 base stations. Each time the mobile device crosses the boundary of the other cell, a hard handover takes place.

The used video sequence is a looped Foreman sequence of length 4 min. The video sequence is encoded using H.264/AVC at a bitrate of about 48 kbps with two non-reference pictures between each two consecutive reference pictures. The IDR frequency rate is at one in every 32 pictures and the frame rate of the sequence is 15 fps.

During the simulation, the streaming session is initiated manually, which explains the slight discrepancy in the starting time of the handover period in the studied cases. However, this does not reduce the reliability of the results as the mobility scenario is exactly the same and the difference is lower than 1 s.

The results for the streaming scenario without handover detection are shown in Fig.2. Fig.4 depicts the same results for the scenario where the fast buffer refill algorithm is activated. The buffer level curves (buffer occupancy and number of NAL units) show the difference in the playout behavior between the two different algorithms. In the case of no handover detection, the buffer level keeps increasing without any data being consumed shortly after the handover. This reflects the period when a long picture freeze took place. In the case of fast buffer refill, the receiver is simultaneously receiving and consuming data units shortly after the handover period. This shows that no playout interruption took place, as the receiver continuously had media units to playout, although it was not playing at the original frame rate. On the other hand, the buffer duration after the fast refill period is on the same level as in the case where no handover detection occurred.

Fig.5 compares the PSNR curves for the corresponding pictures. The PSNR curve of the fast buffer refill scenario reflects the decisions of the fast buffer refill scheduling algorithm. It shows a longer degradation period than the no-handover-detection scenario, where the video quality loss is significantly smaller. The causative reason for the longer degradation period is the scheduling algorithm which picks up a subset of the pictures for transmission out of a period that lasts longer than the handover period. The average PSNR values are 32.55 dB for the fast buffer refill algorithm and 31.71 dB for no handover detection scenario.



(a)

(b)

(c)

Fig.4 Receiver buffer during and after a hard handover operation with fast buffer refill activated. (a) Receiver buffer duration; (b) Amount of NAL units in the receiver buffer; (c) Receiver buffer occupancy

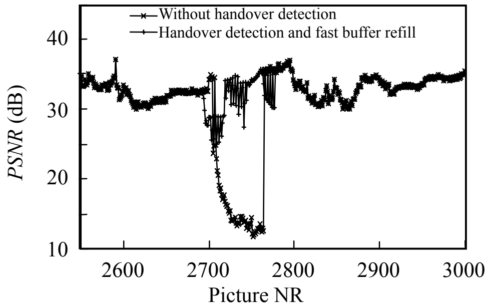


Fig.5 *PSNR* curves for the cases of no handover detection and handover detection with fast buffer refill algorithm

CONCLUSION

In this paper, we presented a fast buffer refill algorithm for 3GPP packet-switched streaming compliant servers, which copes with hard handover situations. The algorithm achieves significant improvements in terms of the video quality at the cost of slightly higher transmission bitrate during the refill period.

References

Banerjee, N., Wu, W., Das, S.K., Dawkins, S., Pathak, J., 2003. Mobility support in wireless Internet. *IEEE Wireless Communications*, **10**(5):54-61. [doi:10.1109/MWC.2003.1241101]

Bouazizi, I., Wenger, S., 2004. Distortion Optimized Rate Shaping for TCP-Friendly Video Streaming. Proceedings of the 14th International Packet Video Workshop (PVW 2004). Irvine, USA.

Curcio, I.D.D., Leon, D., 2005. Evolution of 3GPP Streaming for Improving QoS over Mobile Networks. IEEE International Conference on Image Processing. Genova, Italy.

ETSI TS 126 234 V 6.5.0, 2005. Universal Mobile Telecommunications System (UMTS); Transparent End-To-End Packetswitched Streaming Service (PSS); Protocols and Codecs. Technical specification, ETSI.

Friedman, T., Caceres, R., Clark, A., 2003. IETF I-D RTCP XR, RTP Control Protocol Extended Reports (RTCP XR).

Kampmann, M., Baldo, N., 2004. Adaptive Wireless Streaming Using Transmission Rate Control and Priority-Based Packet Scheduling. Proceedings of the 14th International Packet Video Workshop (PVW 2004). Irvine, USA.

Rey, J., Leon, D., Miyazaki, A., Varsa, V., Hakenberg, R., 2005. IETF I-D Retransmission, RTP Retransmission Payload Format.

Schierl, T., Wiegand, T., 2004. H.264/AVC Rate Adaptation for Internet Streaming. Proceedings of the 14th International Packet Video Workshop (PVW 2004). Irvine, USA.

Schierl, T., Kampmann, M., Wiegand, T., 2005. 3GPP Compliant Adaptive Wireless Video Streaming Using H.264/MPEG4-AVC. Proceedings of the ICIP 2005. Genova, Italy.

Schulzrinne, H., Casner, S., 2003. IETF RFC 3551, RTP Profile for Audio and Video Conferences with Minimal Control.

Schulzrinne, H., Rao, A., Lanphier, R., 1998. IETF RFC 2326, Real Time Streaming Protocol.

Tian, D., Hannuksela, M.M., Gabbouj, M., 2005. Sub-Sequence Video Coding for Improved Temporal Scalability. Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS).

Wang, Y.K., Hannuksela, M.M., Caglar, K., Gabbouj, M., 2003. Improved Error Concealment Using Scene Information. Proceedings of the International Workshop VLBV03. Madrid, Spain.

Wisely, D., Eardly, P., Burness, L., 2002. IP for 3G-Networking Technologies for Mobile Communications. John Wiley and Sons.