

무선 환경에서 병렬 서버를 위한 멀티미디어 전송 및 동기화 기법

준회원 신 광 식, 백 석 균, 윤 완 오, 정 진 하, 정회원 최 상 방*

Multimedia Transmission and Synchronization Scheme for Parallel Servers in Wireless Environment

Kwang-sik Shin, Seok-Kyun Baek, Wan-oh Yoon, Jin-ha Cheong, *Associate Members*
Sang-bang Choi* *Regular Members*

요 약

본 논문은 무선 환경에서 병렬서버를 이용한 멀티미디어 전송 및 동기화 기법을 제안한다. 제안된 방법은 MPEG-4 비디오 스트림을 VOP 타입(I-, P-, B-VOP)에 따라 서로 다른 서버에 저장하여 서비스하는 병렬 서버 시스템을 위하여 개발되었다. 이러한 환경의 멀티미디어 전송에 있어서는 특히 부하 불균형, 정체현상, 지터문제를 해결하여야 한다. 따라서 본 연구는 데이터 분할방법, 우선순위 큐, 멀티미디어 동기화 문제에 초점을 맞춘다. 첫째, VOP 타입에 따라 데이터를 분할하여 개별의 서버에 나누어 저장하는 병렬서버 구조를 개발한다. 둘째, 병렬 서버 환경에서의 우선순위 알고리즘을 설계한다. 기지국에서는 이 알고리즘의 우선순위에 따라 스케줄링을 수행한다. 마지막으로, 지터를 보상하기 위한 동기화 알고리즘을 설계한다. 기지국은 각 서버로부터 불규칙하게 들어오는 비디오 스트림을 고안된 동기화 알고리즘에 따라 버퍼에 저장한 후, 이동국으로 일정하게 전송한다. 시뮬레이션 분석 결과 기지국이 혼잡한 상황에서 제안한 병렬 서버 구조에 기반을 둔 멀티미디어 스트림 전송 방식이 기존의 방식보다 이동국의 수신 상태가 57% 향상됐음을 알 수 있다.

Key Words : Multimedia stream, data striping, priority queue, synchronization, MPEG-4

ABSTRACT

This paper presents new multimedia transmission and synchronization scheme for parallel server in wireless environment. The proposed scheme is developed for the parallel server that stores MPEG-4 data according to VOP (video object plane) types, i.e., I-, P-, and B-VOP. The goal of our research is to develop the enhanced multimedia transmission scheme that copes with load unbalancing, congestion, and jitter. Thus, we consider three issues; the data striping, the priority queuing, and the multimedia synchronization. First, we develop a parallel server architecture in which each server separately stores video data that is striped by the MPEG-4 VOP types.

I. 서 론

최근 인터넷 접속기술과 멀티미디어 기술의 급속

한 발전은 VoD와 같은 온라인 멀티미디어 서비스를 가능하게 하였다. 그런데 이러한 VoD 서비스를 일반적인 데이터 전송과 비교할 때 많은 디스크 저

* 인하대학교 전자공학과 컴퓨터구조 및 네트워크 연구실(sangbang@inha.ac.kr)

논문번호 : KICS2005-02-079, 접수일자 : 2005년 2월 22일

※ 본 논문은 정보통신부의 정보통신연구진흥원에서 지원하고 있는 정보통신 기초기술연구지원사업의 연구결과임.

장용량과 대역폭을 요구한다. 이러한 차이점이 있음에도 불구하고 지금까지의 VoD 서비스는 대부분 기존의 텍스트 기반 서비스와 같은 단일 서버 모델을 통해 서비스되어 왔다.

이러한 서비스는 몇 가지 문제점을 안고 있다. 첫 번째 문제는 바로 서버의 서비스 가능 용량과 관련이 있다. 클라이언트로부터의 요청이 서버 측에서 서비스할 수 있는 대역폭을 초과할 경우 서버는 일부 클라이언트들의 요청을 거부해야만 한다. 따라서 동일한 내용을 저장해 놓은 미리 서버들을 사용하여 이들의 요구를 수용하게 된다. 그렇지만 이 경우 똑같은 비디오를 다른 서버에 저장해야 하기 때문에 데이터 저장측면에서 오버헤드가 크다. 또한 비디오 서버 간의 부하의 불균형적인 분배 문제가 발생한다. 이런 오버헤드와 부하 불균형 문제를 해결하기 위해서 콘텐츠 복제와 관련된 다양한 연구가 진행되고 있다.

두 번째 문제는 오류 허용 능력과 관련이 있다. 만약 비디오를 한 서버에만 저장할 경우 그 서버에 치명적인 오류가 생겼을 때, 해당 비디오에 대한 모든 요청은 서비스가 불가능하다. 물론 미리 서버를 두면 오류에 대한 허용이 가능하지만 동일한 데이터에 대한 중복 저장으로 인한 디스크 오버헤드가 발생한다. 이러한 복제에 관련된 문제들을 해결하기 위한 방식 중에 하나로 데이터 분할 기법이 제안되었다. 이러한 기법을 이용하여 멀티미디어 데이터는 몇 개의 하위 데이터 구조로 나누어져 각각의 병렬 서버에 분산 저장된다[1]. 병렬 서버 구조를 위한 데이터 분할 기법은 디스크 저장 용량과 시스템의 신뢰도를 높여준다.

본 논문에서는 MPEG-4 데이터를 I-, P-, B-VOP로 나누어 각각의 서버에 저장하는 새로운 병렬 서버 구조를 제안한다. 또한 이를 위해 무선 환경에서의 병렬 서버 구조를 위한 멀티미디어 데이터 전송과 동기화에 관련된 기법을 제안한다. 본 연구에서 각각의 서버는 VOP 타입에 따라 비디오 데이터를 분리하여 해당 VOP만을 저장한다. 이동국이 멀티미디어 서비스를 요청할 경우, 각각의 서버들은 각 VOP를 미리 정해진 VOP의 순서에 맞춰 전송한다. 무선 네트워크 환경에서, 각 서버로부터 전송된 모든 패킷은 이동국에 도착하기 위해 반드시 하나의 기지국을 통과해야 한다. 우리는 이 기지국에서 각 서버로 분산된 VOP를 원래 순서에 맞게 하나의 스트림으로 재구성한다. 이때 패킷 혼잡이 발생하면 손실을 줄이기 위해 우선순위에 따라 스케줄링 한

다. QoS를 제공하기 위해서 I-VOP에 가장 높은 우선순위를 적용하고, P-VOP는 두 번째, B-VOP는 가장 낮은 우선순위를 적용한다. 왜냐하면 한 GOP에서 I-VOP가 손실되거나 손상될 경우 그 GOP를 구성하는 모든 VOP들이 손상되기 때문이다. 또한 각 서버 스트림이 전부 다른 서버로부터 전송되기 때문에, VOP 타입들은 다른 지연 특성을 보인다. 그러므로 이러한 지터를 보상하는 동기화 기법이 필요하다.

우리는 실제 비디오 파일을 이용한 시뮬레이션을 통하여, 기존의 병렬서버 모델과 제안된 병렬서버 모델을 비교하였다. 본 논문에서의 비교하는 기존의 병렬서버 모델은 데이터를 저장하는데 있어서 타입에 관계없이 똑같은 사이즈로 나누어 각 서버에 저장하는 병렬 서버구조로 정의한다.

