

# 실시간 인터넷 동영상 전송을 위한 효율적인 패킷화 기법

정회원 김 효 현, 유 국 열\*

## An Efficient Packetization Method for the Real-time Internet Video Transmission

Hyo-Hyun Kim, Kook-Yeol Yoo\* *Regular Members*

### 요 약

본 논문에서는 패킷화 부가 정보를 효과적으로 감축할 수 있는 패킷화 기법을 제안한다. 이를 위해 기존의 인터넷 동영상 전송을 위한 패킷화 기법을 패킷 길이에 따른 패킷 손실률에 대한 특성을 검증하였다. 인터넷망을 이용한 동영상 전송 실험을 통해 패킷의 길이가 경로 MTU (path MTU)를 넘을 경우 급격한 패킷 손실률이 증가하고, 이에 따른 화질의 급격한 저하 현상이 발생함을 확인하였다. 그러나 패킷의 길이가 감소하게 되면, 각 패킷별로 부가정보들을 전송해야 하므로, 페이로드에 해당하는 부호화된 동영상 비트열에 할당할 수 있는 비트율의 감소로 인한 화질이 감소하게 된다. 그러므로 효율적인 패킷화를 위해서는 각 패킷들이 경로 MTU에 근접한 길이를 갖도록 패킷화해야 한다. 본 논문에서는 패킷화 문제가 운영체제의 메모리 관리 기법과 유사함을 지적하고, 운영체제에서 사용되는 메모리 관리 기법을 동영상 패킷화에 적용하는 것을 제안하였다. 실험을 통해서 기존의 대표적인 패킷화 방식인 순차적 패킷화 방식과 비교하였고, 제안방식이 패킷화 부가정보를 최대 28.3%까지의 감축할 수 있음을 보였다.

**Key Words** : internet video transmission; packetization; overhead reduction, dynamic memory allocation

### ABSTRACT

In this paper, we propose an efficient packetization method to reduce the packetization overhead. For the purpose, we firstly verify the relationship between packet length and packet loss rate. The empirical results show that as the packet length is larger than the path MTU, the packet loss rate is drastically increased, producing poor visual quality at the receiver side. However, as the length of the packet is reduced, we should transmit more packets per frame and the packetization overhead will be increased. This increase in the packetization overhead reduces the number of bits allocated to the video data, resulting in the low visual quality. Therefore, each packet should be packetized to have the packet length close to the path MTU. In this paper, we show that the this process of the packetization with the constraint on the packet length is very similar to the dynamic storage allocation in the operating system. We had thoroughly surveyed the dynamic storage allocation methods used in the recent operating systems and propose to use the allocation methods for the video packetization. We empirically show that the proposed method can reduce the packetization overhead upto 28.3%, compared with the conventional sequential packetization method which have been widely used in Internet video transmission.

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음

\* 영남대학교 전자정보공학부 미디어통신연구실 (kyoo@yu.ac.kr)

논문번호: KICS2006-04-173, 접수일자: 2006년 4월 15일, 최종논문접수일자: 2006년 6월 16일

## I. 서론

초고속 인터넷의 광범위한 보급으로 인해 인터넷을 통한 동영상 전송에 대한 관심과 응용은 폭발적으로 증가하고 있다. 부호화된 동영상 신호는 일반적인 데이터들과 매우 상이한 특성을 가지고 있기 때문에 기존의 데이터 전송 문제와는 확연히 구별된다. 특히 실시간 동영상 전송 문제의 경우에는 종단간 지연(end-to-end delay)이 150msec 이상인 경우 올바르게 수신된 패킷도 손실된 패킷으로 간주된다<sup>1),2)</sup>. 또한 동영상 압축 시에 높은 부호화 효율을 위해서 널리 사용되는 가변길이장 부호(Variable Length Codeword; VLC)는 패킷이 손실된 경우 수신단에서 가변길이장 복호화 과정에서 동기화 문제를 유발한다. 즉, 압축된 동영상 비트열 중에 일부의 비트열의 손실은 이후의 비트열들의 복호화를 불가능하게 만든다. 이런 문제점과 각 프레임별 및 일정 크기의 세부 단위별로 동기 코드를 제공한다. 이런 연유로 기존의 부호화된 동영상 비트열의 패킷화는 복수개의 동기 단위들을 패킷에 담아서 전송한다. 이는 한 동기 단위의 동영상 비트열이 복수개의 패킷으로 전송될 경우에는 한 개의 패킷이라도 손실된 경우 가변길이장 복호화를 불가능하게 만들 수 있다는 사실에 기인한다<sup>3),4),5)</sup>.

