Synchronization Scheme of Multimedia Streams in Mobile Handoff Control

Gi-Sung Lee

Dept. of Computer Science, Howon Univ., Korea
ygslee@sunny.howon.ac.kr

Abstract. This paper presents a synchronization scheme for enabling a smooth presentation of multimedia streams during a handoff in mobile environments. As changes are made to base stations in wireless environments, mobile host-initiated handoffs result in multimedia data loss in a base station as well as low quality of service (QoS) for multimedia streams. As a result of evaluations, the proposed schemes, compared to the previons schemes, delivered a continuous playout of multimedia streams while achieving low packet loss.

1 Introduction

Driven by the drastically increasing use of the Internet, mobile computing-based services are emerging in rapid succession. Presentations of multimedia data stored in a mobile host are not readily implemented, largely due to the dynamic nature of mobile network's connectivity, for example, higher data loss rate, higher delays, and lower network bandwidth[6]. For this reason, buffers are used in a lot of distributed multimedia systems connected to wireless networks in order to address network-specific problems including latency and data loss. A Base Station(BS) transmits subframes from its several multimedia servers. If, due to unexpected delay and increased traffic, their projected playout time is faster than their actual arrival time, the playout of subframes cannot be implemented[4,7]. In order to address these problems, buffering at a BS is required to reduce packet delay and jitter between a multimedia server and a BS. One advantage of a mobile network is host mobility. Though, this advantage can become a disadvantage due to some cumbersome processes; not only resources must be secured for transmission of multimedia streams but also multimedia streams transmitted to a BS must be transmitted again. With the purpose of minimizing those drawbacks, this paper places focus on configuring two jitter buffers for a BS and one jitter buffer for a mobile host. Streams occurring and lost during a handoff are transmitted to an Old BS (BS_{old}), and substreams in odd numbered buffers are transmitted to a New Base Station (BS_{new}). During this process, a mobile host is adjusted to the maximum playout delay jitter. This scheme implements the playout of substreams in BSnew buffer that require retransmission, eliminating the need for retransmission. However, media loss adversely affects playout.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 389–396, 2004.

2 Related Work

Research has reached a level where mobile-based synchronization schemes and previous schemes are converging. M. Woo, N. U. Qazi, and A. Ghafoor defined the BS as wired/wireless interface. For wired networks, the interface is defined as a BS buffer used to reduce delay and jitter between packets. The disadvantage of this approach is its attempt to apply synchronization to buffers through assignment of existing wireless communication channels [5]. D. H. Nam and S. K. Park proposed a buffer management and feedback scheme that allows transmitted multimedia unit storage in the buffer of a MH(MH), and that adjusts playout time to the medium buffer length [2,3]. However, there is a disadvantage to this approach that the adjustment of palyout time lowers the quality of multimedia data even when a mobile network is in a normal state. That's because the stable state does not correspond to the buffer of a MH. Azzedine Boukerche proposed an algorithm that enables message transmission among multimedia servers, BS and mobile hosts while interfering with multimedia servers as well as startup time for transmission [1]. The merit of this approach is a well-defined role of the multimedia server, BS, and MH. Nonetheless, this scheme has some disadvantages in terms of its lack of a playout policy as well as the application of only startup time for communications transmission as means of solving the synchronization problem.

