

# 협상비트 결정 알고리즘을 이용한 RTP 프로토콜의 동적필드 압축

정회원 김 경 신\*

## Dynamic Field Compression in RTP Protocols using Negotiation Bits Decision Algorithm

Kyung-shin Kim\* *Regular Member*

요 약

RTP프로토콜의 헤더 사이즈를 줄이는 여러 가지 헤더압축 방법으로 RFC2507, RFC3095 등의 기법이 있다. 이 논문에서는 협상비트결정 알고리즘을 제안하였다. 이 방식은 기존의 헤더압축 프로토콜보다 더 효율이 높은 압축방식을 가지고 있으며, 특히 Compressor와 Decompressor간 압축률과 안정성을 위해 압축비트를 협상하여 결정하는 방식을 사용하였다. 제안된 알고리즘의 타당성을 시뮬레이션을 통하여 증명하였다.

**Key Words** : Header compression, NCB, RTP

### ABSTRACT

Existent header compression scheme that can eliminated repeated field in header are RFC2507, RFC3095 scheme. This paper proposed a negotiation bits decision algorithm that have better performance than existent compression algorithm for header compression. Especially, proposed algorithm include new method that can negotiate compression bits length for better compression rates and stability between compressor and decompressor. Computer simulations proved validity of proposed algorithm using simulation package.

### 1. 서 론

최근의 다양한 통신서비스와 각종 인터넷 프로그램 들, 예를 들어 VoIP(Voice over IP), 인터랙티브 게임(interactive game), SNS 단문 메시지 패킷 등은 그들의 IP 패킷의 페이로드(payload)가 헤더의 사이즈보다 같거나 적게 구성 된다<sup>[1]</sup>.

RFC 3095로 대표되는 헤더압축 프로토콜은 이러한 헤더를 압축함으로써 비싼 무선 대역폭의 낭비를 막았으며 패킷 크기를 줄임으로써 패킷 스트림의 용량을 늘렸다. ROHC(RObust Header Compression) 알고리즘을 사용하면 IPv6에서 사용되고 있던 60바이트 헤더가 1바이트 내외로 줄어들게 됨으로써

300% 주파수 대역폭 이 개선된다<sup>[2]</sup>.

IETF는 ROHC(RFC 3095)를 헤더압축의 표준의 하나로 채택했다. 이 압축 프로토콜은 IP 프로토콜을 사용하는 무선 링크를 대상으로 한 것으로 RFC 3095는 패킷의 크기를 줄여서 스펙트럼 효율성과 패킷 손실과 지연을 줄임으로써 무선 네트워크에서의 상호 작용성을 증가시킬 것으로 보인다.<sup>[3]</sup>

특히 IP 패킷의 높은 오버헤드는 IP 네트워크 뿐만 아니라, 무선 링크에서의 주파수 효율도 떨어뜨리게 된다. 인터넷 프로토콜인 IPv4에 있어서는 모바일 핸드셋을 통해 IP 상에서 음성을 전달할 수 있지만 헤더에서 거의 66% 정도의 대역폭이 낭비된다. 이에 더해 IPv6 에서는 75%가 낭비된다.

\* 청강문화산업대학 교수 (update@ck.ac.kr)

논문번호 : 10041-1001 접수일자 : 2010년 10월 1일

RFC 3095로 대표되는 헤더압축 프로토콜은 헤더를 압축함으로써 비싼 무선 대역폭의 낭비를 막았으며 패킷 크기를 줄임으로써 패킷 스트림의 용량을 늘렸다. 견고한 헤더 압축(ROHC) 알고리즘을 사용하면 IPv6에서 사용되고 있던 60바이트 헤더가 1바이트로 줄어들게 됨으로써 300% 주파수 대역폭이 개선된다<sup>[4]</sup>. 본 논문에서는 RTP 헤더를 구성하는 여러 필드들 중에서 동적으로 일률적으로 증가하는 Dynamic 필드를 가장 높은 압축률로 압축하는 기법을 다음과 같이 제안하고 Compressor와 Decompressor간 협정에 의해 전송되는 패킷의 형태를 설계하였다.

