

MPEG Video의 핸드오프 제어 기법

구중두

호원대학교 컴퓨터게임학부

e-mail:ygslee@sunny.howon.ac.kr

Handoff Control Scheme of MPEG Video

Jung Du Goo

Division of Computer Games, Howon University

요 약

본 논문은 인터넷 및 모바일 통신에서 많이 사용되는 MPEG video을 모바일 통신망에서 전송 및 재생하고, 모바일 환경에서 발생하는 Handoff 상에서 부드러운 스트림(stream) 재생(presentation)을 위한 멀티미디어 동기화 기법을 제안한다. 본 논문에서는 멀티미디어 동기화 기법으로 핸드오프 시 MPEG 데이터의 특성을 이용한 핸드오프 제어 방법을 제안하였다. 분석결과, 제안된 기법은 기존의 기법보다 미디어 스트림의 끊김없는 재생(stream of continuous play out)을 제공하는 것뿐만 아니라 패킷의 높은 재생률(play out rate)과 낮은 손실률(loss rate)을 보였다.

1. 서론

2. 관련 연구

최근의 폭발적인 인터넷 이용 증가 경향은 모바일 컴퓨팅 환경을 기반으로 하는 서비스들의 등장 가속화하고 있다. 이에 따른 기존의 단순한 클라이언트와 서버간의 통신을 기반으로 하는 서비스 구조로는 다양한 멀티미디어 서비스를 고속으로 제공하고자 하는 이용자의 요구를 만족시킬 수 없게 되었다. 이러한 기존 통신서비스 구조의 한계는 서비스 제공 시스템의 개념을 단일 시스템에서 무선망으로 연결된 모바일 시스템으로 확장하면서 극복할 수 있게 되었으며, 모바일 환경은 이러한 성능 확장을 가능하게 하는 핵심적인 기능을 제공하였다. 특히 멀티미디어 서비스에서는 모바일 환경의 멀티미디어 서버에 분산 저장하는 방식(distributed store technique)에 많은 관심을 갖게 되었다.

또한 핸드오프 시 발생되어 잃어버리는 MPEG video 의 GOP(I,P,B) 을 구 베이스 스테이션(BSold)의 버퍼의 MPEG video 의 I Picture 를 새로운 스테이션(BSnew)으로 전송하게 된다. 이때 모바일 호스트는 최대 지연 지터 만큼을 느리게 재생하게 된다. 이렇게 함으로서 재전송되어야 할 BSnew 의 버퍼의 서브 스트림을 재전송 없이 재생할 수 있는 기법을 제안한다. 이러한 결과는 미디어가 손실됨으로서 재생에 영향을 미치게 된다.

무선 통신 상에서 제안된 동기화 기법은 기존의 동기화 기법을 접목 시키는 수준에 이르고 있다.

D. K. Y. Yau, S.S. Lan 에서는 비디오 서버의 CPU 에서의 프레임 비율조정 정책으로 프레임의 전송 비율에 따라 CPU 의 프로세서 타임을 주기마다 조정하여 스케줄링이 이루어진다. 커널 스레드를 사용하여 통신망의 부하를 모니터링하고, 이에 따라 프레임 전송 비율을 증감하는 방법으로 트래픽을 감소시키는 방법을 사용한다. 하지만 이 정책은 서버에서의 제어만 이루어진다는 단점을 가지고 있다[1] M. Woo, N. U. Qazi, and A. Ghafoor 는 기지국은 유/무선 사이의 하나의 인터페이스로 정의하였다. 유선망에서는 인터페이스는 패킷 간의 지연지터를 줄이기 위해 기지국에서 버퍼링을 정의하였다. 이 기법의 단점은 버퍼의 기법을 기존 이동통신의 채널 할당으로 동기화를 적용하려고 하였다[2].