본 논문은 다음과 같이 구성된다. 2절에서는 MPEG-4 비디오 데이터의 구조와 특징을 설명한다. 3절에서는 기존의 병렬 서버에서 데이터 분할 방법에 대해 설명하고, 제안한 병렬 서버 구조와 데이터 분할 기법을 설명한다. 4절에서는 VOP 타입을 고려한 세 가지 부스트림의 우선순위 큐 관리 정책을 제안한다. 또한 기지국에서의 스트림 재조합 과정을 논의한다. 5절에서는 멀티미디어 동기화 알고리즘을 제안하고, 이어서 6절에서는 시뮬레이션 결과를 보이고 분석한다. 마지막으로 7절에서는 본 논문의 결론을 제시한다.

II. MPEG-4 비디오 데이터 구조

제안된 병렬 서버 구조에서 사용하는 데이터는 MPEG-4 표준을 기초로 한다. MPEG-4는 객체 지향적인 압축 표준이며 무선 채널에서 문제시되는 높은 비트 당 에러율에도 불구하고 강한 오류 복구 능력을 지니고 있다. 또한 기존의 자연 영상뿐만 아니라 컴퓨터 그래픽이나 합성 영상 또한 압축할 수 있으며 수정, 편집 기능 또한 지니고 있기 때문에 무선 멀티미디어 압축 표준으로 널리 사용될 것으로 기대된다.

MPEG-4는 기존의 MPEG-1, MPEG-2에 비해 여러 가지 부가적인 기능이 지원되는 것은 사실이지만 자연 영상을 무선 채널에 보낸다고 가정하면 압축 데이터를 복원하는 알고리즘 및 비디오 데이터의 구성은 MPEG-4와 MPEG-1이나 MPEG-2와 비교할 때 크게 다르지 않다. 이 경우 MPEG-4의 비디오 데이터는 VOP 단위로 구분되며 I-, P-, B-VOP로

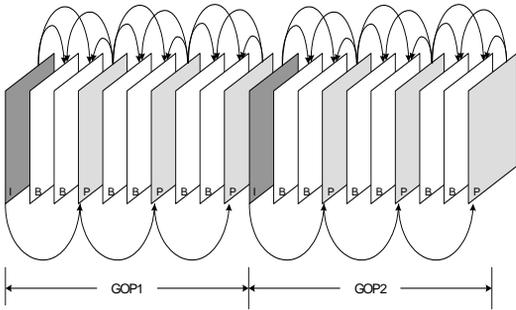


그림 1. MPEG-4의 GOP 구조

이루어진다. 이들 중 I-VOP가 손실되면 해당 GOP 내의 모든 VOP가 손실된다. 따라서 MPEG-4에서 비디오 데이터는 VOP 간의 중요도가 달라진다. 인트라 코딩을 하는 I-VOP는 오직 해당 VOP 내에 포함된 정보만을 이용한 다양한 압축 코딩 기법을 사용한다. 즉, 다른 VOP간의 시간적인 정보변화에 따른 코딩 방식이 아니라 현재 VOP 내에 있는 공간적인 정보변화에 국한된 코딩방식을 이용한다 [3-4].

그림 1은 MPEG-4 VOP의 구조를 보여준다. 그림에서 화살표는 각 VOP가 디코딩 할 때 참조하는 VOP의 방향을 나타내고 있다. 그러나, 만약 시간에 따른 정보변화를 잘 활용한다면, 상당히 많은 압축 효율을 얻을 수 있다. 예를 들어 대부분 시퀀스 내의 연속적인 VOP 집합은 특정 VOP를 기준으로 볼 때 전후의 VOP와 매우 비슷하다는 것을 알 수 있다. 이러한 특징을 이용하여 P-, B-VOP를 만드는 데 이를 인터 코딩기법이라 한다. GOP는 I-VOP로 시작하기 때문에 인코더는 다음 VOP를 예상해서 진행할 수 있다. 일반적으로 이것은 시간적으로 볼 때 순방향적인 참조(과거의 VOP를 참조)를 사용하는 P-VOP에 의해 참조된다. 양방향적인 참조를 갖는 VOP(B-VOP)는 과거와 미래의 VOP를 참고한다. 높은 압축을 하기위해서, 동작 보상은 가장 가까운 과거 또는 미래의 P- 또는 I-VOP를 기초로 수행 될 수 있다. B-VOP는 참조되지 않는다. 이미 인코딩 된 가장 가까운 I- 또는 P-VOP에 기초하는 동작 보상 예측을 사용하여 B-VOP를 인코딩 할 수 있다.

만약 I-VOP가 손실 되거나 손상됐을 경우, 이 I-VOP가 속한 GOP 내에 있는 모든 VOP들은 손상된다. 그러므로 QoS를 향상시키기위해 I-VOP에 가장 높은 우선순위를 주고, 기지국에서 이런 우선 순위에 따라 스케줄링 한다.

III. 병렬 서버 구조와 데이터 분할 기법

데이터 분할 기법은 용량성과 잠재적 신뢰도를 향상시키기 위해 여러 서버에 데이터를 분산 저장하는 일반적인 방법이다. 데이터 분할 기법은 크게 시간 분할(time striping)과 공간 분할(space striping) 두 가지로 나눌 수 있다. 비디오 스트림은 비디오 프레임의 연속으로 볼 수 있다. 비디오 스트림이 여러 서버에 프레임 단위로 분할 저장되어 있을 때, 시간 분할 방식은 라운드 로빈 방식을 사용하는 반면에, 공간 분할 방식은 비디오 스트림을 일정한 크기의 분할된 유닛으로 나누고 그것을 각 서버에 저장한다.

데이터 분할을 사용하는 병렬 서버 구조는 여러 서버가 복제된 데이터를 똑같이 저장하는 병렬 서버 구조에 비해 몇 가지 장점을 갖는다. 첫째, 데이터 분할을 사용하는 병렬 서버 구조는 분할된 데이터를 여러 서버에 나누어 저장하기 때문에 디스크 오버헤드가 적다. 둘째, 각각의 분할 데이터를 저장한 병렬 서버에, 각 서버의 데이터 검색률을 서로 갖게 조절 할 수 있기 때문에 부하 불균형 문제를 감소시킬 수 있다.

마지막으로, 특정 서버 고장으로 인한 서비스 중단을 막을 수 있다. 왜냐하면 각 서버는 특정 서버에 오류가 발생할 경우 이를 복원하기위해 데이터 복구를 위한 여분의 데이터를 가지고 있다면, 해당 서버를 제외한 나머지 서버는 멀티미디어 서비스를 계속해서 제공할 수 있기 때문이다.

본 논문에서, 비디오 스트림을 제공하기 위한 시간 분할 기법 중 비디오 스트림을 I-, P-, B- VOP으로 분할하는 방식을 제안한다. 각 부 스트림이 병렬 서버에 저장되기 전 인코딩 시간에 비디오 스트림의 분할이 이루어진다. 각 VOP를 세 가지 파일로 분류하고 저장하는 인코더에 VOP 분할 모듈을 구현했다.