## II. Dynamic 필드의 압축

다음 그림 1은 압축이 이루어지게 되는 ROHC/IP/UDP/RTP 에서의 헤더구조, 프로토콜 스택, 그리고 캡슐화(encapsulation)과정의 모습이다.

헤더를 구성하는 여러 필드들 중에서 고정 값으로 규칙적으로 증가하는 필드를 동적필드라 하며, RTP 헤더의 일련번호 SN과 타임스탬프 TS가 여기에 속한다.

본 논문에서는 이러한 동적필드를 가장 높은 압축률로 압축하는 기법을 다음과 같은 방법으로 제안한다.

첫째, 압축기와 복원기 사이에 송수신되는 패킷간의 SN과 TS의 증가분이 다음과 같이 일정하게 유지 된다고 가정할 때

- SN(Sequence Number) : 각 RTP 패킷 한 개씩 전송될 때마다 1씩 증가한다. 초기치는 랜덤하게 만들어진다.
- TS(Time Stamp) : TS\_STRIDE 값의 배수로 증가한다.

두번째, 압축되지 않은 SN과 TS는 총 48비트로

구성된다. 압축기와 복원기는 이 48비트를 몇 개의 비트로 압축할것인가를 먼저 결정한다.

이 과정은 다음 절에서 자세하게 설명하였다.

몇 비트로 압축할 것인가를 결정할 때, 트래픽의 서비스 형태를 기본으로 결정하는데, 0-32비트로 결정할 수도 있고, 아니면 압축률을 높이기 위해 변화되는 차이값만을 고려하여 0-12 비트로 결정할 수도 있다.

이렇게 결정된 압축비트수를 NCB(Negotiation Compression Bits)라고 부르기로 한다.

세번째, 압축기는 총 48비트의 SN과 TS를 앞에서 결정한 NCB n비트로 압축할 수 있고, 이 n비트를 복원기에게 송신한다. 복원기는 이 값을 이용하여 SN과 TS를 복원한다.

결론적으로 16비트 SN과 32비트 TS, 도합 48비트를 NCB n비트로 압축할 수 있게 된다.

그러나 이러한 RTP 헤더의 동적필드 중에서 일련번호 SN의 압축문제는 보다 더 신중하게 고려해야 한다.

SN은 패킷의 올바른 순서와 전송간의 패킷손실 방지를 위해 필요한 필드으로써, 이것이 손실 또는 손상될 경우 헤더를 복원 하는 것이 불가능하게 된다.

따라서 NCB 결정 알고리즘에서는 노드간의 상태나 전송되는 트래픽의 상태에 따라 SN의 압축여부를 결정하도록 하였다.

### 2.1 협상비트 결정 알고리즘

압축기와 복원기는 RTP로 운반되는 페이로드 트래픽의 특징과 회선의 에러율, 그리고 회선의 속도 등의 조건을 고려하여 협상 압축 비트인 NCB(Negotiation Compression Bits)를 결정한다.

본 논문에서는 압축기와 복원기 노드간의 타임스탬프에 대한 NCB(Negotiation Compression Bits, 협상 압축 비트수)를 결정하는 알고리즘을 다음과 같이 제안하였다.

인터넷상의 패킷은 잃어버릴 수도 있고 지연되거나 패킷순서가 뒤집혀 오는 경우가 있다. 이런 문제를 극복하기 위하여 RTP헤더에는 시간정보와 일련번호가 포함된다. 시간정보는 수신측에서 정확하게 20ms 마다 한 패킷을 재생하는데 사용되고, 일련번호는 순서대로 재생하면서 잃어버린 패킷을 계산하게 된다.

여기서 4바이트 타임스탬프는 RTP 데이터 패킷이 처음 발생될 때 만들어 지는데 랜덤한 초기치를 갖는다.

