Journal of The Korea Society of Computer and Information
Vol. 18, No. 10, October 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.10.055

RTP 프로토콜에서 Time Stamp필드의 압축을 위한 향상된 협상비트 결정 알고리즘

김 경 신*

Time Stamp Compression in RTP Protocols using Enhanced Negotiation Bits Decision Algorithm

Kyung-shin Kim*

요 약

패킷헤더의 중복된 필드를 제거하여 헤더의 사이즈를 줄이는 방법을 헤더압축이라고 할 때, 헤더에서 연속된 패킷간 일정 수치만큼씩 고정적으로 증가되는 Dynamic필드를 어떻게 고효율로 압축할 수 있느냐가 헤더압축의 중요한 이슈라고 할 수 있다. RTP프로토콜의 헤더 사이즈를 압축하는 기존의 방법으로 RFC2507, RFC3095 그리고 ROHC와 E-ROHC 등의 기법이 있다. 본 논문에서는 RTP 패킷의 Dynamic필드인 TS필드를 BCB(Basic Compression Bits) 기본 비트로 압축하거나 또는 NCB(Negotiation Compression Bits, BCB + 추가적인 비트) 비트로 압축하는 기존 방법보다 향상된 방법(Enhanced Method)을 제안하였다. 여기에서 제안한 새로운 향상된 헤더압축 기법의 성능을 검증하기위해, 제안한 방식의 비디오패킷을 대상으로 Visual SLAM을 사용하여 시뮬레이션 하였다.

▸ Keywords : 헤더압축, 협상비트, 기본비트

Abstract

The important issue in header compression would be how to compress the dynamic field increasing constantly between consecutive packets in the head of IP wireless networks. Existent header compression scheme that can eliminated repeated field in header are RFC2507, RFC3095 and E-ROHC scheme. In this paper, I propose a new method of compressing TS fields, which are the Dynamic fields of the RTP packet, into BCB (Basic Compression Bits) basic bits or NCB (Negotiation Compression Bits, BCB + additional bits) bits. In order to verify the proposed header

•제1저자 겸 교신저자 : 김경신
•투고일 : 2013. 10. 8, 심사일 : 2013. 10. 13, 게재확정일 : 2013. 10. 19.
* 청강문화산업대학교 컴퓨터공학과(Dept. of Mobile Security, ChungKang College of Cultural Industrial)

compression method, I have simulation about proposed video packets of IP wireless networks. using Visual SLAM.

▸ Keywords : Header compression, NCB, RTP

I. 서 론

최근의 스마트 모바일환경에서 다양한 통신서비스와 각종 인터넷 프로그램 들, 예를 들어 바이버나 보이스톡 같은 VoIP, 인터랙티브 게임(interactive game), 트위터나 밴드 같은 SNS 단문 메신저 패킷 등은 그들의 IP 패킷의 페이로드(payload)가 헤더의 사이즈보다 같거나 적으므로 RFC 3095로 대표되는 헤더압축 프로토콜은 이러한 헤더를 압축함으로써 비싼 무선 대역폭의 낭비를 막았으며, 패킷 크기를 줄임으로써 송수신 효율을 향상시켰다. IETF는 RTP/IP헤더압축의 표준으로 ROHC(RFC 3095)를 채택했다. IP 프로토콜을 사용하는 무선 링크를 대상으로 한 RFC 3095는 패킷의 크기를 줄여서 스펙트럼 효율성을 높이고 전송지연을 줄임으로써 무선 네트워크에서의 처리율을 증가시킬 것으로 보인다[1]. 견고한 헤더 압축(ROHC) 알고리즘을 사용하면 IPv6에서 사용되고 있던 60바이트 헤더가 1바이트로 줄어들게 됨으로써 300% 주파수 대역폭 이 개선된다[2]. 기존 방식의 핵심은 RTP 헤더를 구성하는 여러 필드들 중에서 일률적으로 증가하는 Dynamic 필드인 32비트 타임스탬프를 가장 높은 압축률로 압축하고 통신 양단간에 전송 윈도우사이즈를 결정하는 알고리즘을 제안하는 것이었다[3]. 이 논문에서는 이 과정에서 발생한 협상비트 알고리즘의 문제점을 개선하고 수정하여 ROHC알고리즘의 성능향상을 위한 향상된 협상비트 결정 알고리즘을 제안하는 것이다.