실시간 동영상 비트열의 전송에는 RTP/UDP/IP (Real-time Transport Protocol)이 널리 사용되고 있다<sup>6)</sup>. 수신단에서 복호된 프레임(frame)을 올바른 시간에 재생하기 위해서는 각 패킷에 담겨 있는 프레임의 시간정보를 알아야 한다. 일반적으로 부호화된 프레임은 복수개의 패킷에 나누어서 전송되므로, 각 패킷별로 프레임의 시간 정보를 포함시켜서 전송해야 한다. 또한 각 프레임별로 부호화 방식과 파라미터들이 달라질 수 있으므로, 이에 대한 정보도 각 패킷별로 제공해야 한다. 최소의 필요한 부가정보를 제공하기 위해서 IETF에서는 RTP 페이로드 형식(payload format)을 정의하고 있다<sup>7),9)</sup>. 그러나 수신단에서 복호화를 위해 제공되는 이런 부가 정보들은 네트워크를 통해 동영상 패킷을 전송할 경우에 추가적인 대역폭을 요구하므로, 최소의 개수의 패킷으로 전송하는 것이 전송 효율 측면에서 유리하다. 즉, 부가 정보량을 줄임으로써 이에 해당되는 비트량을 동영상 부호화에 추가적으로 할당하여 전송 화질을 높일 수 있다.

패킷화 효율을 높이기 위해 패킷별로 페이로드의

크기, 즉 패킷 길이를 크게 할 경우에는 다른 문제를 유발시킬 수 있다. 본 논문의 2절에서 실제의 네트워크상에서 수행한 실험 결과에 따르면 각 라우터(router)에서 패킷의 길이가 MTU(Maximum Transmission Unit)를 넘어설 경우에는 패킷의 분할(segmentation)이 발생하게 되고, 동영상 신호의 실시간성과 결부되어서 패킷 손실률이 급격히 증가함을 실험을 통해서 검증하였다. 그러므로 효율적인 패킷화를 위해서는 각 패킷별 길이가 MTU 크기를 상한선(upper bound)를 하면서 패킷의 개수를 최소화하는 기법이 필요하다. 또한 동영상 신호는 시공간적인 특성과 가변길이장 부호화에 의해서 같은 크기의 영상 영역들에 대해서 서로 다른 비트량을 발생시키므로, 가변적인 크기를 가지는 동기 단위의 비트열들을 패킷 크기의 상한선을 만족하면서 패킷의 개수를 최소화시켜야 한다.

본 논문에서는 이런 패킷화 문제가 운영체제에서 효율적인 동적 기억장치 할당 기법과 동일함을 지적하고, 기존의 동적 기억장치 할당 기법들을 패킷화에 적용하여 효율적인 패킷화 방식을 제안한다. 제안 기법을 기존의 가장 대표적인 패킷화 기법인 순차적 패킷화 방식과 비교하여 성능을 평가하였고, 실험 결과에 따르면 제안 방식은 기존 방법에 비해 효과적으로 패킷의 수를 감소시켜 최대 28.3%의 패킷화 부가 정보량을 감축할 수 있음을 확인하였다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 패킷화 방식에 대해 설명한다. III장에서는 동영상 신호의 패킷화와 운영체제의 메모리 관리 기법과 동일함을 보이고, 제안 패킷화 방식에 대해 서술한다. 제안 방식에 대한 성능 평가와 결론은 IV장과 V장에서 기술한다.

## II. 실시간 인터넷 동영상 전송을 위한 기존의 패킷화 방식 분석

### 2.1 실시간 인터넷 동영상 전송 시스템 및 순차적 패킷화 방식

그림 1에 전형적 실시간 인터넷 동영상 전송 시스템을 나타내었다. 부호화하지 않은 원영상은 영상 부호기(video encoder)를 통해서 압축된 비트열로 변환된다. 각 프레임별로 시공간적인 특성에 따라 압축된 비트열은 각 프레임별로 가변적인 길이를 갖는다. 한 개의 프레임은 복수개의 독립적으로 복호화 및 동기화를 할 수 있는 영역들로 분할된다. 독립적으로 복호화 할 수 있는 단위는 동영상 압축