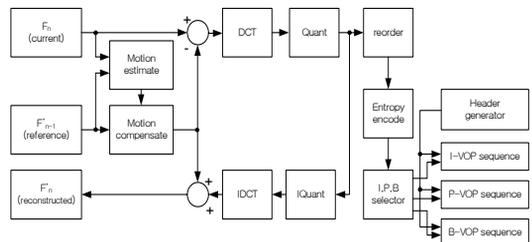


그림 2. 각각의 VOP를 세 개의 프레임 시퀀스로 분류하고 저장하는 새로운 MPEG 인코더의 구조

표 1 MPEG-4 비디오의 VOP 타입에 대한 정보 : Starwars-IV

	VOP의 총 개수	VOP의 총 크기	VOP의 최대 크기(bytes)	VOP의 최소 크기(bytes)	VOP의 평균크기(bytes)
I-VOP	2250	3557508	4299	372	1581.12
P-VOP	6750	2356986	3542	39	349.18
B-VOP	17998	2106544	2031	26	117.04

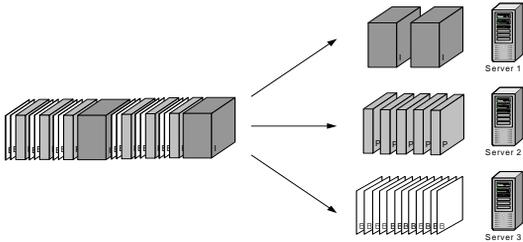


그림 3. VOP 타입에 따라 비디오 스트림을 나누는 제안된 비디오 분할 방식의 개념적인 모습

그림 2는 새로운 MPEG 인코더의 구조를 보여준다. 인코딩할 때 기지국에서 세 가지의 부 스트림을 합치기 위해 시퀀스 넘버, 개개의 GOP 넘버와 함께 MPEG-4 표준에는 없는 추가적인 VOP 넘버를 VOP 헤더에 포함시킨다.

그림 3은 비디오 스트림을 나누고 나누어진 각각의 부 스트림을 VOP 시퀀스에 맞게 저장하는 제안된 비디오 분할 방식의 개념적인 모습을 보여준다. 각 VOP 시퀀스는 단 하나의 VOP 타입으로 구성되고 서로 다른 서버에 저장된다. 표 1에서 볼 수 있듯이 각 VOP 타입에 따라 VOP의 수와 크기가 크게 다른 것을 볼 수 있다. 그러나 각 VOP 타입의 총 크기에는 큰 차이가 없기 때문에 병렬 서버들은 일정 수준의 부하 균형을 이룬다.

IV. 우선순위 패킷 큐와 기지국에서의 재조합 과정

QoS를 보장하기 위해, 기지국과 라우터에서는 VOP 타입에 따라 VOP를 구별해야 한다. 왜냐하면 라우터는 IP 계층에서 작업하고, IP 헤더는 I-, P-, B-VOP를 처리해야 하기 때문이다. 그림 4에서는 VOP 타입에 따라서 패킷에 우선순위를 주기 위해,

4 bits version	4 bits header length	8 bits type of service (TOS)	16 bits total length	
16 bits identification		3 bits flag	13 bits fragment offset	
⋮				

그림 4. TOS bit을 사용하는 새로운 IP 헤더

라우터는 IP 헤더에 있으면서 현재 사용되지 않고 있는 TOS 필드의 처음 3-bit를 사용한다. 중요한 부 스트림에 대한 전송 지연과 패킷 손실은 패킷에 높은 우선순위를 적용하여 라우터에서 더 많은 서비스를 받게 함으로써 감소시킬 수 있다[5-6].

제안된 기법에서, 재 정렬과 병합 절차는 기지국에서 이루어진다. 그림 5는 수신, 병합, 재 정렬 절차를 보여준다. 수신 버퍼는 I-, P-, B-VOP 버퍼로 이루어진다. 각 입력 버퍼로 들어온 패킷들은 스케줄러에 의해 모아졌다가 송신 버퍼로 보내진다. 송신 버퍼는 지터 보상에 대한 책임을 진다. 송신 버퍼는 지터를 보상하기 위한 초기 지연 시간을 설정한다. 이것은 클라이언트에서의 재생 버퍼와 같은 역할을 한다[7]. 송신 버퍼에서의 지연 시간은 수신 버퍼로부터 들어온 첫 번째 VOP에 의해 설정된다. 이것에 대해서는 다음 절에서 자세하게 설명될 것이다.

각 서버로부터 전송된 패킷은 수신 버퍼에 순차적으로 쌓이게 된다. 그런 다음, 그 패킷들은 스케줄러에 의해 송신 버퍼로 옮겨진다. 그림 6은 스케줄러의 작업을 나타낸다. 버퍼가 꽉 찼을 경우, 더

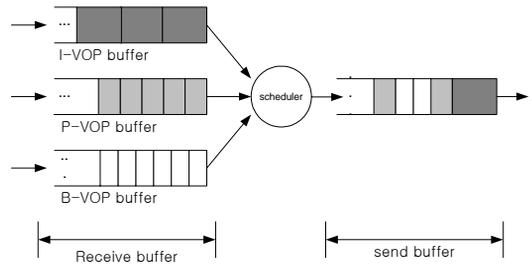


그림 5. 기지국에서의 패킷 병합과 스케줄링 프로세스

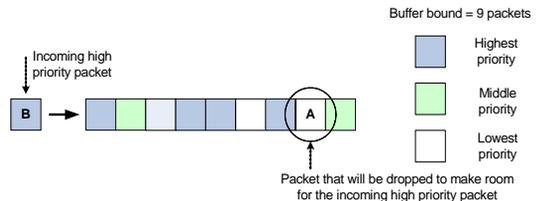


그림 6. 버퍼가 꽉 찼을 경우의 스케줄링 프로세스

높은 우선순위를 가지고 들어오는 패킷(B)을 위한 공간을 마련하기 위해 버퍼에 있는 패킷 중 가장 낮은 우선순위를 갖는 패킷(A)이 버려진다.

본 논문에서 기지국은 I-, P-, B-VOP를 받아서 GOP를 복구한다. 만약, I-VOP의 부분이 손실된다면, 이 I-VOP를 포함하는 GOP 내의 모든 VOP가 손상된다. P-VOP 중 일부가 손실된다면, 이전의 B-VOP, 다음의 B-VOP 그리고 손실된 P-VOP가 포함된 GOP 내의 다른 P-VOP 등, 손실된 P-VOP와 관련된 VOP들의 영역으로 손실의 영향이 전파된다. 그렇지만 B-VOP 중 일부의 손실은 다른 VOP에 영향을 미치지 않는다[8]. 한편, 멀티미디어 서비스는 지연에 민감하기 때문에, 모든 VOP는 그것의 데드라인 전에 전송되어야만 한다. 만약 기지국이 데드라인 전에 I- 또는 P-VOP를 받지 못한다면, 같은 GOP 내의 관련된 VOP는 버려진다[9].