3 Mobile Network-Based Buffer Management Scheme

3.1 System Configuration

This system supports a k multimedia server node, m BS, and n MH. The BS communicates with an i mobile host in mth cell. The MH is necessarily supposed to access the server via the BS. This system allows a BS to manage variations of startup time for transmission as well as buffers, using variables such as delay jitter and arrival time of subframes transmitted from the multimedia server. Therefore, this system contributes to effectively dealing with mobile communications-specific small memory footprint and low bandwidth. Some of the advantages of the proposed scheme are reduced server startup time, lighter network traffic, and reduced buffer size, all of which are achieved by storing a movie in the k server. The multimedia server saves multimedia data in a distributed way that splits data streams according to logical time. Data streams are compressed data split by logical time. Accordingly, their configurations are made according to a synchronization group, not the same byte. Such split streams are called substreams, and this technique is referred to as subframe stripping. Figure 1 shows the configuration of the proposed system. The Message Manager in the multimedia server skips and transmits subframes using the offset control mode notified by the Feedback Manager at the BS. The BS saves the arrival time of the subframes received by individual servers, and it sends/receives a dummy packet to identify the startup time for transmission. The Communication Manager in the multimedia server distinguishes feedback packets from messages notified by the BS, and the Communication Manager in the BS arranges the sub-frames received from individual servers in an orderly manner.

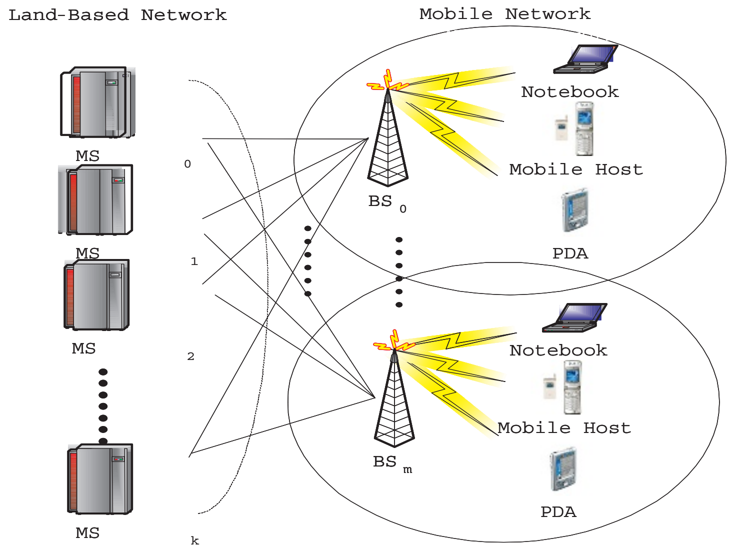


Fig. 1. System Configuration

3.2 Handoff Management

Due to its use of soft handoff, CDMA systems have some problems involving multiple BSs. Handoff management eventually leads to the increase in channel capacity. The information such as multimedia units, data, and voice is transmitted via multiple BSs. We also present an algorithm that enables a mobile host to implement the playout of media streams, without any multimedia data loss, or without any additional delay accompanying message transmission, or within the reasonable QoS limits during a handoff. In this scheme, the MH (HM_i) transmits multimedia services via the BS (BS_m) from the Server (S_k). The BS is classified into the primary and non-primary BSs. The former is responsible for sending multimedia streams to the MS, and the latter adjoins the primary BS. The BS buffer is of the two-jitter size, and the MH buffer is of the one-jitter size. Compared against buffer configurations made only for the BS, buffer configurations like this gives the advantages of no multimedia data stream loss during a handoff while complementing the MH with small memory footprint. Figure 2 shows the system configuration applied when a handoff is being processed. A handoff occurs when MH_i moves from $BS_{current}$ to BS_{new}.$BS_{current}$ sends a handoff message to the multimedia server, and starts to implement handoff processing in the following sequence:

1. MH_i implements playout as long as (1jitter $\times$ r) $\times$ λ time, regardless of handoff. /* r refers to the playout speed of a substream, and λ means the maximum jitter within the media */
2. Setting is made to $BS_{old} = BS_{current}$.
3. Setting is made to $BS_{current} = BS_{new}$.
4. BS_{old} sends a $Handoff_{on}$ message to the multimedia server. Then, the multimedia server gives a notification to stop transmission of multimedia data streams.
5. BS_{new} receives only odd numbered multimedia data streams present in BS_{old}.
6. The multimedia server sends multimedia streams only to the servers that contain odd numbered subscripts.
7. Once BS_{new} has received odd numbered multimedia data streams from BS_{old} buffer, it sends a $Handoff_{off}$ message to the multimedia server to inform that handoff has been successfully dealt with.
8. BS_{new} sends a request message to individual servers and normal transmission is made.