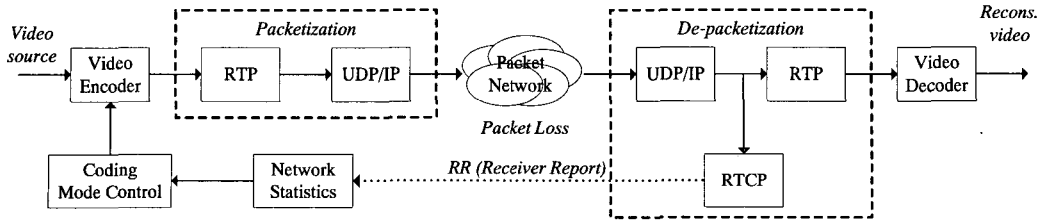


그림 1. 실시간 인터넷 동영상 전송 시스템

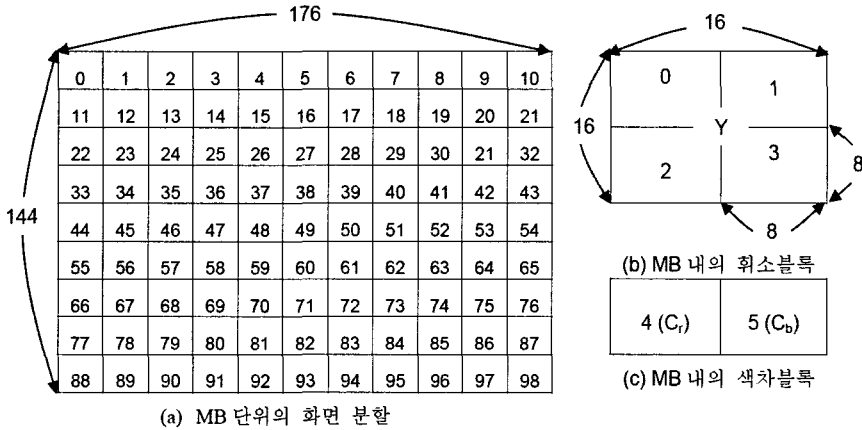


그림 2. 프레임 분할 방법 및 MB과 GOB의 구성; QCIF 영상(휘도성분 176화소/줄x144줄, 색차신호의 경우 4:2:0형식으로 부표본화됨)의 경우 화면 당 99개의 MB들, 각 MB별로 4개의 휘도 블록 및 2개의 색차블록으로 구성됨. MB들로 구성된 가로줄을 GOB라고 부름

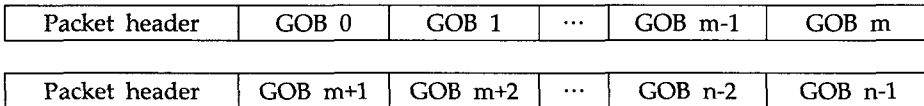


그림 3. 2개의 패킷으로 전송되는 압축된 프레임 비트열

표준의 종류에 따라 슬라이스(slice; MPEG-1, MPEG-2 및 MPEG-4) 또는 GOB(Group of Block; H.261, H.263, H.264)이라고 불린다. 본 논문에서는 H.263 부호기를 실험에 사용했으므로, 편의상 상기한 독립적인 복호화 단위를 GOB으로 통칭한다. 그림 2는 QCIF(176 화소/줄 x 144줄, 4:2:0 칼라 부표본화 형식) 영상의 매크로 블록 및 GOB 단위의 분할을 보여준다. 그림 2에서 각 블록들은 16x16 크기의 휘도 성분을 나타내고, 이에 해당하는 두 개의 8x8 색차 신호 블록을 매크로 블록(macroblock; MB)이라 부른다. 그림 2에서 매크로 블록의 줄들이 GOB를 형성한다. 즉, 0번째 GOB는 그림에서 0번에서 10까지의 매크로 블록들로 형성되고, 5번째 GOB는 44번에서 54의 매크로 블록으로 구성된다. 부호화된 영상 비트열 중에 복수개의 GOB들은 패킷단위로 묶여져서 RTP(Real-time Transport Protocol) 및 각 동영상 부호화 표준별로 정의된 RTP 페이로

드 형식에 따라 부가정보들이 추가된 후에 UDP/IP 프로토콜을 이용해서 전송이 된다.

각 패킷의 페이로드는 복수개의 GOB들이 프레임내의 위치하는 순서별로 각 패킷에 페이로드로 담기게 된다. 그림 3은 2개의 패킷들에 한 개의 프레임에 해당하는 압축된 비트열이 페이로드로 담기는 모습을 보여준다. 그림 3에서 입력 영상의 형식이 그림 2와 같은 QCIF 형식인 경우에는 n=9에 해당하게 된다. 또한 그림 3에서 패킷 헤더에 해당하는 데이터는 각 표준별로 페이로드 형식에서 정의한 데이터들로 구성된다. H.263의 경우에 패킷 헤더의 정보는 RFC-2190에서 정의하고 있다.

### 2.2 패킷의 길이에 따른 재생화질과의 상관관계 분석<sup>(10)</sup>

본 분석에서는 인터넷으로 실시간 동영상 비트열을 전송함에 있어서 패킷의 길에 따른 영상의 화질

표 1. 패킷화 오버헤더 (CIF, 10Hz)

Packets/picture	1	2	3	6	9	18
Packetization Overhead (kbps)	4.0	8.6	13.1	26.8	40.5	81.5

과 패킷 손실의 영향을 실제 네트워크를 통한 전송 실험을 통해 분석함으로써 인터넷 동영상 전송할 때 고려해야 점을 보인다. 본 실험에서 사용한 영상 부호기는 H.263을 사용하였고, 전송환경은 한국통신에서 제공하는 대학망과 개인 가정의 vDSL간의 실시간 인터넷 동영상 전송 실험을 수행하였다. 실시간 영상 데이터 전송을 위해 RTP/UDP/IP 기반의 프로토콜을 사용하였다. 표 1은 10Hz의 CIF(화도 크기 352x288; 4:2:0 컬러 부표본화 형식) 영상을 패킷화를 할 때 프레임별 패킷수에 따른 오버헤더 정보를 보여준다. 그림 4는 128kbps로 부호화된 container, foreman, hall CIF영상(300프레임, 10Hz) 들에 대해 패킷화에 따른 화질의 변화를 측정하였다. 그림 4에 따르면 패킷의 개수가 많아질 수록 패킷화 부가정보의 증가로 동영상 부호화에 할당할 수 있는 비트수가 감소하고, 이에 따른 재생화질의 저하로 나타남을 알 수 있다. 그러므로 보다 나은 재생 영상의 화질을 위해서는 패킷 길이 또는 프레임별 패킷의 개수는 줄여야 함을 알 수 있다.