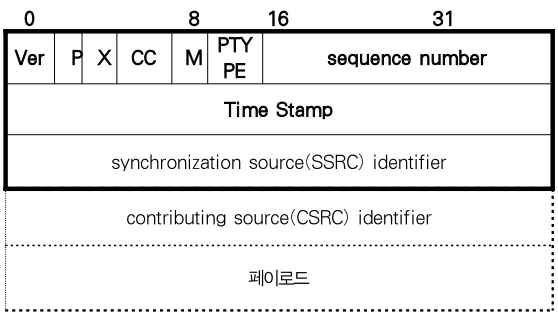

그림 1. RTP 패킷 포맷
Fig1. RTP Packet Format

II. Time Stamp 필드 압축

2.1 NCB협상비트

그림 1은 압축의 대상이 되는 RTP패킷의 포맷을 보여준다. 헤더를 구성하는 여러 필드들 중에서 고정 값으로 규칙적으로 증가하는 필드를 동적필드라 하며, 이중에서 RTP 헤더의 Time Stamp를 대상으로 한 압축방법의 개선이 이 논문의 주제이다. 압축되지 않은 Time Stamp는 32비트로 구성된다. 압축기와 복원기는 이 32비트를 몇 개의 비트로 압축할 것인가를 먼저 결정한다. 이렇게 결정된 압축비트수를 NCB(Negotiation Compression Bits)라고 부르기로 한다. 압축기는 32비트의 TS를 앞에서 결정한 NCB n비트로 압축할 수 있고, 이 n비트를 복원기에게 송신한다. 복원기는 이 값을 이용하여 Time Stamp를 복원한다. 결론적으로 32비트 Time Stamp를 NCB n비트로 압축할 수 있게 된다.

2.2 기존의 협상비트 결정 알고리즘

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[4]. 비디오 정보와 관련된 헤더압축 프로파일은 참조클럭으로 90kHz를 사용한다. 타임스탬프 TS는 다음 식과 같이 나타낼 수 있다[5].

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

TS는 초기값에 TS_STRIDE값을 곱한 형태로 증가한다. TS_STRIDE는 보폭이라는 말의 의미처럼 고정적으로 증가하는 값을 말한다[6][7].

TS_SCALE은 RTP 패킷발생시점의 시간값이므로 여기에 TS_STRIDE 값을 곱하여 TS값이 형성된다. 따라서 TS_OFFSET와 TS_SCALE값은

TS_OFFSET=TS mod TS_STRIDE (2)

TS_SCALE=TS/TS_STRIDE (3)

값이 된다[8][9]. 이렇게 고정적으로 변하는 TS_STRIDE 값을 이용하여 다음과 같은 식으로 NCB값을 구할 수 있다.

$$NCB = \max(g(Vref_{\min}, k), (g(Vref_{\max}, k)) \quad (4)$$

Vrefmin는 해당 프레임에서 최소 TS_STRIDE 일때의 프레임이고, Vrefmax는 해당 프레임에서 최대 TS_STRIDE 일때의 프레임이다. 이것에 대입하여 계산한 결과, 스트림이 길거나 큰 용량의 비디오 트래픽은 12비트에 근접하는 값으로, 오디오 파일 같은 작은 트래픽은 7비트내외로 압축하면 된다.

III. 향상된 협상비트 결정 알고리즘

기존의 압축기와 복원기 노드간의 타임스탬프에 대한 NCB(Negotiation Compression Bits, 협상 압축 비트수)를 결정하는 알고리즘을 앞 절의 식(1)로 나타낼 수 있다. 이것이 유도되기 위하여 최종 압축비트 k를 구하는 함수는 원하는 트래픽의 TS_STRIDE값에 대한 최소 압축 비트수 k를 결정하는 함수 Decision(Vref, k)를 다음 식(5)로 정의한 것으로부터 출발하였다.