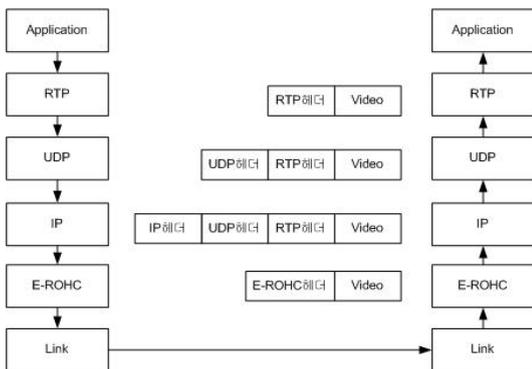


그림 1. RTP/IP 프로토콜의 캡슐화

그다음 패킷은 계속 증가되는 값을 갖는데, 어떤 데이터에 대해서는 증가되는 값을 갖지 않을 수도 있다. 특히 동기화와 지터계산을 위하여 일률적으로 증가되는 클럭을 가져야 한다. 클럭은 동기화를 할 수 있고 지터의 계산을 충분히 할 수 있도록 세밀하게 해야 한다.

MPEG 스트림의 경우, RTP를 사용해 비디오를 압축할 경우, 한 개의 패킷은 하나 이상의 그림을 포함하지 않으며, 한 개의 그림은 두 개 이상의 RTP 패킷으로 나누어 질 수도 있다. 이때 두 개로 나누어진 패킷들의 TS값은 같다.

RTP marker bit은 하나의 그림을 가지고 만들어진 패킷 중 마지막 패킷에서 1로 설정된다. 그리고 비디오 정보와 관련된 헤더압축 프로파일은 참조클럭으로 90kHz를 사용한다.

타임스탬프 TS는 다음 식과 같이 나타낼 수 있다<sup>5)</sup>.

$$TS = TS\_SCALE \times TS\_STRIDE + TS\_OFFSET \quad (1)$$

TS는  $TS\_SCALE \times TS\_STRIDE + TS\_OFFSET$ 으로 나뉘어 생각 할 수 있다.

TS\_STRIDE는 보폭이라는 말의 의미처럼 고정적으로 증가하는 값을 말한다. 오디오 데이터의 경우가 값은 160씩 증가한다. 오디오는 통상 8KHz 샘플링 주파수에 20ms의 주기를 가지므로 TS\_STRIDE는  $8000 \times 0.02 = 160$ 이 된다. 따라서 오디오데이터를 가지는 RTP패킷의 TS는 거의 160값씩 증가한다고 보면 된다.

MPEG-4 비디오의 경우 이 값은 3000씩 증가한다. MPEG은 코덱이 90KHz 샘플링에 초당 30프레임을 기본으로 하기 때문에 TS\_STRIDE는  $90000 / 30 = 3000$ 이 된다. 물론 무조건 3000씩 증가하는 것이 아니라 0 일수도 있고 3000보다 더 클 수도 있으므로 일률적으로 이 값을 인덱스처럼 송신 할 수는 없을 것이다<sup>6)</sup>.

식(1)에서 알 수 있듯이, TS\_STRIDE 값만 알 수 있다면 TS값은 물론, TS\_OFFSET, TS\_SCALE 값을 알 수 있게 된다.

TS\_SCALE은 RTP 패킷발생시점의 시간값이므로 여기에 TS\_STRIDE 값을 곱하여 TS 값이 형성된다. 따라서 TS\_OFFSET와 TS\_SCALE값은

$$TS\_OFFSET = TS \bmod TS\_STRIDE \quad (2)$$

$$TS\_SCALE = TS / TS\_STRIDE \quad (3)$$

값이 된다. 따라서 본 논문에서는 TS 필드를 압축하기 위하여, 적절한 방식으로 TS\_STRIDE값만을 전송하는 방식을 사용할 수 있을 것이다.

TS의 값이 항상 일정하게 증가한다면 이 값 자체가 정적필드가 되므로 송신할 필요가 없어지게 되나, TS 값의 증가분이 항상 일정하지는 않다는 문제가 있기 때문에, 송신노드는 일정분의 증가되는 값을 항상 보내야만 한다.

따라서 TS가 0일 때는 0이 송신되어야 하고 TS가 3000일 때는 3000이 전달되어야 한다.