그림 5에는 패킷 길이가 1,500 바이트 이상이 되는 경우 패킷의 손실률이 급격하게 증가함을 보여 준다. 여기에서 1,500 바이트가 현재 실험환경인 LAN의 MTU(Maximum Transmission Unit)를 의미한다. 이것은 IP망에서 일어나는 패킷 분할이 패킷 손실률에 매우 큰 영향을 준다는 것을 의미한다. 즉 MTU가 비디오 신호와 같은 대용량 데이터를 전송함에 있어서 신뢰할 수 있는 전송을 위해서 패킷 길이의 임계값이 될 수 있음을 보여준다. 그러므로 패킷화 과정에서 패킷별 길이는 이 임계값 이하로 선택되어야 함을 알 수 있다. 나아가서 이러한 통신이 원거리의 종단 간 응용에서 이루어진다면, 많은 경로 상에 존재하는 라우터들에 의해서 서로 다른 MTU 존재하게 된다. 이 경우 경로 상에 존재하는 라우터의 MTU 중에서 가장 작은 MTU를 경로 MTU (Path MTU)라고 하는데, 이 경로 MTU를 패킷 길이의 임계값으로 정하게 될 것이다.

그림 5를 확대해서 그린 그림 6에서 384kbps의 평균 패킷 길이가 534byte인 경우의 PSNR은 36dB

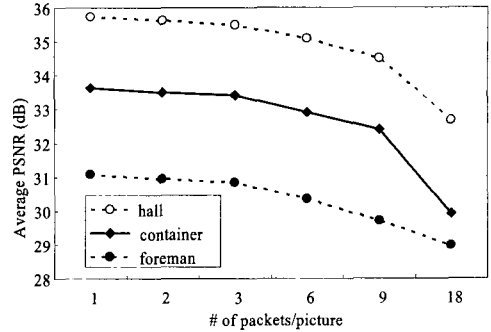


그림 4. 패킷화 오버헤드가 PSNR에 미치는 영향

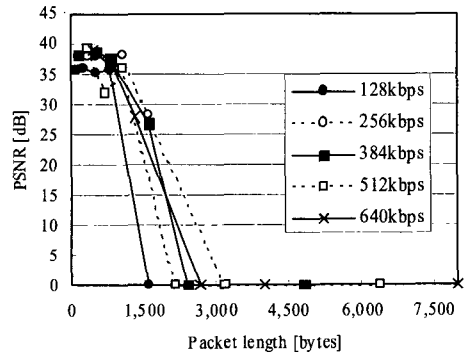


그림 5. 패킷 길이에 따른 평균 PSNR의 비교

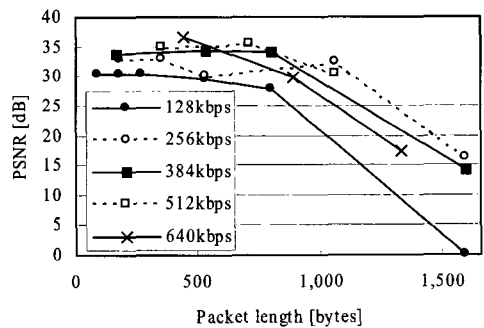


그림 6. 패킷 길이에 따른 평균 PSNR의 비교(CIF, foreman)

이나, 640kbps의 평균 길이가 1,333바이트인 패킷의 경우에는 평균 PSNR값이 17dB에 불과했다. 주의할 점은 평균 길이가 1,333 바이트인 경우에 모든 패킷들의 길이가 1,500바이트 미만임을 의미하지 않는다. 이와 같은 결과는 주어진 네트워크 환경에서 패킷 길이를 주의 깊게 선택하였다더라면 보다 적은 대역폭을 사용함으로써 보다 나은 재생 영상의 화질을 제공할 수 있음을 보여준다.

이상의 실제 네트워크를 통한 실험 결과에서 다음과 같은 결론을 유도할 수 있었다. 기존의 패킷화

방법에서는 가변길이장 부호화를 사용하는 경우에 패킷의 길이가 1,500 바이트의 임계값을 넘지 않도록 패킷화를 해야 함을 알 수 있었다. 그리고 한 영상을 구성하는 패킷의 개수를 최소화해야 정해진 대역폭에서 영상 부호화에 보다 많은 대역폭을 할당할 수 되어 보다 나은 재생영상 화질을 수신측에 제공할 수 있다.