3.3 Synchronization Algorithm on Handoff

We now formally describe the algorithm for a BS_m, a MH_n, and a, S_k by means of pseudocode. Handoff control algorithm describes the variables used by Pseudocode. The handoff occurs when MH_m moves from the current BS_m to another BS_m.

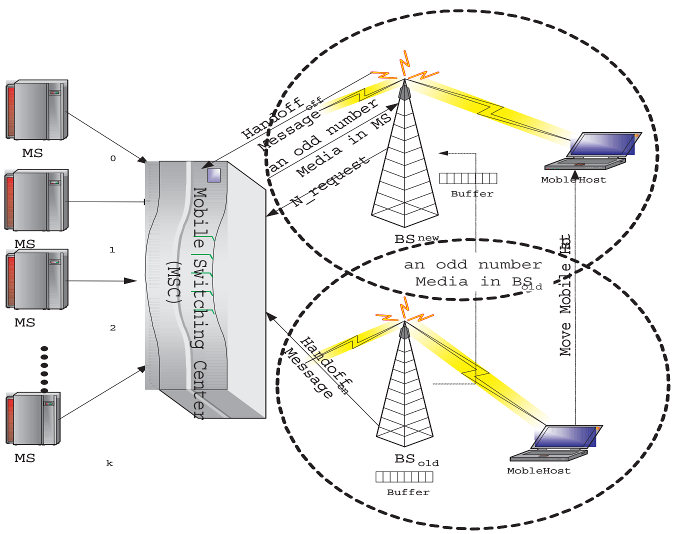


Fig. 2. Handoff Process Structure

During the handoff control process, multimedia data streams stored in are sent to a new BS_m for playout within the QoS limits. The algorithm is designed to solve the problems arising out of handoffs by the use of an approach to multimedia synchronization. This paper deals with two types of messages, which are addressed as follows:
1. Send (source, action; argument);
2. Receive (source, action; argument);
where action ← Request| N-request| Reply | Update | INT | $Handoff_{off}$ | $Handoff_{on}$ | Done , and arguments $\subset \{MS_k^i, DT_k^i, RT_k^i, ST_k^i, S_k\}$.
The algorithm is described as follows:

Synchronization on Handoff

Mutimedia Server$_k$

Receive (BS_m**, DummyPacket);**{
  Send (BS_m**, reply**; **CurrentMultiserverTime);**}
Receive (BS_m**, Request; MMU**$_k^i$);**{**
  For $\mathbf{i = 0,\ i < k, i++}$ **Do**
    Send(BS_m**, reply; MMU**$_k^i$);**}**
Receive (BS_m**, N − Request; MMU**$_k^i$);**{**/∗**MMU**$_k^i$ **is i-th multimedia steame*/**
  Make schedule for (**N − 1**) **requests**
  Save RT_k^i **and** D_k^i**;** /∗ RT_k^i **is Round trip time*/**
  Send (BS_m**, reply; MMU**$_k^i$);**}**/ ∗ D_k^i **is Delay time*/**
Receive (BS_m**, Handoff**$_{on}$**);**{
Do
   Wait;
While (Receive(BS_{new}**, Handoff**$_{off}$))**}**

Mobile Host$_i$

  IF (Receive(BS_m**,Handoff**$_{on}$))**{**
  Play − out is MMU$_k^i$ **at** τ_t**+10ms**;**}**
  Else{
  Send(BS_m**,CurrentMultiserverTime, Requst : MMU);**
  Receive(BS_m, **MMU**$_k^i$; τ_t**);**
  Play − out is MMU$_k^i$ **at** τ_t;**}**