본 논문에서는, 기지국에서 이루어지는 우선순위 큐 관리에 대한 정책을 제안한다. 그림 7은 우선순위 큐 관리 정책에 대한 블록 다이어그램을 보여준다. 첫 번째로, 기지국의 큐 관리자가 어떤 타입의 VOP를 전송받았는지 확인한다. 만약 I-, P-, 또는 B-VOP를 받았을 때 그것이 임계시간이 지났다면, 그 VOP를 폐기하게 된다. 그렇지 않고 임계시간 전에 받았다면, 큐가 가득 찼는지 확인한다. 이때 큐가 가득 차있으면, B-VOP를 제거하고 I- 또는 P-VOP를 저장한다. I-VOP의 경우, 큐에 B-VOP가

없다면, P-VOP를 버리고 I-VOP를 저장한다. 각 서버에 나뉘어 저장된 멀티미디어 데이터는 기지국에서 원래 순서에 맞게 재조합된다. 이 과정에서 MPEG-4 표준을 만족시키기 위해, 시퀀스가 재조합되기 전에 VOP 헤더에서 병렬 서버의 데이터 분할을 위해 추가했던 VOP 넘버를 제거한다.

성능 분석을 위해, 기지국의 우선순위 큐잉을 시뮬레이션하고, 확률 모델을 사용하여 패킷의 손실량을 측정한다. 그림 8과 그림 9는 기존 방식과 제안된 방식에서 각각의 초당 I-, P-, B-VOP의 손실량을 보여준다. 그림 10은 I-, P-, B-VOP의 총 손실량을 보여준다. 앞서 언급했듯이, 제안된 방식은 기존의 방식에 비교해서 더 적은 패킷 손실량을 기록했다. 특히, I-VOP의 손실이 현저히 감소된 것을 확인할 수 있다. 때때로 동일한 GOP내에서 하나 이상의 VOP 타입이 손실되는 것이 가능하다. 그러므로, 표 2에서 볼 수 있듯이 모든 VOP의 평균 손실은 각 VOP의 손실을 합한 것과 정확하게 일치하지 않는다. 제안된 방식이 기존의 방식에 비해 VOP 손실을 평균 61.7%까지 감소시킨 것을 알 수 있다.

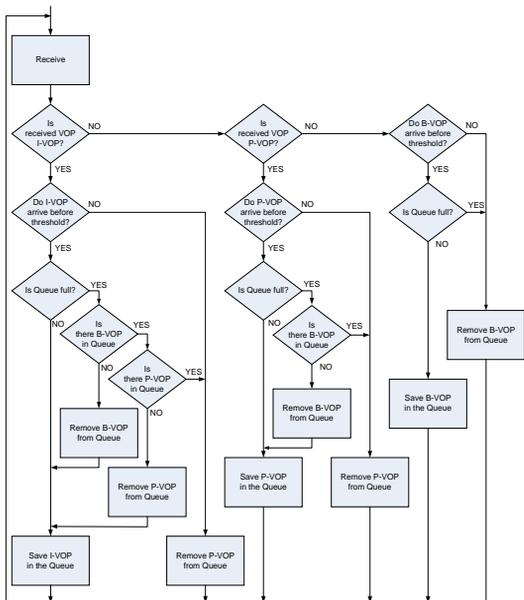


그림 7. 우선순위 큐 관리 정책의 블록 다이어그램

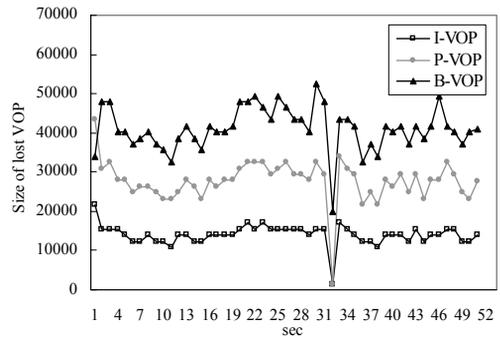


그림 8. 기존 방식에서의 각 VOP 타입의 초당 손실량

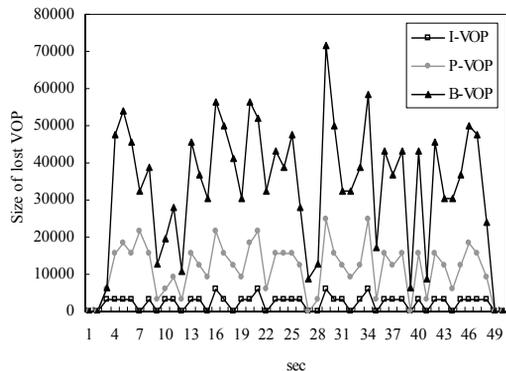


그림 9. 제안된 방식에서의 각 VOP 타입의 초당 손실량

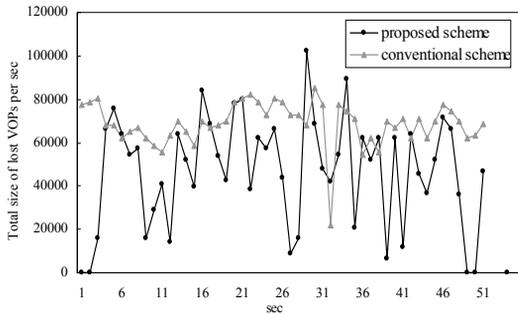


그림 10. 기존 방식과 제안된 방식에서 세 가지 VOP 타입의 총 손실량

표 2. 기존 방식과 제안된 방식에서의 VOP의 평균 손실

	기존 방식	제안된 방식
I-VOP의 평균 손실	13799	2289
P-VOP의 평균 손실	27227	11447
B-VOP의 평균 손실	41057	23638
계	75895	46829

V. 멀티미디어 동기화

클라이언트로부터 요청된 멀티미디어 콘텐츠는 오디오, 비디오, 텍스트 또는 이미지와 같은 다양한 타입의 미디어로 이루어져 있다. 이러한 스트림의 요소 사이에는 일시적인 관계가 존재한다. 병렬 서버 환경에서 멀티미디어 시스템은 데이터의 저장, 전송, 재생 시 이 관계를 유지해야 한다. 일반적으로, 하나 또는 여러 미디어 스트림의 시간적 순서를 유지하는 과정을 멀티미디어 동기화라고 한다[10].

본 논문에서는 멀티미디어 동기화를 유지하기 위해 전송률 제어 기법과 지연 버퍼를 사용한다. 전송률 제어 기법은 서버의 송신률을 결정한다. 그리고 지터를 보상하기 위해 지연버퍼를 사용한다. 이처럼 지터를 보상해줌으로써 인트라-스트림 동기화를 유지할 수 있다. 그림 11에서 볼 수 있듯이, 지터를 보상하기위해 버퍼를 이용하여 멀티미디어 스트림에 추가적인 지연을 만들어낸다[11].

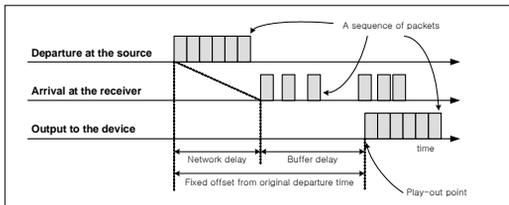


그림 11. 지터 보상을 위한 지연 버퍼

5.1 멀티미디어 서비스를 위한 초기화 절차

이동국에 여러 개의 부 스트림으로 구성된 멀티미디어 스트림을 제공하기 위해서는 초기화 절차가 필요하다. 각 서버로부터 받은 VOP는 서로 다른 지연 특성을 갖는다. 초기화 과정은 다음과 같이 동작하는 초기 인터-프레임 동기화 알고리즘을 포함한다. 먼저 이동국이 멀티미디어 서비스를 요청할 때 주고받은 메시지로부터 얻은 정보를 이용하여 각 서버의 지연을 측정한다. 그 다음 각 부-스트림의 시작 시간을 계산하고, 각 서버들에 대한 멀티미디어 서비스 요청 시간에 따라 타이머를 적절히 설정한다. 마지막으로, 타이머가 종료되면 Media_request 메시지를 생성하고 전송한다. 그림 12는 초기화 절차를 위해서 메시지를 교환하는 과정을 보여준다.