### III. 운영체제의 동적 기억장치 관리 기법을 응용한 제안 패킷화 방식

#### 3.1 운영체제의 동적 기억장치 관리 기법 및 패킷화 방식의 유사성

본 절에서는 운영체제에서 사용되는 동적기억장치 할당기법(dynamic storage allocation)<sup>[11, 12]</sup>과 패킷화시의 페이로드 결정 방식<sup>[13, 14]</sup> 사이의 유사성을 보이고자 한다.

운영체제에서 메모리를 관리하는데 있어서 데이터를 메모리에 동적으로 저장하는 경우에 가용 메모리 공간을 일정한 크기의 용량(n)의 메모리 블록으로 나누고, 이 기본 블록 단위로 데이터를 채워 나가게 된다. 동적 기억장치 할당 문제는 가용 공간 리스트로부터 크기 n 요구를 어떻게 만족시켜 줄 수 있느냐 하는 문제로 요약된다. 기억 장치에서 가용 공간의 집합이 어느 가용 공간에 할당하기가 가장 좋은지를 검색할 때 사용된다. 이러한 검색을 위해 사용되는 방식들로 다음과 같은 순차 적합(Next-Fit), 최초 적합(First-fit), 최상 적합(Best-Fit), 최악 적합(Worst-Fit)의 네 가지 방식이 가장 일반적인 방식들로 널리 알려져 있다.<sup>[12]</sup>

- 순차 적합 (NF) : 마지막에 사용한 가용 공간에 할당한다. 가용 공간이 충분하지 않을 경우 다음 가용 공간에 할당한다.
- 최초 적합 (FF) : 충분히 큰 첫 번째 가용 공간에 할당한다. 검색은 가용 공간 집합의 시작에서 하거나 이전의 최초 적합 검색이 끝났던 곳에서 시작될 수 있다. 충분히 큰 가용 공간을 찾았을 때 검색을 끝낼 수 있다.
- 최상 적합 (BF) : 충분히 큰 가용 공간들 중에서 가장 작은 가용 공간을 할당한다. 리스트가 크기순으로 되어있지 않다면 전 리스트를 검색해야만 한다. 이 방법은 가장 작은 또 다른 가용 공간을 만들어 낸다.
- 최악 적합 (WF) : 가장 큰 가용 공간에 할당

한다. 크기순으로 정렬되어 있지 않으면 전 리스트를 검색해야한다. 이 방법은 가장 큰 가용 공간에 할당하기 때문에 최상 적합의 경우의 가장 작은 또 다른 가용 공간을 만들어 내는 것 보다 더 유용하다.

표 2에 본 논문에서 해결하고자 하는 동영상 비트열의 패킷화 과정과 운영체제에서 사용하는 동적 메모리 할당 기법간의 유사성을 정리하여 나타내었다.

#### 3.2 제안한 패킷화 방식

기존의 패킷화 방법<sup>[9, 13, 14]</sup>들은 순차적으로 비트 스트림을 패킷화하는 방법으로 3.1절에서 설명한 NF(Next Fit) 방식과 동일하다. 압축된 영상의 비트 스트림을 GOB단위로 임계값을 넘지 않는 범위 안에서 순차적으로 하는 방식으로 패킷화된다. 이 방법은 각각의 패킷이 독립적으로 복호화하기 위해서는 영상의 헤더정보를 포함하여 보내야 한다. 이로 인해 IP/UDP/RTP 헤더정보와 영상 헤더정보의 중복되는 부가비트를 사용하게 된다. 생성되는 패킷의 개수가 최소가 될수록 부가 비트량을 줄일 수 있으며, 이로 인한 영상에의 비트 할당량을 증가 시킬 수 있다.

FF 패킷화 방식과 BF 패킷화 방식 또한 동일하게 순차적으로 GOB 단위의 패킷화를 하는 방법이다. FF 패킷화 방식은 현재 생성된 패킷들 중에서 현재의 GOB 스트림을 추가해도 임계값을 넘지 않는 범위를 만족하는 패킷을 검색하여서 처음 만나는 패킷에 추가하여 패킷화하는 방법이다. 그리고 BF 패킷화 방식은 FF 패킷화 방식과 달리 패킷을 검색할 때 범위를 만족하는 패킷 중에서 패킷화를

표 2. 동영상 비트열의 패킷화 문제와 동적 메모리 할당간의 유사성

문제	동영상 비트열의 패킷화	동적 메모리 할당
대상 데이터	전송하려는 압축된 동영상 비트열	저장하고자 하는 데이터
처리 특성	GOB단위로 독립적인 복호화 가능	입력 데이터별로 저장함.
처리할 데이터의 크기	가변적인 GOB 데이터의 크기	가변적인 입력 데이터의 크기
기본단위크기의 상한	경로 MTU 이하로 패킷화 수행	기본단위이하로 데이터를 저장함.