$$Decision(Vref, k) = floor(\log_2 TS\_STRIDE) + 1 \quad (5)$$

3.1 향상된 알고리즘의 유도

본 논문에서 제안하는 것은 RTP페이로드의 영상품질을 해치지 않는 범위내에서 최대 압축으로 통신속도를 보장하려는 것이다. 이때 중요한 것은 TS_STRIDE값을 표현해 주기 위한 비트수이므로 Decision(Vref, k)함수는 주로 위 식과 같이 최소비트수의 추출 알고리즘으로 이루어진다. 물론 +1 비트를 하여 Floor 정수화 손실에 대한 보정을 하였다. 여기에서 32비트 타임스탬프의 특성을 고려해야한다. 타임스탬프는 같은 스트림 내에서도 계속해서 TS_STRIDE값만큼씩 변한다고 가정할 때, 이 TS_STRIDE 값 범위 만큼씩의 WLSB 값만을 송신한다면, 전체 비트를 송신하는 것보다 많은 비트를 압축(절약)할 수 있게 된다. Time Stamp 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)하여 송신하게 된다. 자연스럽게 송신 패킷이 압축되는 것이다. 그렇다면 TS값을 몇 비트로 압축(몇 비트의 LSB)할 것인가를 결정하는 알고리즘을 산출하기 위해 페이로드 데이터 타입을 고려한 최소값과 최대값을 고려하였다. MPEG 비디오 프레임의 프레임 주파수가 최소값과 최대값 범위 내에서 바뀔 때마다 TS_STRIDE값이 변할 수 있기 때문에 식(4)와 같이 NCB 값을 그 범위에 맞도록 결정해 놓으면 원래의 TS필드를 복원할 수 있게 된다. 압축기와 복원기는 RTP로 운반되는 페이로드 트래픽의 특징과 회선의 에러율, 그리고 회선의 속도 등의 조건을 고려하여 협상 압축 비트인 NCB(Negotiation Compression Bits)를 결정한다.

여기서 Vref 는 참조주파수 Freqref를 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. 먼저 최소값은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} Min\,Bits &= Decision(Vref_{\min}, k) \\ &= floor(\log_2 TS\_STRIDE_{\min}) + 1 \end{aligned} \quad (6)$$

같은 방식으로 최대값일 경우는 다음과 같다.

$$\begin{aligned} Max\,Bits &= Decision(Vref_{\max}, k) \\ &= floor(\log_2 TS\_STRIDE_{\max}) + 1 \end{aligned} \quad (7)$$

위와 같이 스트림 내에서의 최소값과 최대값을 계산하여 이중에서 하나의 결과로 유도한 것이 기존의 방식인 식(4)이다. 결론적으로 압축기로부터 32비트 Time Stamp값이 최대 NCB($0 \leq n < 32$)로 압축되어 송신된다.

그러나 식(4)를 고려하여 구현을 할 때 약간의 정리가 필요했다. 즉, 압축할 최소 비트수와 최대비트수를 Max 함수로 결정하는 방식은 그 자체에 모순을 담고 있을 수 있다. 위 식(4)와 같은 경우 Max함수에 의한 결정은 당연히 Max값이 선택될 것이기 때문이다. 따라서 이 논문에서 제안하는 향상된 협상비트 결정 알고리즘은 다음과 같이 수정 될 수 있다.

$$Decision(Vref_{\min}, k) = floor(\log_2 TS\_STRIDE) + 1$$

$$Decision(Vref_{\max}, k) = floor(\log_2 TS\_STRIDE) + 1$$

$$NCB = Decision(Vref_{\max}, k)) \quad (8)$$

3.2 향상된 알고리즘을 사용한 TS압축

TS_Stride 값을 구하기 위해 다음과 같이 참조클럭을 이용하여 계산하였다.

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

H.261에서는

$$TS\_STRIDE = \frac{90000}{PCF(29.97Hz)} = 3003$$

이고, H.263에서는

$$TS\_STRIDE = \frac{90000}{25} = 3600$$

이다. MPEG-4인 경우는

$$TS\_STRIDE = \frac{90000}{30} = 3000$$

가 된다.

또한 H.263 ver 2, MPEG-4 의 PCF는 각각 25Hz, 30Hz가 되는데, 이 값은 각각 3600, 3000의 배수로 TS값이 증가하게 된다. TS의 압축 인코딩 과정을 의사코드(pseudo code)로 표현한 압축 알고리즘이다. NCB비트가 정해지면 그 값만큼 Right_Bit_Trim 함수를 이용하여 추출하였다.

```
TS_Compress()
    IF TSn-1 〉 TSn   'Wrapped around Case
        TS_compress = Right_Bit_Trim(TS, NCB);
                '우측 NCB비트 Trim
    else
        TS_compress = Right_Bit_Trim(TSn - TSn-1, NCB);
    end IF
END TS_Compress
```