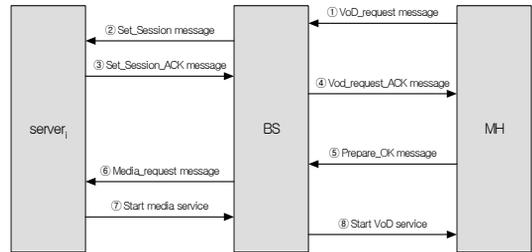


그림 12. 인터-프레임 동기화를 위한 멀티미디어 서비스의 초기 메시지 흐름

표 3. 알고리즘을 위한 기호

Symbol	Meaning
BS	Base Station
Server _i	Server with index i
tr _i	Time stamp for returning packet from Server _i
st _i	Media request time of Server _i
d _{max}	Maximum RTT from the BS to Server _i
SCR	System clock reference
δ	Maximum st _i
MH	Mobile Host
ts	First time stamp
△	Value of jitter bound
bi	Requested value of a buffer slot for Server _i
ds _i	RTT form the BS to Server _i
B _i	Buffer size for Server _i
r _i	Service rate of Server _i
BLT	Lower threshold of buffer level
BUT	Upper threshold of buffer level
B _{max}	Maximum size of available memory
MMUK	Multimedia unit generated from Server _i

```

In base station
Receive VoD_request message from the BS
while ( Serveri )
    ts ← Current time ; // stamping local time of the BS
    Send Set_Session Message to the server ;
Receive Set_Session_ACK message from each server
while ( receive Set_Session_ACK message from Serveri )
    Bi ← [ 2 · Δ · ri ] · bi // allocating buffer for the server
    tri ← Current time; // stamping local time of BS
    dsi ← tri - ts; // calculating RTT from the BS to the server
    dmax ← max { dsi | for all i } ; // calculating maximum
RTT
while ( Serveri )
    sti ← dmax - dsi ; // calculating Media_request time of
Serveri
    δ ← max { sti | for all i } ; //calculating maximum sti
Send VoD_request_ACK message to the MH ;
Receive Prepare_OK message from the MH
Set Sending timer for Serveri ;
Set SCR ← Current time; // setting of system clock
reference
while ( Serveri )
    if ( Sending timer expired for the server )
        Stamping SCR on the Media_request message packet
;
        Send Media_request message for the server ;
        Set SCR ← SCR - dmax + δ - [ 2 · Δ · ri ] ; // reset of
SCR
Receive MMU from Serveri
Start VoD_Service Procedure ;

In server
Receive Set_Session message from the BS
Write bi, ts and ri on Set_Session_ACK message packet;
Send Set_Session_ACK message to the BS;
Receive Media_request message from the BS
Read SCR(1) from the Media_request message;
Set SCR(1) for the stream i;
Start Media_service to the BS;

In mobile host
Send VoD_request message to the BS;
Receive VoD_request_ACK message from the BS
Allocate buffer for the VoD;
Send Prepare_OK message to the BS;
    
```

그림 13. 멀티미디어 서비스를 초기화하기 위한 절차

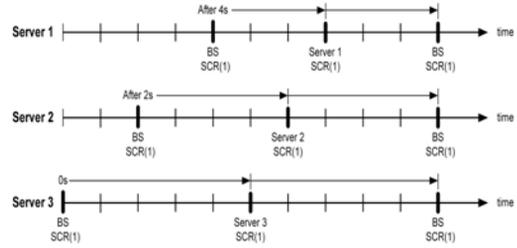


그림 14. 시작 시점 계산의 예

본 논문에서 제안한 알고리즘에서는 각 서버의 클럭 간의 동기를 맞추기 위하여 글로벌 클럭이나, NTP(Network Time Protocol) 같은 복잡한 프로토콜은 사용하지 않는다. 그 대신 부 스트림간의 동기를 유지하기 위하여 타임스탬프와 클럭 드리프트 문제를 해결하기 위한 재동기화 절차를 사용한다. 그림 13은 제안된 알고리즘의 슈도코드를 보여주고, 표 3은 알고리즘에서 사용하는 기호를 설명한다.

세 개의 서버, S1, S2, S3가 있고, 서버로부터의 각 경로는 3, 4, 5초의 일정한 지연을 갖는다고 가정한다. 그러면 기지국으로부터 각 서버까지의 RTT는 각각 6, 8, 10초이다. S1과 S2로부터 전송 받은 패킷들을 위해서 기지국에서 불필요한 버퍼 공간을 없애기 위해, 모든 서버의 미디어 서비스 시작 시간이 링크 중 가장 큰 지연 시간으로 고정된다. 즉, S3에서의 지연 시간을 기준으로 Media_request 메시지의 전송 시간은 S1은 4초, S2는 2초, S3은 0초로 정해진다. 이동국으로부터 Prepared_OK 메시지를 받은 후에, 기지국은 즉시 S3에게 Media_request 메시지를 전송하고 S2와 S1에게는 각각 2초와 4초 후에 전송하게 된다.

MPEG의 시스템 표준은 두 시스템 클럭의 차이를 동기화하기 위해 타임 스탬프, 즉 SCR(System Clock Reference)를 사용한다[12]. SCR(System Clock Reference)의 시작 값이 기지국에서 같아야 하기 때문에, 기지국은 SCR의 최대 지연을 고려해서 SCR을 10으로 재설정한다. 그 예는 그림 14에서 볼 수 있다.

5.2 버퍼 관리 절차

멀티미디어 서비스 초기화 절차를 수행한 후, 이동국은 기지국으로부터 멀티미디어 스트림을 전송받기 시작한다. 이전 절에서 언급한 것처럼 멀티미디어 부 스트림은 기지국에서 재조합된다. 그러므로, 이동국으로 끊임 없는 멀티미디어 스트림을 제공하고, 기지국의 오버헤드를 줄이기 위해 복잡한 버퍼

제어 기법이 필요하다[13]. 제안된 알고리즘은 버퍼의 오버플로우와 언더플로우를 감지하기 전에 버퍼의 임계레벨(BUT, BLT)에 따라 동기화 절차를 수행한다. 버퍼 레벨이 하한 또는 상한 임계레벨에 도달하면, 동기화 절차를 시작하고 서버에게 전송률 제어 메시지를 보낸다. 서버는 이 메시지를 받은 후, 기지국의 요청에 따라 패킷 생성률을 조절한다. 그림 15는 제안된 동기화 알고리즘의 슈도 코드를 보여준다.