표 3. QP 값에 따른 추가 비트 발생량 비교 (CIF, Foreman 영상) [ bits/frame ]

QP	NF (기존 방식)	제안 방식			
		FF	BF	FFD	BFD
5	1,888.6	1,773.9	1,773.3	1,746.8	1,746.8
7	1,143.8	1,104.9	1,104.8	1,095.6	1,095.6
10	720.7	712.9	712.9	709.9	709.9
15	452.0	451.8	451.8	451.8	451.8

표 4. QP 값에 따른 추가 비트 발생량 비교 (CIF, Pipo 영상) [ bits/frame ]

QP	NF (기존 방식)	제안 방식			
		FF	BF	FFD	BFD
5	2,195.1	1,785.8	1,785.8	1,725.9	1,725.8
7	1,440.1	1,351.9	1,351.4	1,342.6	1,342.6
10	863.7	841.7	841.7	841.4	841.4
15	637.3	628.7	628.7	621.0	621.0

한 후 임계값에 가장 근접하는 패킷에 현재 GOB를 추가하여 패킷화하는 방식이다. 이와는 다른 방법으로 FFD, BFD 패킷화 방식들이 있다. 이 방식들은 ITU-T H.263 부호기를 통하여 생성된 GOB의 크기들을 이미 알고 있는 시점에서 GOB들은 길이 기준으로 정렬을 한 후에 패킷화를 한다. 이러한 패킷화를 통한 실험 결과는 다음 장에서 자세히 다루기로 한다. FFD 패킷화 방식은 GOB 단위의 길이 정보를 이용하여 GOB 단위로의 내림차순 정렬을 한 후에 이 길이 목록을 기준으로 FF 패킷화를 하는 방식이다. BFD 패킷화 방식도 GOB 단위의 정렬된 길이 정보를 이용하여 BF 방식을 적용한 것이다. 이런 패킷화 방식을 적용하기 위해서는 GOB별로 독립적으로 복호화가 가능해야 한다. 즉 GOB들이 순서에 상관없이 있을 경우에도 복호화가 가능해야 한다. 이런 GOB별로 독립적인 복호화가 가능하다는 근거는 참고 문헌<sup>15, 16)</sup>에 증명이 되어 있다.

#### IV. 실험 및 분석

##### 4.1 실험 환경

III장에서 제안된 NF, FF, BF, FFD, BFD 패킷화 방식을 ITU-T H.263 부호기를 이용해 생성된 비트열에 적용하여 실험을 수행하였다. 다양한 영상들에 대해 제안 방식을 검증하기 위해 CIF 형식의

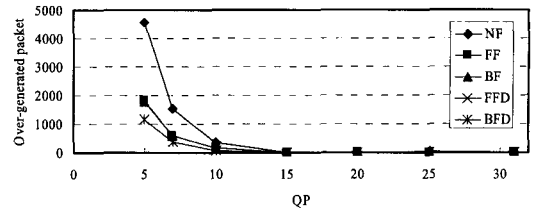


그림 7. 패킷화에 따른 추가 발생 패킷의 개수 비교(CIF, Foreman); NF는 기존의 패킷화 방식[9]을 의미

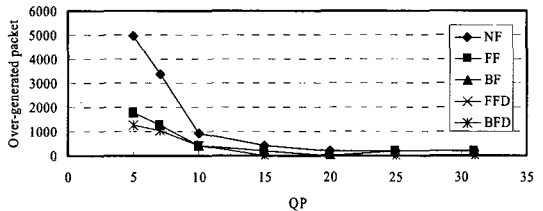


그림 8. 패킷화에 따른 추가 발생 패킷의 개수 비교(CIF, Pipo); NF는 기존의 패킷화 방식[9]을 의미

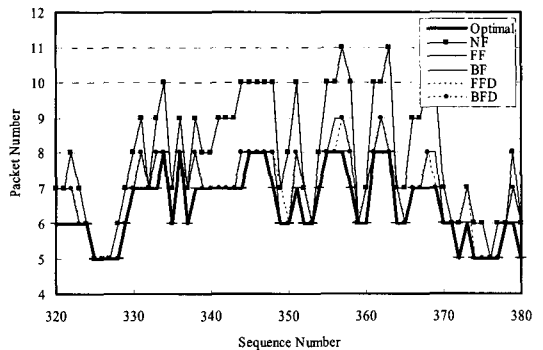


그림 9. 패킷화에 따른 영상 순서에 따른 발생 패킷 개수 비교(CIF, Foreman, QP = 5, Sequence No. 320 ~ 380), NF는 기존의 패킷화 방식[15]을 의미, Optimal은 패킷 개수의 하한치(lower bound)를 의미

“Foreman” 및 “Pipo” 실험 영상들을 사용하였다. NF, FF, BF, FFD, BFD 패킷화 방식의 성능 평가 함수로 생성된 패킷의 개수 및 패킷화 후 발생한 비트량을 사용하였다. 부호화시의 양자화 간격(QP)을 5, 7, 10, 15로 고정하여 다양한 비트율(bitrate)에 대해 성능을 평가하였다. 안정적인 결과치를 얻기 위해 총 10,000 프레임에 대해서 부호화 및 패킷화 과정 실험을 수행해서 평균을 취해 결과치를 얻었다.