3. 모바일 네트워크에서 MPEG 의 GOP 데이터의 핸드오프 기법

3.1 핸드오프 제어

모바일 호스트가 핸드오프가 발생될 때, any additional delay의 메시지 전송 없이 그리고, 멀티미디어 데이터의 손실 없이 그리고 QoS 범위 내에서

미디어 스트림을 Playout 하고자 하는 것이 이 논문의 목적이 된다.

이 논문에서는 방법은 모바일 호스트가(HM_i)가 멀티미디어 서비스를 서버 S_k 에서 베이스 스테이션 BS_m 을 통하여 전송되게 된다. 이때 Primary BS와 Non primary BS의 두가지 타입으로 구분된다. 여기서 Primary BS는 MH에게 MPEG의 GOP 단위로 전송하는 BS이고 Non primary BS는 이웃하고 있는 BS이다. 또한 BS 내의 버퍼는 2지터의 버퍼의 크기를 갖고 있으며, MH에는 1지터의 버퍼의 크기를 갖게 된다. 이러한 버퍼의 구성은 BS에만 버퍼를 구성하는 방식 보다 핸드오프가 발생하여도 멀티미디어 스트림을 잃어버리지 않는다는 장점을 가지고 있다. 또한 MH의 적은 메모리의 구성을 보완할 수 있는 방법이 된다.

3.2 핸드오프 제어 알고리즘

본 논문에서는 두 가지 유형의 메시지가 있다. :

1. Send (source, action; argument);
2. Receive (source, action; argument);

본 논문에서는 MH_m 이 현재의 BS_m 에서 다른 BS_m 로 옮겨 가면 핸드오프가 발생되게 된다. 이때 BS_m 에 저장되어 있는 데이터를 새로운 BS_m 으로 멀티미디어 스트림 데이터를 전송하여 QoS 범위에서 Presentation하는 것이며, 이러한 문제를 멀티미디어 동기화 측면에서 접근하여 핸드오프의 문제를 해결하고자 하였다. 이러한 알고리즘은 다음과 같다.

Procedure Handoff Control Algorithm

Mutimedia Server_k

Procedure MutimediaServer_k

Begin

Receive(BS_m , DummyPacket);

Send(BS_m , reply; CurrentMultiserverTime);

Receive(BS_m , Request; GOP_k^i);

For $i = 0$, $i < k$, $i++$ Do

Send(BS_m , reply; GOP_k^i);

Receive(BS_m , UpperLevel; GOP_k^i);

Make schedule for requests start-up time

Send(BS_m , reply ; $GOP(I,P,B)_k^i$);

Receive(BS_m , LowerLevel; GOP_k^i);

Make schedule for requests start-up time

Send(BS_m , reply ; $GOP(I)_k^i$);

Receive(BS_m , Handoff_{on});

Do

Wait;

While (Receive(BS_{new} , Handoff_{off}))

Mobile Hosts_i

Procedure Mobile Hosts_i

Begin

Send(BS_m , CurrentMultiserverTime, Request ; GOP_k^i);

Receive(BS_m , GOP_k^i ; τ_t);

Play-out is GOP_k^i at τ_t ;

Receive(BS_m , Handoff_{on});

Play-out is GOP_k^i at $\tau_t + 10ms$;

End

Base Station_m

Procedure Start-up Time

Begin

BS_time = Current BS Time;

For $i = 0$, $1 < k$, $i++$ Do

Send(BS_m , DummyPacket_i);

Receive(S_k , DummyPacket_i);

End For

End

Procedure FeedBack

Begin

$q_{t,m} = \text{CurrentBufferPoint}$;

$C(q_{t,m}) = \alpha \cdot \bar{b}_{t-1,m} + (1 - \alpha)q_{t,m}$;