여기서 TS 필드의 압축은 WDELSB(Wrapped around Differential Encoding LSB) 방식을 사용하였다. WDELSB 압축방식은 특정 필드값을 송신할 때, 전체 비트를 다 보내는 것이 아니라, 차이가 나는 내용만을 추출(differential)한 후 정해진 비트로 압축하는 방식이다. TS처럼 증가분이 거의 일정한 필드는 전체 비트를 다 송신 하는 것이 아니라, 의미를 정확히 전달할 수 있는 범위 내에서, 우측 몇 개의 비트만을 송신하면 된다.

같은 스트림 내에서도 계속해서 TS\_STRIDE값만 큼씩 변한다고 가정할 때, 이 TS\_STRIDE 값 범위 만큼씩의 WLSB 값만을 송신한다면, 전체 비트를 송신하는 것보다 많은 비트를 압축(절약)할 수 있게 된다.

그렇다면 TS값을 몇 비트로 압축(몇 비트의 LSB) 할 것인가를 결정하는 알고리즘을 다음과 같이 제안하였다. 본래의 TS 필드를 완벽히 복원 가능한 범위 내에서, 송신해야할 최소의 비트를 어떻게 구할 것인가를 다음과 같이 유도하였다.

먼저 원하는 트래픽의 TS\_STRIDE값에 대한 최소 압축 비트수 k를 구하는 함수  $g(V_{ref}, k)$ 를 다음과 같이 정의하였다.

$$g(V_{ref}, k) = \text{floor}(\log_2 TS\_STRIDE) + 1 \quad (4)$$

여기서  $V_{ref}$  는 참조주파수  $Freq_{ref}$ 를 가지는 트래픽을 나타내는 의미이고, k는 최소 압축비트를 나타낸다.

같은 스트림 내에서도 TS값은 변할 수 있기 때문에, NCB값의 범위를 최소값(min)과 최대(max)값만큼으로 정의해야 한다.

$V_{ref\_min}$ 는 해당 프레임에서 최소 TS\_STRIDE 일

때의 프레임이고,  $Vref_{max}$ 는 해당 프레임에서 최대  $TS\_STRIDE$  일때의 프레임이다.

NCB값을 구하는 식을 다음과 같이 나타낼 수 있다.

$$NCB = \max(g(Vref_{min}, k), (g(Vref_{max}, k))) \quad (5)$$

MPEG 비디오 프레임의 프레임 주파수가 최소값과 최대값 범위 내에서 바뀔 때마다  $TS\_STRIDE$ 값이 변할 수 있기 때문에 식(5)와 같이 NCB값을 그 범위에 맞도록 결정해 놓으면 원래의 TS필드를 복원할 수 있게 된다.

결론적으로 압축기로부터 32비트 타임스탬프값이 최대 NCB( $0 \leq n < 32$ )로 압축되어 송신된다.

이것은 식(5)에 대입하여 계산한 결과, 스트림이 길거나 큰 용량의 비디오 트래픽은 12비트에 근접하는 값으로, 오디오 파일 같은 작은 트래픽은 7비트 내외로 압축하면 된다.

## 2.2 알고리즘에 사용한 TS압축

앞 절에서 언급한 것처럼 본 논문의 압축방법은 WDELSB(Wrapped around Differential Encoding LSB)이다.

이 방식은 TS값을 송신할 때, 전체 비트를 다 보내는 것이 아니라, 차이가 나는 내용만을 추출(differential)한 후 정해진 비트(NCB)로 압축하는 방식이다. TS를 압축하기 위한 최소의 압축비트 NCB(bits)를 앞의 식(4)와 식(5)를 이용하여 구하였다.

그 다음 단계로,  $TS\_stride$  값을 구하기 위해 다음과 같이 참조클럭을 이용하여 계산하였다.

$$TS\_STRIDE = \frac{\text{비디오정보 참조클럭 (90KHz)}}{PCF(\text{Picture Clock Frequency})}$$

$$TS\_STRIDE = \frac{90000}{PCF(29.97Hz)} = 3003 \text{ in H.261}$$

$$TS\_STRIDE = \frac{90000}{25} = 3600 \text{ in H.263} \quad (6)$$

$$TS\_STRIDE = \frac{90000}{30} = 3000 \text{ in MPEG-4} \quad (7)$$

로 구할 수 있다.