##### 4.2 제안 방식의 패킷화 성능 평가

표 3과 4에서 보인 결과치는 원래 영상 부호기에서 발생한 비트들에 비해서 패킷화 과정에서 추가적으로 발생한 부가정보(overhead)에 해당되는 비트

를 의미한다. 표 3과 4의 결과치에 따르면 양자화 간격이 작을수록 제안 방식의 성능이 상대적으로 우수하고, 양자화 간격이 커질수록 제안 방식의 효과가 줄어들음을 알 수 있다. 이는 양자화 간격이 커질수록 프레임별로 발생하는 비트수가 줄어들어 자연스럽게 경로 MTU의 제한 조건을 만족하는 빈도가 높아지는 것에 기인한다. 각 제안 방식들의 성능을 비교하여 보면 제안 방식인 FFD 방식과 BFD 방식이 가장 우수함을 알 수 있다.

부호화된 총 10,000 프레임에서 발생한 비트수를 경로 MTU로 나눈 값이 전체 프레임에 대한 패킷의 개수의 하한치를 의미한다. 그림 7과 8에서는 이 값에 비해 각각의 패킷화 방식들별로 추가적으로 발생한 패킷의 개수를 나타냈었다. 그림에서 알 수 있듯이 양자화 간격이 낮은 경우에 기존의 순차적 패킷화 방식에 비해서 제안한 방식들이 매우 작은 수의 패킷을 발생시킴을 알 수 있다. 그러나 이런 제안 방식의 효율성은 양자화 간격이 커질수록 효과가 낮아짐을 알 수 있다. 그러므로 제안 방식의 고화질의 동영상 전송에 적합함을 알 수 있다.

각 프레임별로 제안 방식의 성능을 비교하기 위해서 각 프레임별로 발생한 패킷의 개수를 그림 9에 나타내었다. 그림 9에서 최적 패킷의 개수는 프레임에서 발생한 비트량을 경로 MTU값으로 나눈 값을 의미한다. 그림 9에 따르면, 제안 방식들인 FFD 및 BFD 방식이 최적치와 비슷한 성능을 보임을 알 수 있다.

#### 4.3 제안 방식의 연산 복잡도 평가

IV.2절에서 보인 패킷화 성능 평가 결과를 보면 제안 방식들인 FF, BF, FFD 및 BFD 방식들이 거의 비슷한 성능을 보이고 있고, 상대적으로 FFD 및 BFD 방식이 우수함을 알 수 있다. 그러므로 패킷화 효율성이 유사한 경우에 적합한 패킷화 방식을 선정하는 일은 각 방식별로 필요로 하는 복잡도에 의해 결정할 수 있다. 이를 위해 현재 실험환경에서의 CPU 사용 시간을 이용하여 각 패킷화의 복잡도를 비교하여 보았다. 그리고 현재 실험 영상의 크기가 CIF 영상에 한정되어 있으므로, 영상의 크기를 4CIF, HDTV급의 영상을 이용하여 영상의 크기를 다르게 하면서 실험하여, 패킷화의 복잡도를 비교 분석하였다.

표 5에서 실험에 사용한 영상 종류별 크기와 GOB의 개수를 나타내고, 각 영상의 종류별로 부호화 시간을 나타내었다. 표 6에서 각 방식별로 패킷

표 5. 영상 종류별 크기, GOB 개수 및 평균 부호화 시간 비교; 부호화 파라미터 H.263 baseline, QP = 10

종류	영상의 크기	GOB 개수	부호화 시간 [sec/frame]
CIF	352 × 288	18	0.176
4CIF	706 × 576	36	0.621
HDTV	1280 × 1024	64	2.272

표 6. 영상 크기별 패킷화에 따른 CPU 사용 시간의 비교 (QP = 10); HDTV1 [basketball영상], HDTV2 [military영상]; CPU time per frame [usec]

	NF (기존 방식)	제안 방식			
		FF	BF	FFD	BFD
CIF	0.97	1.87	1.98	3.06	2.54
4CIF	1.25	4.11	7.14	7.39	9.32
HDTV1	2.45	17.98	65.10	36.94	81.92
HDTV2	1.61	7.35	29.27	21.85	59.93

화에 사용된 CPU 시간을 측정하여 나타내었다. 표 6의 결과에 따르면 FFD 패킷화 방식이 BFD 방식에 비해 CPU사용 시간이 적음을 알 수 있었다. 그러나 이 시간을 영상의 평균 부호화 시간과 비교 하였을 때 FFD의 경우 최대 2.8만분의 1에서 최소 5.8만분의 1의 시간이 사용됨을 알 수 있었다. 그러므로 패킷화에 소요되는 시간이 전체 시스템에서 무시해도 되는 크기임을 알 수 있다. 전체적으로 살펴보면 모든 방식 중에서 FFD 및 BFD 방식이 패킷화 효율 측면에서 가장 우수한 성능을 보이고, 이들 간의 성능차이는 커지 않다. 또한 복잡도 측면에서는 FFD 방식이 BFD 방식보다 우수한 성능을 보이므로 동영상 비트열의 패킷화에는 FFD 방식이 가장 적절함을 알 수 있다.