End

Procedure BufferControl

```

Begin
   $\rho_i = 300\text{ms};$ 
  If buffer_point == NormalLevel
     $\tau_t = \rho_i;$  Send( $MH_i, GOP_k^i, \tau_t$ );
    Send( $S_k, \text{Request}, GOP_k^i$ );
  Else If buffer_point == UpperLevel
    Begin
       $BL_{allsize} = BL_{allsize} + 1;$ 
       $BL_{psize} = \text{CurrentBuffer};$ 
       $\omega = 0;$  ;  $\rho_i = 125;$ 
       $\omega = BL_{allsize} / BL_{psize};$  ;  $\tau_t = \rho_i - (\rho_i \times \omega);$ 
      Send( $MH_i, GOP_k^i, \tau_t$ );

      Send( $S_k, \text{UpperLevel}, GOP(I, P, B)_k^i$ );
    End Else If
  Else IF buffer_point == LowerLevel
    Begin
       $\omega = BL_{allsize} - BL_{psize};$  ;  $\omega = \omega / BL_{allsize};$ 
       $\tau_t = \rho_i + \lambda \times \omega;$ 
      Send( $MH_i, GOP_k^i, \tau_t$ );
      Send( $S_k, \text{LowerLevel}, GOP(I)_k^i$ );

    End
  End If
End

```

```

Begin /* Main Program */
Call Start-UpTime;
Start-Up = MAX  $D_k^i - D_k^i$ ;
Set Start-Up Time for Server  $S_k$ ;
Send( $S_k, \text{Start-Up Time}$ );
 $\Delta = DT_k^{\max} - DT_k^{\text{mix}}$  /*  $\Delta$  is jitter */
If Buffer_point == NomalLevel Then
  Send ( $S_k, , \text{Request}; GOP_k^i$ )
Else If Buffer_point == UpperLevel Then
  Begin
    If  $DT_k^i > \Delta$  Then
      Begin
        Feedback_value = Call Feedback;

```

```

      Send ( $S_k, \text{Feedback\_value}(GOP(I, P), D_k^i)$ );
    End
  Else
    Continue;
    Call BufferControl(UpperLevel);
  End Else IF
  Else If Buffer_point == LowerLevel Then
    Begin
      If  $DT_k^i > \Delta$  Then
        Begin
          Feedback_value = Call Feedback;
          Send ( $S_k, \text{Feedback\_value}(GOP(I), D_k^i)$ );
        End
      Else
        Continue;
        Call BufferControl(LowerLevel);
      End
    End If
  End IF
End IF

Receive( $BS_n, \text{Handoff}_{on}$ )
 $BS_{old} = BS_{Current};$ 
 $BS_{Current} = BS_{new};$ 
Send( $MH_i, \text{Handoff}_{on}$ );
Send( $S_k, \text{Handoff}_{on}$ );
For k=1, k≤ K, k+2 Do
   $BS_{new}(GOP(I)_k^i) = BS_{old}(GOP(I)_k^i);$ 
  Send( $S_k, \text{Handoff}_{on}$ ); /*from  $BS_{new}$  to  $S_k$  */
Send( $S_k, \text{Request}; GOP_k^i$ );
End /* Main */

```

4. Conclusions

본 논문은 버퍼관리 및 재생 정책을 베이스 스테이션에서 관리함으로써 핸드오프에 대한 신속한 대처를 할 수 있는 방안을 제시하였으며 이동통신에 대한 변수 즉 적은 메모리, 낮은 대역폭 등의 변수에 적절하게 대응할 수 있는 방안을 제시하였다. 또한 제안한 프레임에 대한 조절된 디스플레이 재생시간 기법은 부드럽고 자연스러운 성능을 나타내었고, 전체 재생 시간은 원래의 재생시간을 벗어나지 않았다.

참고문헌

- [1] D. K. K. Y. Yau, S. S. Lam, "Adaptive Rate Controlled Scheduling for Multimedia Applications", *Multimedia 96 Processing*, 4th ACM, Boston. Ma. pp. 129-140. 1996
- [2] M. Woo, N. U. Qazi, and A. Ghafoor, "A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information," *IEEE Network*, Jan./Feb. 1994.
- [3] 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.
- [4] W. Geyer, "Stream Synchronization in a Scalable Video Server Array," Master's thesis, Institute Eurecom, Sophia Antipolis, France, Sept., 1995.
- [5] 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.