또한 H.263 ver 2, MPEG-4 의 PCF는 각각 25Hz, 30Hz가 되는데, 이 값은 각각 3600, 3000의

배수로 TS값이 증가한다는 것이다.

TS의 압축 인코딩과정을 의사코드(pseudo code)로 표현한 압축 알고리즘이다. NCB비트가 정해지면 그 값만큼  $Right\_Bit\_Trim$  함수를 이용하여 추출하였다.

```

TS_Compress()
  IF  $TS_{n-1} > TS_n$  'Wrapped around Case
     $TS\_compress = Right\_Bit\_Trim(TS, NCB);$ 
    '우측 NCB비트 Trim
  else
     $TS\_compress = Right\_Bit\_Trim(TS_n - TS_{n-1},$ 
     $NCB);$ 
  end IF
END TS_Compress
    
```

이와 같이 압축기는 32-bit TS를 NCB(bits)로 압축하여 송신하게 되고, 이 값을 수신한 복원기는 다음과 같이 복원하게 된다.

다음은 이 복원 알고리즘의 의사코드(pseudo code)이다.

```

TS_Decompress()
  IF  $Left\_Bit\_Trim(TS_n, (Bits\ of\ TS) - NCB) ==$ 
   $0x000000$ 
     $TS = TS\_compress;$ 
  else
     $TS_n = TS_{n-1} + TS\_compress;$  'Differential
  end IF
   $TS\_compress\_ref = TS\_compress;$ 
  'Repair-상태를 위해 보관
END TS_Decompress
    
```

이 방법이 TS 필드에서 어느 정도의 압축률을 가졌는지 다음 식으로 나타내었다. RTP헤더의 TS 압축 비율을 다음과 같이 구할 수 있다.

$$C_{TS} = 1 - \frac{TS_{proposed}}{TS} \quad (8)$$

여기서, TS는 압축하지 않을 때의 패킷헤더의 사이즈(byte)이고,  $TS_{proposed}$ 는 압축한 TS의 사이즈이다.

### III. Simulation

#### 3.1 시뮬레이션 환경

시뮬레이션을 위하여 압축기와 복원기로 구성된 2개의 송수신 노드간의 통신만을 고려하였다. 송신노드와 목적지 노드는 높은 패킷 에러율과 긴 RTT를 가지는 무선 링크로 연결되었다고 가정하였다.

비디오 RTP패킷들은 송신지 노드에서 발생되고 압축된 후 무선 데이터채널을 통해 복원기가 내장된 목적지 노드로 보내진다.

제안된 기법의 타당성을 검증하기 위하여 다음과 같은 기본 가정을 하였다. forward 데이터채널 상에서 전송된 패킷은 손실될 수 있고, 패킷 에러의 특별한 형태가 없다. 패킷의 순서는 압축기와 복원기 사이에서 유지된다.

즉 수신기는 항상 전송 측에서 보낸 순서대로 패킷을 받는다. 피드백 채널 상에서 엄격한 지연이나 에러요구사항이 없다.

제안한 알고리즘은 RTT의 어떤 특별한 분포를 가정하지 않는다. 동적으로 RTT에 대한 변화에 적응할 수 있으며, 헤더압축은 인터리빙과 채널코딩에 영향을 미치지 않는다.

이 논문에서는 제시한 압축 기법에 대한 검증은 하기위해 복원에러율(x 개의 패킷당 1 패킷 복원에러), 처리율(throughput)과의관계를 시뮬레이션 하였고 성능을 평가하였다.