## V. 결론

본 논문에서는 효과적인 실시간 인터넷 동영상 전송을 위해서 실제 인터넷 환경에서의 패킷화 파라미터에 대한 분석과 이를 토대로 기존의 운영체제에서 사용된 동적 기억장치할당 기법인 FF, BF, FFD, BFD 방식을 적용하여 동영상 비트열 패킷화 방식을 제안하였다. 본 논문에서는 제안 방식에 대한 근거로 운영체제에서 연구되어 왔던 동적 기억장치 할당 문제와 영상의 패킷화 과정과 동일함을 증명하였다. 실험을 통해서 FFD 및 BFD 패킷화 방식이 가장 효과적인 패킷화 방식임을 보였다. 또

한 제안 방식은 고화질의 실시간 인터넷 동영상 전송에 효과적임을 보였다. 실험을 통해 제안 방식은 기존의 방식에 비해 기존 방식에 비해 패킷화 부가정보를 최대 28.3% 감소함을 확인할 수 있었다. 제안 패킷화 방식을 수행할 때 필요한 연산량을 비교하였고, 전체 부호화 시간에 비해 패킷화 과정이 요구하는 연산량은 미미하지만 FFD 방식이 BFD 방식에 비해 낮은 연산량을 요구함을 보였다. 각 방식별로 연산량과 패킷화 성능을 종합적으로 비교하여 제안한 FFD 방식이 가장 우수한 방식임을 제시하였다.

참 고 문 헌

[1] B. A. Forouzan, TCP/IP protocol suite, McGraw Hill 2000.

[2] J. Davidson and J. Peters, Voice over IP fundamentals, Cisco press 2000.

[3] K.Y. Yoo, "New packetization method for error resilient video communications," LNCS, Springer-Verlag, vol. 3046, pp.329-338, May 2004.

[4] K.Y. Yoo, "Adaptive resynchronization marker positioning method for error resilient video transmission," IEE Electronics Letters, vol. 34, no. 22, pp. 2084-2085, Oct. 1998.

[5] X.-G. Liu, K.Y. Yoo, and K.-D. Seo, "A fast video mixing method for multiparty video conference," LNCS, Springer-Verlag, vol. 3656, no. 1, pp. 320-327, Sept. 2005.

[6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC1889, Jan. 1996.

[7] C. Zhu, "RTP Payload Format for H.263 Video Streams," IETF RFC2190, Sept. 1997.

[8] Y. Kikuchi, T. Nomura S. Fukunaga, Y. Matsui, and H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams," IETF RFC-3016, Nov. 2000.

[9] S. Wenger, M.M. Hannuksela, T. Stockhammer, and M. Westerlund, "RTP Payload Format for H.264 Video," IETF RFC2190, Feb. 2005.

[10] H.H. Kim and K.Y. Yoo, "Analysis on the effect of packet length on the RTP-based real-time video," IASTED Int'l Conf. on Internet, Multimedia Systems, and Applications (IMSI-2005), vol. 1, Aug. 2005.

[11] Paul R. Wilson, Mark S. Johnstone, Michael Neely, David Boles, "Dynamic Storage Allocation : A Survey and Critical Review," Proc. of Int'l Conf. on Workshop on Memory Management, pp.1-116, Sept. 1995.

[12] Silberschatz Abraham, Operating System Concepts, 6/E (Windows XP Update), Wiley, 2002.

[13] D. Wu, Y.T. Hou, B. Li, Wenwu Zhu, Ya-Qin Zhang, H.H. Chao, "An end-to-end approach for optimal mode selection in Internet video communication : theory and application," IEEE Journal on Selected Areas in Communications(JSAC), vol 18, pp.977-995, June 2000.

[14] Z.G. Li, C. Zhu, N. Ling, X.K. Yang, G.N. Feng, S. Wu, F. Pan, "A unified architecture for real-time video-coding systems," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, pp. 472-487, June 2003.

[15] K.Y. Yoo and K.-D. Seo, "A syntax-based mixing method for H.263 coded bitstreams," in Proc. of IEEE Int'l Conf. on Consumer Electronics, Las Vegas, USA, January 2005.

김 효 현 (Hyo-Hyun Kim)

정회원



2003년 2월 영남대학교 컴퓨터 공학전공 졸업  
 2005년 2월 영남대학교 정보통신공학과 석사  
 2005년 3월 현재 포항제철 PosData 연구원  
 <관심분야> 멀티미디어통신, 영

상압축, 영상처리



유 국 열 (Kook-Yeol Yoo)

정회원



1991년 2월 경북대학교 전자공  
학과 졸업

1993년 2월 한국과학기술원 전  
기및전자학과 석사

1998년 8월 한국과학기술원 전  
기및전자학과 박사

1998년 8월~2001년 3월 삼성전

자 중앙연구소 책임연구원

2001년 4월~현재 영남대학교 전자정보공학부 조교수

<관심분야> 멀티미디어통신, 영상압축, 영상처리