그림 16은 기존 병렬 서버 모델과 제안된 병렬 서버 모델 환경에서 각 기지국이 할당한 버퍼크기의 변화를 보여준다. 제안한 기법은 동기화와 버퍼 관리 절차를 수행하기 때문에, 임계범위를 벗어나는 구간을 줄일 수 있다. 그림에서 제안된 기법은 16초와 85초에 각각 버퍼 오버플로우와 언더플로우를 막기 위해 버퍼 제어 절차를 수행한 것을 볼 수 있다. 반면에 기존 기법에서는 16초에 버퍼에서 오버플로우가 발생하기 때문에 일부 패킷이 버려져야 한다는 것을 알 수 있다. 또한 시간 86초와 91초에는 버퍼가 비어있고, 따라서 이동국에 계속해서 멀티미디어 스트림을 제공할 수 없기 때문에 일시적인 서비스 중단을 예측 할 수 있다.

표 4는 동기화 절차를 수행하기 위한 시뮬레이션의 파라미터를 보여주고, 표 5는 시뮬레이션 결과를 보여준다. 시뮬레이션 결과를 통하여 제안한 기법이 버퍼 오버플로우나 언더플로우 없이 평균 버퍼 사용을 7.4% 감소시킨 것을 볼 수 있다.

표 4. 동기화 절차의 시뮬레이션에서 사용된 파라미터

	Conventional scheme	Proposed scheme
Buffer Size	64 Kbytes	64 Kbytes
Upper Threshold	N/A	85%
Under Threshold	N/A	30%

표 5. 동기화 절차의 시뮬레이션 결과

	Conventional scheme	Proposed scheme
Buffer Overflow	19.4%	0
Buffer Underflow	5.6%	0
Average Buffer Usage	73.4%	66%

```

if ( BLi > BUT ) // if buffer level reaches the upper threshold
    flagtmp ← flag; // current status stores in flagtmp
    
```

```

Set flag ← overflow ;
Send Higher_rate message to the serveri ;
else if ( BLi < BLT ) // if buffer level reaches the lower threshold
    flagtmp ← flag; // current status stores in flagtmp
    Set flag ← underflow;
    Send Lower_rate message to the server;

DetectChange_of_flag in the BS
if ( flag == overflow )
    while ( flag ) // while flag is overflow
        if ( Bmax < Bi + sizeof(MMUk) )
            /*
            If there are enough memory space for allocating additional buffer in the base station, then allocate additional buffer
            */
            Bi ← Bi + sizeof(MMUk) ;
            count ++ ;
        else
            Discard MMUk; incoming packet is discarded
            Start re_sync phase ;
    else if ( flag == underflow ) // setting the timer for calculation of re-sync time threshold
        Set timer;
        while ( flag ) // while flag is underflow
            if ( timer is expired ) // if the timer is expired,
                Start re_sync phase; // it is the changing of network condition
    else if ( flag == normal )
        Send Normal_rate message for the server;
        if ( flagtmp == overflow )
            /*
            past status of flow is overflow calculation of allocated additional buffer size reallocate buffer for current stream
            */
            Bi' ← sizeof(MMUk) ← count;
            Bi ← Bi - Bi';
            free ( Bi' );
    
```

그림 15. 기지국을 위한 버퍼 관리 절차

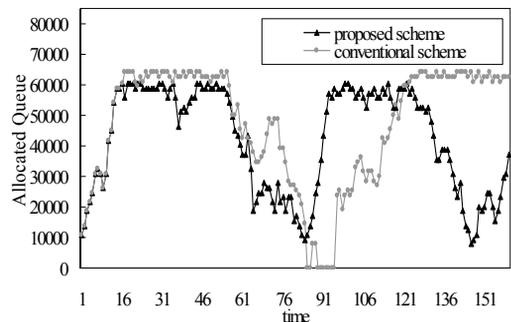


그림 16. 두 병렬 서버 모델의 버퍼 크기 비교

VI. 시뮬레이션 결과 및 분석

이 절에서, 스타워즈-IV 비디오 트레이스 파일과 두 개의 MPEG-4 비디오 소스(Foreman, Football)를 사용하여 분석과 시뮬레이션을 하였다. 스타워드-IV 트레이스 파일을 이용하여 분석결과를 설명하고 그리고 이절의 마지막에서 두 비디오 소스로부터 최종결과를 시각적으로 보여준다.

그림 17은 'Starwars-IV' 비디오 소스에서 각 프레임의 VOP 크기를 나타낸다. 네트워크에서, 오직 한 기지국만 병렬 서버로부터 비디오 스트림을 받는다고 가정한다. 표 6은 시뮬레이션에서 'Starwars-IV' 비디오 소스의 세부정보를 요약한 것이다.

표 7은 시뮬레이션에서 사용하는 네트워크와 기지국 파라미터들을 기술했다. 병렬 서버와 기지국 사이의 네트워크 지연은 평균 100ms의 지수 분포로 모델링된다. 병렬 서버로부터의 부 스트림은 기지국에서 이미 재결합하였기 때문에 기존의 시스템과 제안한 서버모델을 비교하는 관점에서 봤을 때 기지국과 이동국 사이의 네트워크 지연은 무시할 수 있다. 따라서 우리는 기지국과 이동국 사이의 무선 채널이 이상적이라 가정하고 기지국에서 채널 스케줄링을 위해 WFS 알고리즘(wireless fair service)을 사용한다[14].

그림 18은 기지국에 들어오는 멀티미디어 스트림의 트래픽을 나타낸다. 기지국의 대역폭이 2Mbps로

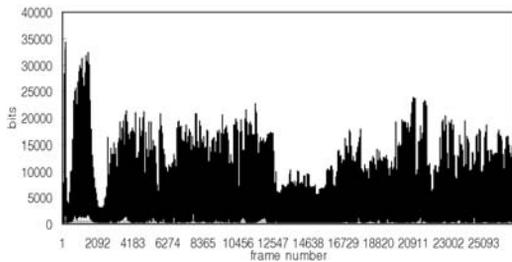


그림 17. Starwars-IV의 VOP 크기 대 프레임 수

표 6. Starwars-IV의 비디오 소스 정보

File format	QCIF
Number of frame per second	7.5 (frames/sec)
Running time	3600 sec
Total number of frames	26998
Total number of GOPs	2250
Number of frame per GOP	12

표 7. 시뮬레이션 파라미터

Network delay model between the server and a base station	Exponential distribution
Network delay between the single server and a base station	Avg: 100ms Min: 30ms Max: 120ms
Network delay between the parallel server and a base station	[I frame server] Avg: 100ms Min: 30ms Max: 120ms
	[P frame server] Avg: 60ms Min: 30ms Max: 120ms
	[B frame server] Avg: 80ms Min: 30ms Max: 120ms
Channel scheduling algorithm of the base station	WFS
Bandwidth of the base station	2Mbps
Buffer size of the base station	10Mbytes
Allocated bandwidth for mobile host	40Kbps
Buffer size of mobile host	90Kbytes
Upper threshold for buffer of the mobile host	0.75
Lower threshold for buffer of mobile host	0.3

제한되어있기 때문에 들어오는 스트림의 트래픽이 제한된 대역폭을 초과할 때 기지국은 초과되는 스트림을 서비스하지 못한다.