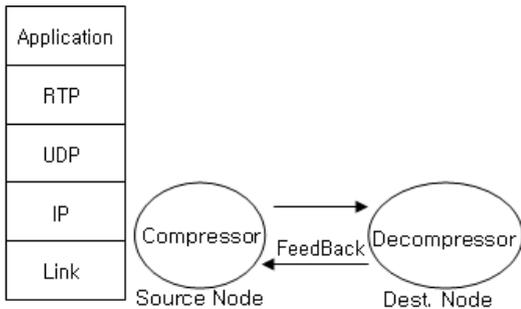


그림 2. 시뮬레이션 환경

#### 3.2 시뮬레이션 모델

기존의 헤더압축 프로토콜인 IPHC나 ROHC와 비교했을 때 이 논문에서 제안한 헤더압축의 큰 차이점은 RTP 헤더의 일련번호 SN과 타임스탬프 TS의 압축률 차이로 볼 수 있다. 다음과 같은 헤더의 압축률을 가지고 시뮬레이션을 수행하였다.

TS를 n 비트로 압축하였을 때, 압축률을 구해보

면 다음 식(9)와 같다.

$$C_{TS} = 1 - \frac{H_{compress}}{H_{TS}} \quad (9)$$

TS는 총 32비트이다. 이것을 NCB 8비트로 압축하였을 때의 압축률을 구하면 다음과 같다.

$$C_{SVTS} = 1 - \frac{8bits}{32bits} = 0.75$$

BCB와 extended\_n 비트를 이용한 헤더의 압축률과 그 성능을 비교하기위해 식 (9)를 이용하여 표 1과 같은 실험모델을 구하였다.

여기서 표 1의 맨 아래쪽의 변화범위는 w-LSB에서의 윈도우 사이즈를 나타낸다.

위 표는 TS의 압축된 비트에 대해 압축률과 변화범위를 나타내었다. 이때 고려해야할 요소가 BER (Bit Error Rate)이다. BER은 압축비트수와 처리율, 지연시간 등 압축통신이 일어나는 이 프로토콜의 성능에 큰 영향을 미치는 요소로서, 특히 압축 비트수를 결정하는데 큰 영향을 미친다.

BER이 큰 경우, 즉, 비트 압출에러율이 높은 경우, NCB 압축비트 길이가 길어질 수록 에러 복구율이 높아지고, NCB 압축비트수를 작게 할 경우에는, 에러에 대한 복구율이 낮아진다.

표 1. 시뮬레이션 모델

Number of NCB (Bits)	8 bit	10 bit	12 bit	14 bit	...	n bits
Compression Rate	0.75	0.6875	0.625	0.5625	...	$1 - \frac{n\ bits}{32\ bits}$
Change Scope	[0, 255]	[0, 1023]	[0, 4095]	[0, 16383]	...	[0, $2^n - 1$ ]

#### 3.3 성능평가

실험을 위해 복원에러율을 다음과 같이 모델링 하였다. 이 단계에서는 순수한 복원기 측에서의 복원에러율만을 고려하였다. 그래서 송수신 단계에서의 패킷 손실은 없다고 가정하였다. 복원 에러율은 몇 개의 패킷을 처리하는 중에 1개의 패킷 에러가 발생하는 가로 나타낼 수 있는데, 여기서는 정상적인 경우를,  $1000^{-1}$ 로 표기하였으며 이것은 1000개의 패킷당 1개의 에러가 발생하는 것으로 설정하였다.

따라서 다음의 실험결과를 나타내는 그림에서 복원에러를 나타내는 그래프의 X축은  $1000^{-1}$ ,  $900^{-1}$ ,

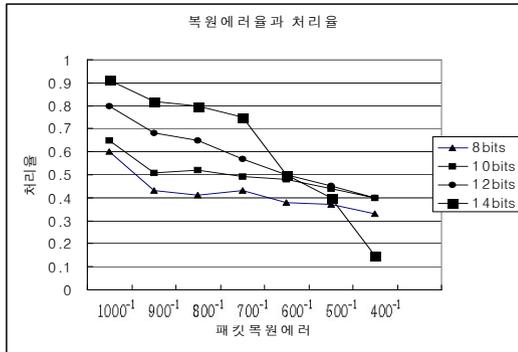


그림 3. 복원 에러율과 처리율

800<sup>-1</sup>, 700<sup>-1</sup>, 600<sup>-1</sup>, 500<sup>-1</sup>, 400<sup>-1</sup>로 정하였고 처리율을 나타내는 Y축은 0 - 1까지를 범위로 정하였다.