다음은 이 복원 알고리즘의 의사코드(pseudo code)이다.

```
TS_Decompress()
    IF Left_Bit_Trim(TSn, (Bits of TS) - NCB) ==
0x00000
        TS = TS_comnpress;                          else
       TSn = TSn-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.

$$C_{TS} = 1 - \frac{TS_{proposed}}{TS} \quad (9)$$

여기서, TS는 압축하지 않을 때의 패킷헤더의 사이즈(byte)이고, TSproposed는 압축한 TS의 사이즈이다.

3.3 윈도우사이즈의 협상

압축기는 복원기의 ACK나 NAK 수신 없이 압축된 패킷을 송신한다. 세션이 연결될 때 압축기와 복원기는 윈도우사이즈 WD_send(Packets)를 결정한다. 그리고 최초 결정된 WD_send값은 수시로 압축기와 복원기 간의 상호 요구에 의해 재협상한다. 제안하는 윈도우사이즈 WD_send (packets) 산출수식은 다음과 같다.

$$WD\_send(packets) = floor(\frac{BER\_rev\_valuevalue}{MTU}) \quad (10)$$

여기서 MTU는 기 설정되어있는 Maximum Transmission Unit 이고, BER_rev_value는 송수신단에서 협상한 BER값의 -1승 값을 기본으로 가감한 값이다.

이 링크의 BER(Bit Error Rate)이 10-5이면, (10-5)-1 은 105=100000이 된다.

이때 MTU가 4000비트라면, WD_send(packets)는 25(packets)가 된다.

이 값을 프레임의 최대 전송단위 MTU로 나눈 값을 기본적인 윈도우사이즈 WD_send(packets)로 정의하였다. 이 과정을 다음과 같이 자세히 설명하였다.

① 압축기는 복원기에게 BER_request라는 제어신호를 보낸다. 이 패킷은 WD_send 값의 협상을 요구하는

패킷으로 현재 이 링크의 BER값이 들어가 있다.

② 복원기는 수신한 BER_request 값을 체크한 후, 적당히 가감한 값을 확인 송신한다.

$$BER\_rev\_value = BER-1 \quad (11)$$

③ 압축기는 수신한 BER__rev_value 값을 이용하여 MTU값과의 비율을 구한 후 WD_send(packets)값을 확정한다. 이 값을 BER_reconfirm 제어패킷을 이용하여 복원기에 송신한다. 압축기와 복원기 간의 협상 Call 플로우는 다음과 같다.

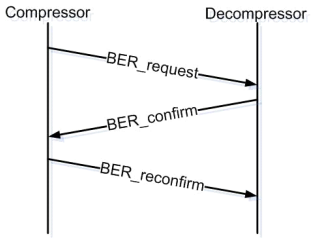

그림 2. 윈도우사이즈 협상
Fig 2. Negotiation of Window size

Ⅳ. Simulation

4.1 시뮬레이션 환경

제안된 알고리즘으로 32비트 Time Stamp 필드를 몇 가지 Case로 압축하고 복원하는 과정을 나타내었고 이를 Visual SLAM3.0을 이용하여 시뮬레이션 하였다. 이것을 통해 주어진 Case에서의 에러율과 처리율을 가지고 성능분석을 할 수 있었다. 이 논문에서 제안한 NCB최소비트 결정알고리즘과 송수신 윈도우사이즈 협상 알고리즘은 기존의 ROHC, E-ROHC등에 적용할 수 있다. 특히 E-ROHC에서 사용한 알고리즘을 수정한 것이므로 제시한 압축과 복원 기법에 대한 검증을 하기 위해 E-ROHC기법의 시뮬레이션과 유사한 방식으로 모델링하였고 복원에러율(x 개의 패킷당 1 패킷 복원에러), 처리율(throughput)을 중심으로 시뮬레이션 하였다. 또한 송신노드와 목적지 노드는 높은 패킷 에러율과 긴 RTT를 가지는 무선 링크로 연결되었다고 가정하였다.

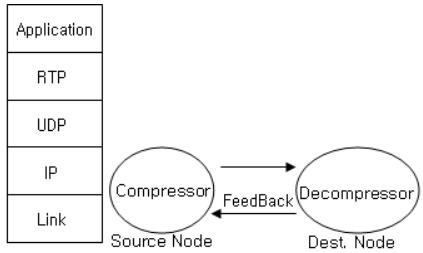

그림 3. 시뮬레이션 환경
Fig 3. Simulation Environment

4.2 시뮬레이션 모델

다음과 같은 헤더의 압축률을 가지고 시뮬레이션을 수행하였다. TS를 n 비트로 압축하였을 때, 압축률은 위에서 언급한 것처럼 식(9)와 같다.

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

TS는 총 32비트이다. 예를 들어, 이것을 NCB를 5비트라고 했을 때의 압축률을 구해보면 다음과 같다.

$$C_{SNTS} = 1 - \frac{5bits}{32bits} = 0.84$$

헤더의 압축율과 그 성능을 비교하기 위해 식 (9)를 이용하여 표 1과 같은 실험모델을 구하였다.(E-ROHC와 동일환경으로 구성함)

여기서 표 1의 맨 아래쪽의 변화범위는 w-LSB에서의 윈도우사이즈를 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.