Base Station$_m$

   BufferControl(); {
  ρ_i =**125ms;**
    If (buffer_point == NormalLevel)
      $\tau_i = \rho_i$**; Send(**$MH_i,\ MMU_k^i,\ \tau_t$**);**
    Else If (buffer_point == UpperLevel){

```
        BL_allsize = BL_allsize+1;         /*BL allsize is overall buff size*/
        BL_psize = CurrentBuffer;          /*BL psize is buffer Level size*/
        ω = 0;  /* Weight Value */   ρ_i=125;
        ω = BL_allsize  / BL_psize;   τ_t = ρ_i − (ρ_i × ω);
        Send(MH_i, MMU_k^i, τ_t);}
      Else IF (buffer_point == LowerLevel){
        ω = BL_allsize − BL_psize;   ω = ω/BL_allsize;   τ_t = ρ_i + λ × ω;
        Send(MH_i, MMU_k^i, τ_t);}}    /*λ is Maximum delay jitter*/
  Receive(BS_n, Handoff_on)
    BS_old = BS_current;
    BS_current = BS_new;
    Send(MH_i, Handoff_on);
    Send(S_k, Handoff_on);
    For k=1, k ≤ K, k+2 Do
      BS_new(MMU_k^i) = BS_old(MMU_k^i);
    Send(S_k, Handoff_on);   /* from BS_new to S_k */
  Send(S_k, Request; MMU_k^i); }

  Main(){   /* Main Program */
  Call Start − UpTime;   /*Δ is jitter*/
  Start-Up=MAX D_k^i − D_k^i;
  Set Start Up Time for Server S_k;
  Send(S_k, Start Up Time);
  Δ = DT_k^max − DT_k^max  /* Δ is jitter */
  If (Buffer_point == NomalLevel)
  Send(S_k, MMU_k^i, Request;MMU_k^i)
  Else If (Buffer_point==UpperLevel){
      If (DT_k^i > Δ){
      Feedback_value = Call Feedback;
      Send(S_k,Feedback_value, D_k^i);}
      Else{
      Continue;
      Call BufferControl(UpperLevel);}}
  Else If (Buffer_point == LowerLevel){
      If (DT_k^i > Δ){
      Feedback_value = Call Feedback;
      Send(S_k,Feedback_value, D_k^i);}
      Else{
      Continue;
      Call BufferControl(UpperLevel);}
    }}end of main
```

4 Performance Evaluation

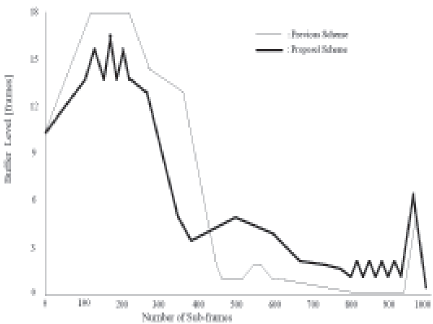
Experiments were carried out using IBM-compatible PCs with Pentium processor, and interfaces and algorithms were implemented using Java development

Kit JDK 1.3. Outputs obtained from the experiments were stored in Microsoft MDB as simulation.mdb files. 1Kbyte audio data were encoded using a PCM encoding technique, and the video frames used had a resolution of 120 X 120 pixels. The frames used in the experiment were those encoded with 24 frames per second. This paper assumes simulations have been performed in a mobile environment. In order to process individual packets properly, the information used in the actual simulations was applied equally to mobile networks using Poisson distribution. One thousand frames were used in performance evaluation experiments where the maximum delay jitter time of 600 ms was applied. This paper presents our work on the algorithm that disallows any loss of data in the BS buffer to smooth the playout of multimedia streams during a handoff. The comparative evaluation presented in this paper is focused not only on the feedback policy toward maintaining the buffer at a normal level but also on the play-out and loss time where the buffer level control-based playout policy is applied Figure 3 shows how much the level of the buffer varies. As illustrated in Figure 3, the data stored in BS_{old} buffer are sent to BS_{new} buffer in order to prevent streams against loss in the frames numbered from 800 through 960 during a handoff. Additionally, during the handoff, MH_i delays the playout time of MMU_k^i to the maximum delay jitter of 10ms in order to secure more time for moving substream data in BS_{old} buffer to BS_{new} buffer. As shown in Figure 3, existing conventional schemes appear to be less effective in dealing with handoffs, thus suffering from loss of substream data stored in BS_{old} buffer. Figure 3 shows that overflow occurs in the frames numbered from 150 through 200, and also shows that the buffer level of the proposed scheme is kept in a more stable state than existing schemes. As it were, existing conventional schemes could not avoid starvation occurring in the buffer due to handoff, as the results obtained in figure 3. However, the proposed scheme was able to prevent starvation from occurring in the frames numbered from 800 through 960. Furthermore, it could prevent overflow from occurring in the frames numbered between 150 and 200.