#### IV. 결론

제한된 대역폭을 갖는 무선네트워크에서 비디오 스트림의 효율적인 전송은 매우 중요하며, 적시성을 갖는 다양한 텍스트나 저용량 데이터의 고속전송도 매우 중요한 사항이다. 이 경우 RTP의 헤더사이즈는 페이로드에 비해 너무 큰 문제로 인해 대역폭의 낭비와 적시성의 데이터 전송을 해치는 요인이다. 이 논문에서는 프로토콜의 안정성과 압축률에 따라 송신 양단간에 전송 비트를 협상으로 결정하여 헤더를 압축하는 알고리즘을 제안하였다.

제안된 알고리즘의 성능을 검증하기 위해 AweSim3.0 패키지를 사용하여 시뮬레이션을 수행하였다. 이러한 실험을 통해, 제안된 알고리즘으로 32비트 타임스탬프를 다양하게 압축하였을 때 복원성공률을 포함한 성능분석을 할 수 있었다. 처리율이 가장 높을 때가 해당 노드의 효율성이 가장 높다는 것을 의미한다.

이 실험 결과를 보면, 압축률을 높여서 NCB가 14일 경우, 복원 에러율이 가장 적은 1000패킷 시점에서 처리율이 가장 높게 나오는 것을 볼 수 있다. 그러나 이 경우 복원 에러율이 500<sup>-1</sup>인 지점부터 급격히 처리율이 감소하여 400<sup>-1</sup>일 때는 기능이 정지될 정도의 수치가 나오는 것을 볼 수 있다. 즉, 압축률이 높은 경우 일정 수준 이상의 복원 에러율이 유지되어야만 한다는 것을 알 수 있다.

이에 반하여 압축률을 8bit 값으로 낮추어 실험한 경우는, 에러율의 낮고 높음에 거의 영향을 받지 않고 0.4 정도의 평균값을 지속적으로 유지하는 것을 알 수 있다. 이상과 같이 본 논문에서 제안한 협

상비트 결정알고리즘을 활용한 동적필드 압축방식의 타당성을 증명 할 수 있다.

#### References

- [1] www.efnet.com “An Introduction to IP Header Compression”, efnet White paper, Feb. 2004.
- [2] Stephen Casner, Van Jacobson, “Compressing RTP/UDP/IP Headers for Low-Speed Serial Links”, RFC 2508, February 1999.
- [3] C. Bormann, Ed., “Robust Header Compression(ROHC)” RFC 3095, June 2001.
- [4] H. Schulzrinne et al., “RTP: A Transport Protocol for Real Time Applications,” IETF RFC 1889, Jan. 1996.
- [5] Haipeng Jin, Raymond Hsu, Jun Wang, “Performance comparison of Header Compression schemes for RTP/UDP/IP Pckets”, WCNC2004/IEEE Communications Society, pp1691, 2004
- [6] Michael Degermark, Bjorn Nordgren, Stephen Pink, “IP Header Compression”, RFC 2507, FEB 1999.
- [7] 서영건, “멀티미디어 통신”, 인솔미디어, pp281
- [8] D.Hoffman, G.Fernando, “RTP Payload Format for MPEG1/MPEG2”, RFC2250, Jan. 1998.
- [9] V.kikuchi, T. Nomura, “RTP Payload Format for MPEG-4 Audio/Visual Streams”, RFC3016, Nov. 2000.
- [10] Emre Ertekin, “Internet Protocol Header Compression, Robust Header Compression and Their Applicability in the GIG” IEEE Communications Magazine Nov 2004.

김경신 (Kyung-shin Kim)

정회원



1986년 금오공과대학교 전자공학과 학사

1993년 연세대학교 대학원 전자공학과 석사

2007년 경희대학교 대학원 컴퓨터공학과 박사

2000년~현재 창강문화산업대학 교수

<관심분야> 근거리전산망, 라우팅프로토콜, RTP/IP 헤더압축