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

표 1. 시뮬레이션 모델
Table 1. Simulation Model

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.

4.3. 성능평가

실험간 송수신 단계에서의 패킷 손실은 없다고 가정하였다. 그리고 복원 에러율은 몇 개의 패킷을 처리하는 중에 에러가 발생하는 가로 나타내는데, 여기서는 정상적인 경우를, 1000-1로 표기하였으며 이것은 1000개의 패킷당 1개의 에러가 발생하는 것으로 설정하였다. 실험결과의 그래프에서 X축은 패킷복원 에러율을 나타낸다. 1000-1, 800-1에서 400-1까지 실험하였다. Y축은 실험환경의 처리율을 표시하였고, 0에서 1까지 나타냈다.

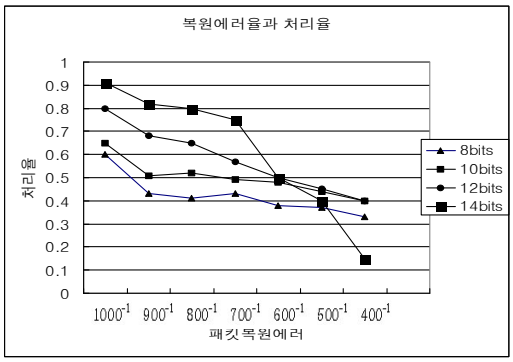

그림 4. 복원에러율과 처리율
Fig4. Error Recovery and Throughput

V. 결론

제안한 알고리즘을 시뮬레이션을 통해 분석한 결과 패킷 복원 에러율이 처리율에 미치는 영향을 볼 수 있었다. 에러율이 거의 없는 환경에서는 NCB값의 숫자가 영향을 크게 미치지 않았다. 그러나 에러율이 높을때는 NCB의 비트수에 관계없이 처리율이 낮은 것을 볼 수 있었다. 또한 NCB를 14까지 올렸을 때 복원 에러율이 가장 적은 1000패킷 시점에서 처리율이 가장 높았다. 그러나 이 경우 복원 에러율이 700-1인 지점부터 급격히 처리율이 감소하여 600-1일 때는 처리율이 0.4이하가 되는 것을 볼 수 있다.

E-ROHC의 경우와 마찬가지로 압축률이 높은 경우에는 에러율이 큰 변수로 작용한다. 망의 에러율이 800-1일 경우까지는 이 방식을 사용했을 때 높은 처리율을 얻을 수 있다는 것을 알 수 있다.

그러나 그 이하의 에러율을 가지는 환경에서는 NCB를 낮추어 압축을 하는 것이 바람직한 것으로 분석된다. 또한 NCB값이 낮을 경우에는 망의 에러율에 의한 영향을 비교적 덜 받는 것을 알 수 있었다.

참고문헌

[1] C. Bormann, Ed., "Robust Header Compression(ROHC)" RFC 3095, Jun. 2001.

[2] H. Schulzrimme et al., "RTP: A Transport Protocol for Real Time Applications," IETF RFC 1889, Jan. 1996.

[3] Kyung-shin, Kim. "Dynamic Field Compression in RTP Protocols using Negotiation Bits Decision Algorithm", Journal of Korea Communications Society, Vol. T. Jun. 2010,

[4] Sung-Yeol Yun, Seok Cheon Park. "A study on Header Compressiom Algorithm for the Effective Multimedia Transmission over Wireless Network." Journal of Korea Ocean Information Communication, Nov. 2009, Vol. 14. no.2.

[5] Haipeng Jin, Raymond Hill, Jun Wang, "Performance comprasion of Header Compression schemes for RTP/UDP/IP Pckets", WCNC2004/IEEE Communications Society, pp1691, 2004

[6] www.efnet.com "An Introduction to IP Hader Compression", effnet White paper, Feb. 2004.

[7] Michael Degermark, Bjorn Nordgren, Stephen Pink, "IP Header Compression", RFC 2507, FEB. 1999.

[8] Seo young gun. "Multimedia communications", Insol Media, pp281

[9] Stephen Casner, Van Jacobson," Compressing RTP/UDP/IP Headers for Low-Speed Serial Links", RFC 2508, Feb. 1999.

[10] D.Hoffman, G.Fernando, "RTP Payload Format for MPEG1/MPEG2", RFC2250, Jan. 1998.

저 자 소 개

김 경 신

1986: 금오공과대학교
전자공학과 공학사.
1993: 연세대학교
전자공학과 공학석사.
2007: 경희대학교
컴퓨터공학과 공학박사
현 재: 청강문화산업대학교
모바일 보안과 교수
관심분야: 컴퓨터공학
Email : update@ck.ac.kr