그림 19-(a)와 그림 20-(a)는 각각 기존과 제안한 병렬 서버 환경에서 'Starwars-IV' 비디오 스트림에 대한 이동국의 수신율을 나타낸다. 그리고 그림 19-(b)와 그림 20-(b)는 두 병렬 서버 모델들의 차이점을 보이기 위해 같은 데이터에 대해 정규화된 모바일 호스트의 수신율을 나타낸다. 기존의 병렬

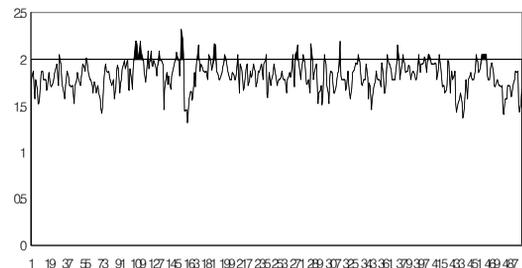


그림 18. 기지국으로 들어오는 멀티미디어 스트림의 트래픽

서버 모델에서 비디오 스트림을 전송할 때, 충돌이 발생한 기지국은 VOP 타입에 대한 고려 없이 들어온 모든 패킷을 버린다. 이 경우 I-VOP 하나의 손실은 한 GOP 내의 모든 VOP에 영향을 끼치고 그 시간동안 수신율은 0이 된다. 다시 말해, 한 GOP 내의 모든 VOP가 손상된다. 그러나, 제안한 병렬 서버 환경의 경우 QoS를 보장하기 위해 병렬 서버가 VOP 타입으로 비디오 VOP들을 식별하고 각기 다른 우선순위에 따라 처리하기 때문에 한 GOP 내의 모든 VOP가 손실되지는 않는다.

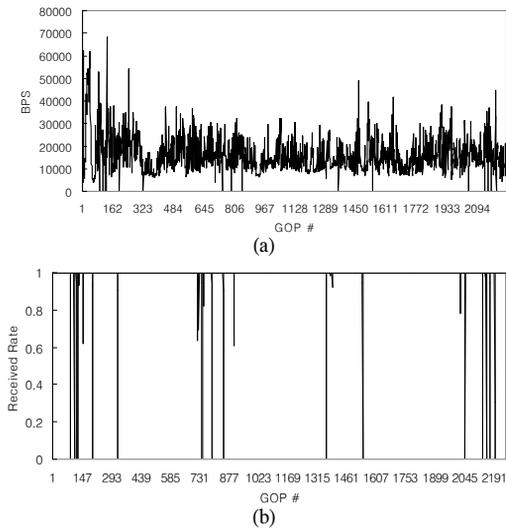


그림 19. (a) 기존 병렬 서버 환경에서 이동국의 수신율 (b) 기존 병렬 서버 환경에서 이동국의 정규화 된 수신율

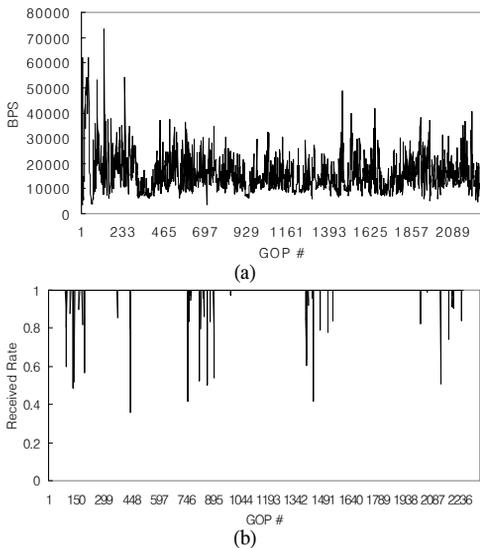


그림 20. (a) 제안된 병렬 서버 환경에서 이동국의 수신율 (b) 제안된 병렬 서버 환경에서 이동국의 정규화 된 수신율

이제 시뮬레이션 결과를 분석하고 두 병렬 서버 모델을 정량적으로 비교할 것이다. 기존 모델의 손실률은 180bit/GOP인 반면 제안한 모델의 손실률은 105bit/GOP이다. 그러므로 손실률만 고려하면 57%의 성능이 향상되었다.

끝으로, 두 개의 MPEG-4 비디오, 'Foreman'과 'Football'을 사용하여 시각적 관점에서 두 병렬 서버를 비교하였다. 그림 21-(a)와 그림 22-(a)는 원본 비디오의 표본 그림을 나타낸다. 그림 21-(b)와 그림 22-(b)는 기존 병렬 서버 환경에서의 이동국에서

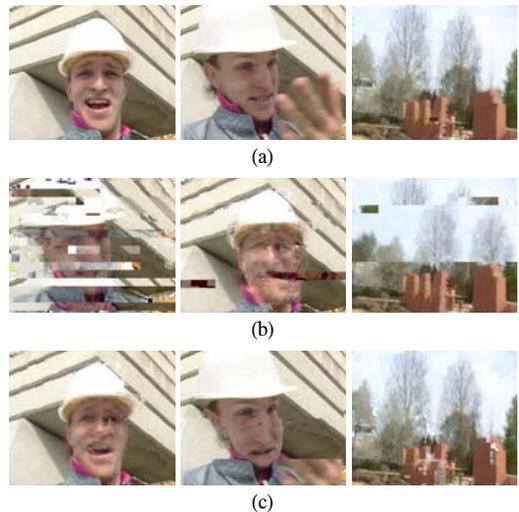


그림 21. (a) 'Foreman'의 원본 영상, (b). 기존 병렬 서버 환경에서 이동국이 받은 영상, (c). 제안된 병렬 서버 환경에서 이동국이 받은 영상

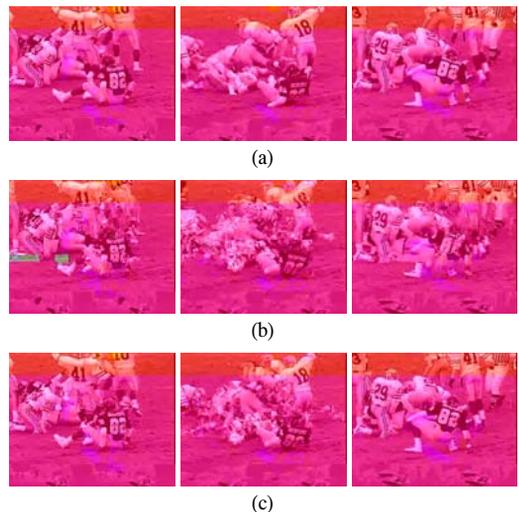


그림 22. (a) 'Football'의 원본 영상, (b) 기존 병렬 서버 환경에서 이동국이 받은 영상, (c) 제안된 병렬 서버 환경에서 이동국이 받은 영상

재생된 표본 그림을 나타낸다. 그리고 그림 21-(c)와 22-(c)는 제안된 병렬 서버 환경에서의 표본 그림을 나타낸다. 표본 그림으로부터, 제안된 모델이 멀티미디어 전송에서 손실의 영향을 줄일 수 있음을 알 수 있다.

VII. 결 론

본 논문에서, 병렬 서버의 무선 네트워크 환경에서 VOP 타입에 의거한 멀티미디어 전송과 동기화 모델을 제안하였다. 우리 연구의 목표는 부하 불균형, 충돌, 그리고 지터에 대처하는 향상된 멀티미디어 전송 기법을 개발하는 것이다. 따라서, 우리는 데이터 분할, 우선순위 큐잉, 그리고 멀티미디어 동기화로 분류되는 세 가지 이슈들을 고려했다.