Figure 4 depicts what effect the proposed scheme has on the buffer level-based playout policy. The proposed scheme delays playout time even though a handoff has occurred in the frames numbered from 800 through 960. For existing conventional schemes, however, the occurrence of a handoff results in the loss of data in BS_m buffer, causing not only a failure of the playout of data streams but also skipped frames numbered between 150 and 200. For the proposed scheme, the playout time of the frames numbered from 800 through 960 varies according to the playout policy while the playout time of the frames numbered from 150 through 200 decreases gradually, ultimately leading to the reduction in overflow.

5 Conclusions

This paper presents a scheme that disallows lower playout rates within the QoS limits for multimedia presentations during a handoff in mobile networks. The proposed scheme enables a base station to manage buffers and playout policies, and further deals with handoffs in a fast and efficient way. Additionally, this scheme offers a suitable approach to effectively dealing with limiting factors for



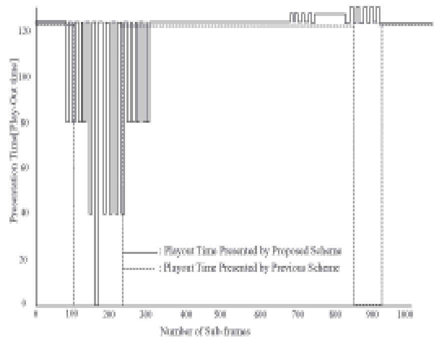


Fig. 3. Buffering Fluctuations at BS

Fig. 4. Playout Time Presented by Proposed Scheme

mobile communications such as small memory footprint and low bandwidth. The proposed scheme's adaptive playout time delivers soft and smooth handoff performance where the entire playout time is maintained within the playout time of the original's.

References

1. Azzedine Boukerche, Sungbum Hong and Tom Jacob, "MoSync : A Synchronization Scheme for Cellular Wireless and Mobile Multimedia System", Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems IEEE 2001
2. D. H. Nam and S. K. Park, "Adaptive Multimedia Stream Presentation in Mobile Computing Environment," Proceedings of IEEE TENCON, 1999.
3. Ernst Biersack, Werner Geyer, "Synchronization Delivery and Play-out of Distributed Stored Multimedia Streams", Multimedia Systems , V.7 N.1 , 70-90, 1999
4. M. Woo, N. U. Qazi, and A. Ghafoor, "A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information," IEEE Network, Jan./Feb. 1994
5. T. D. C. Little, and Arif Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," IEEE Journal on selected Areas in Comm., Vol. 9, No.9, Dec. 1991.
6. W. Geyer, "Stream Synchronization in a Scalable Video Server Array," Master's thesis, Institute Eurecom, Sophia Antipolis, France, Sept., 1995.
7. Gi-Sung Lee, Jeung-gyu Jee, Sok-Pal Cho, "Buffering Management Scheme for Multimedia Synchronization in Mobile Information System," Lecture Notes in Computer Science Vol. 2660, pp 545-554, June, 2003.