첫째로, 데이터를 효과적으로 분할하기 위해 MPEG-4 비디오 스트림을 분할하고 재조합하는 프로그램 모듈을 구현한다. 인코딩 시 세 개의 부 스트림을 병합하기 위해 VOP 번호를 VOP 헤더에 삽입한다. 각 서버는 MPEG-4 VOP 타입으로 분할된 비디오 데이터를 분리하여 저장한다. 재병합 과정에서, MPEG-4 표준을 만족시키기 위해 VOP 넘버는 순서가 재배열되기 전에 VOP 헤더에서 삭제되어야만 한다. 둘째로, 혼잡 상황에서의 손실을 줄이기 위해 우선순위 큐 관리 정책을 제안한다. 큐가 가득 찼을 경우, 큐에서 높은 우선순위를 가진 패킷이 들어올 공간을 확보하기 위해 가장 낮은 우선순위를 가진 패킷을 삭제한다. 기지국에서 QoS를 높이기 위해 I-VOP에 가장 높은 우선순위를 주고, 다른 VOP도 그것의 우선순위에 따라 스케줄링 한다. 왜냐하면 한 GOP 내의 I-VOP가 손실되거나 손상되면 그 GOP안의 모든 VOP가 손상되기 때문이다. 셋째로 지터를 보상하기 위해 동기화 알고리즘을 제안한다. 멀티미디어 동기화를 유지하기 위해 전송률 제어 장치와 지연 버퍼를 사용한다. 버퍼 오버플로우나 언더플로우 같은 정상적이지 않은 버퍼 상황이 감지되기 전에, 제안된 알고리즘은 버퍼 레벨 즉, BUT와 BLT의 임계값이 감지되는 것에 따라서 동기화 절차를 수행하기 시작한다. 따라서 제안된 기법은 비정상적인 버퍼 상황을 상당히 줄일 수 있다.

마지막 절에서는, 두 병렬서버 모델의 시뮬레이션 결과를 수치적으로 분석하였으며, 2개의 MPEG-4 비디오 소스, 'Foreman'과 'Football'을 사용하여 가시적으로 비교하였다. 기존 모델의 손실률은 180 bit/GOP이다. 그러나, 제안된 모델의 손실률은 105

bit/GOP이며 손실률만 고려하면 57%의 성능이 향상되었다. 끝으로, 우리는 두 개의 MPEG-4 비디오를 사용한 영상 인식의 관점에서 두 병렬서버를 비교했다. 표본 그림으로부터, 제안된 기법이 멀티미디어 전송에서 패킷 손실의 영향을 줄일 수 있다는 결과를 이끌어 낼 수 있었다. 따라서 제안한 방법은 현재 인터넷 환경에서 양질의 멀티미디어 서비스를 제공하기 위한 유용한 기술이 되리라 기대된다.

참 고 문 헌

- [1] J. LEE, "Parallel Video Servers: A tutorial," IEEE Multimedia April-June 20-28, 1998.
- [2] K. R. Rao, Z. S. Bojkovic, and D. A. Milovanovic, Multimedia Communication Systems, Prentice Hall PTR, 2002.
- [3] F. Pereira and T. Ebrahimi, The MPEG-4 Book, Prentice Hall PTR, 2002.
- [4] W. T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," IEEE Transaction on Circuits and Systems for Video Technology, Volume: 11, Issue: 3, pp. 373-386, March 2001.
- [5] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 2nd Edition, Addison Wesley, 2003.
- [6] Z. Wang, Internet QoS, Morgan Kaufmann Publishers, 2001.
- [7] N. Shivakumar, C. J. Sreenan, B. Narendran, and P. Agrawal, "The Concord algorithm for synchronization of networked multimedia streams," in Proc. of IEEE International Conference on Multimedia Computing and Systems, pp. 31-40, May 1995.
- [8] J. Takahashi, H. Tode, and K. Murakami, "QoS enhancement methods for MPEG video transmission on the Internet," Proceedings of the Seventh International Symposium on Computers and Communications, pp. 106-113, July 2002.
- [9] X. Meng, H. Yang, and S. Lu, "Application-oriented multimedia scheduling over lossy wireless networks," Proceedings of the Eleventh International Conference on Computer Commu-

nications and Networks, pp. 256-261, Oct. 2002.

[10] A. Boukerche, S. B. Hong, and T. Jacob, "An efficient synchronization scheme of multimedia streams in wireless and mobile systems," IEEE transactions on Parallel and Distributed Systems, Vol 13, No. 9, Sept. 2002.

[11] C. J. Sreenan, J. C. Chen, and B. Narendran, "Delay reduction techniques for playout buffering," IEEE Transactions on Multimedia, June 2000.

[12] P. V. Rangan, S. S. Kumar, and S. Rajan, "Continuity and synchronization in MPEG," IEEE Journal on Selected area in Communications, Jan. 1996.

[13] H. Zhu, G. Zeng, and I. Chlamtac, "Control scheme analysis for multimedia inter-and intra-stream synchronization," ICC '03 IEEE International Conference on Communications, Vol. 1, pp. 11-15, May 2003.

[14] S. Lu, T. Nandagopal and V. Bharghavan, "Design and analysis of an algorithm for fair service in error-prone wireless channels," ACM Wireless Networks Journal, Vol. 6, No. 4, pp. 323-343, Jan. 2000.

신 광 식 (Kwang-sik Shin) 준회원



2001년 2월 인하대학교 전자공학과 졸업
 2003년 2월 인하대학교 전자공학과(석사)
 2003년 3월~현재 인하대학교 전자공학과 박사과정
 <관심분야> 멀티미디어 데이터

통신, 컴퓨터 네트워크, 무선 통신, 시스템 프로그래밍

백 석 균 (Seok-Kyun Baek) 준회원



2004년 2월 인하대학교 전자공학과 졸업
 2004년 3월~현재 인하대학교 전자공학과 석사과정
 <관심분야> 컴퓨터 구조, 시스템 디자인

윤 완 오 (Wan-oh Yoon) 준회원



2000년 2월 경기대학교 전자공학과 졸업
 2002년 2월 인하대학교 전자공학과(석사)
 2002년 3월~현재 인하대학교 전자공학과 박사과정
 <관심분야> 분산 처리 시스템, 병렬프로그래밍, 컴퓨터 아키텍처

정 진 하 (Jin-ha Cheong) 준회원



1992년 2월 인하대학교 전자공학과 졸업
 1994년 2월 인하대학교 전자공학과(석사)
 1994년~1999년 한미 기술연구소(주)
 2000년 3월~현재 인하대학교

전자공학과 박사과정
 <관심분야> 컴퓨터구조, 병렬 및 분산 처리 시스템, 시스템 디자인, Fault-tolerant computing

최 상 방 (Sang-bang Choi) 정회원



1981년 2월 한양대학교 전자공학과 졸업
 1981년~1986년 LG 정보통신(주)
 1988년 3월 University of Washington(석사)
 1990년 8월 University of Washington(박사)

1991년~현재 인하대학교 전자공학과 교수
 <관심분야> 컴퓨터 구조, 컴퓨터 네트워크, 무선 통신, 병렬 및 분산 처리 시스템, antFault-tolerant computing