

이동성 IP기반 네트워크에서 헤더 압축기법을 이용한 효율적인 비디오 트래픽 관리기법

정회원 강문식*

An Efficient Video Traffic Management Scheme using Header Compression in IP-Based Mobile Networks

Moon-sik Kang* *Regular Member*

요 약

본 논문에서는 이동성 IP기반 네트워크에서 효과적인 헤더압축 기법 및 압축상태 복구기법을 이용한 효율적인 비디오 트래픽 관리기법을 제안한다. 이동성 무선 링크의 대역폭은 물리매체와 무선 구간내 데이터 전송에 따른 주파수의 사용에 제약을 받는다. 또한 셀룰러 액세스 링크의 고비용과 링크손실 특성 등이 IP기반 네트워크의 성능에 중요한 요소가 된다. 그러므로 자원을 효율적으로 사용하는 것이 필요하며, 셀위치에 따른 메시지 교환 절차에 대하여 기술한다. 이를 위해 효율적인 헤더압축기법이 요구된다. 따라서 본 연구에서는 지금까지 제안된 기법을 분석하여, 헤더압축 및 이동호스트의 핸드오프 발생시 신속하게 압축상태를 복구하는 기법에 대한 연구를 수행하였다. 제안된 기법은 패킷손실을 줄이는 압축 컨텍스트 값을 인지하도록 하여 신속하여 압축상태로 복원되도록 하였다. 제안된 기법의 성능을 분석하기 위해서 컴퓨터 시뮬레이션을 수행하고, 그 결과에 의해서 제안된 기법의 성능이 개선됨을 입증하였다.

ABSTRACT

In this paper, we propose the efficient video traffic management scheme using header compression and the fast recovery schemes of compression states under handover operations in IP-based networks. The bandwidth of wireless links will probably be limited due to properties of the physical medium and regulatory limits on the use of frequencies for radio communications. Also, both high cost of cellular access link and link loss properties are major two factors for the transmission of the packets over IP-based Networks. Therefore, It is necessary to utilize the bandwidth of air interface efficiently. We discuss the exchange procedures of message for change of cell positions. We have studied the fast recovery scheme of compression state by analyzing the previous methods. We will show that indication of compression context values preventing from packet losses can provide the fast recovery of compression state. Computer simulations show that the proposed scheme has better performance than the previous one.

I. 서론

패킷 네트워크를 통해서 음성이나 비디오 같은 실시간 데이터를 전송할 때 사용되는 RTP(Real

time protocol)에서 전송되는 패킷은 IP 헤더(20 octets), UDP(User datagram protocol) 헤더(8 octets), RTP(12 octets) 등으로 캡슐화 되어, 헤더 부분의 크기가 40 바이트를 차지하게 된다^[1,5]. IPv6

* 강릉대학교 전자공학과 네트워크 연구실(mskang@kangnung.ac.kr)

논문번호 : K01105-0326, 접수일자 : 2001년 3월 26일

※ 본 논문은 정보통신부에서 지원하는 2001년도 대학기초연구지원사업으로 수행된 결과임

를 사용할 경우에는 60 바이트의 크기를 갖게 되며^[4], 또한 여러 가지 확장 헤더들은 라우팅 정보나 인증 정보 등 추가정보를 제공하기 위해서는 기본 헤더에 추가된다. 무선링크 상에서 인터넷 기술의 효율성을 감소시키는 요소로는 IPv6의 사용과, 이동성 지원을 들 수 있다. Mobile IP의 경우 이동호스트가 위치해 있는 네트워크로 패킷을 보내기 위해서 Home Agent가 사용하게 되는 터널링 기법은 원래의 패킷에 추가적으로 패킷을 캡슐화해야 하기 때문에 100바이트 이상의 헤더를 무선링크 상에서 전송해야 하는 부담이 생긴다^[6]. RF 스펙트럼이 전체 시스템 운용비용의 30%를 차지하는 경우, 무선링크 상에서 패킷을 전송했을 때 한정된 스펙트럼을 효율적으로 사용해야 한다. 또한 링크의 손실특성을 해결할 수 있는 링크계층 프로토콜의 설계도 해결해야 할 과제이다^[9]. 대개 Interleaving과 FEC를 사용함으로써 무선채널은 적절한 채널성능을 제공할 수 있지만, 이는 채널에 고정된 지연을 발생한다는 단점이 있다^[10]. 무선채널 상에서의 RTT 지연에 더하여서, DSP 처리지연, 노드의 내부지연, 전송지연 등이 손실을 가져올 수 있는 변수들이다. 음성데이터의 경우 보통 100~200ms 가 실시간에 적당하다고 했을 때, 유료부하를 제외한 100바이트 이상의 헤더로는 실시간 서비스를 제공할 수 없어서 이러한 문제를 해결하기 위해서는 헤더부분의 압축이 필요하게 된다. 헤더압축은 사용자에게 큐잉과 전파지연을 고려하여 10ms이하로 지연을 줄일 수 있고, 45.2kbit/s를 요구하는 GSM 네트워크에서는 11.6kbit/s의 대역폭효율을 얻을 수 있다. 또한 이더넷 프레임의 최대 크기인 1500 바이트의 경우 71%의 오버헤드를 0.4%까지 줄일 수 있게 된다^[11].

따라서 본 연구에서는 이동성 IP기반 네트워크에서, 비디오 트래픽 전송을 위한 효율적인 헤더압축 기법을 제안하고자 한다. 이를 위해 지금까지 제안된 기법을 분석하여, WCDMA 링크와 같은 손실특성을 가진 링크에서 기존 기법의 문제점을 분석하고, 이동호스트의 핸드오프로 발생하는 패킷 손실로 인해서 압축상태가 파손되었을 때, 허용 가능한 시간 안에 압축 상태를 신속하게 복구하는 기법을 제안한다.

II. 헤더압축 알고리즘

헤더압축은 패킷스트림(Packet Stream)에 그 개념을 두고 있는데, 동일한 스트림(세션) 내에 존재하

는 헤더영역의 대부분은 동일한 값을 갖게 된다. 예를 들어, IP나 UDP/TCP헤더 내에 존재하는 주소영역과 포트영역은 같은 스트림내에서는 같다. 또한 RTP의 TimeStamp나 Sequence Number같은 영역은 값이 순차적으로 일정한 크기를 가지고 변하며, UDP 길이영역은 링크계층의 영역과 중복된다. 이러한 패턴분석을 기반으로, RTP/UDP/IP 헤더영역의 중복되는 부분은 제거하고, 일정하게 증가하는 영역의 값에 대해서는 변화량만을 전송하면 헤더압축이 가능하다는 방법이 현재 RFC 2508로 승인되어 있다. 이는 UDP 체크섬(checksum)이 사용되지 않았을 때 최대 2바이트의 압축이 가능하며, 그 외에 프로파일 식별자에 의해서 compressor와 decompressor 사이에 사용되는 미디어 타입이나 채널특성 및 기타 요구사항에 대해서 협상을 할 수 있는 압축 프로파일을 정의하고 있다. decompressor측에서 context invalidation과 패킷 스트림 간격, 그리고 갱신주기 등을 지시할 수 있는 향상된 피드백 기법을 제공하는 이 방법은 비디오 트래픽에 대한 압축패킷이 6비트 CRC 비트를 포함하여 2 바이트의 크기를 가질 수 있음을 보여주며, 또한 24개의 연속된 패킷 손실에 대해서도 좋은 성능을 보이고 있다^[4].

ACE(Adaptive header ComprESSION) 기법은, Compressor가 비압축상태에서 출발해서 점차적으로 더 높은 압축상태로 전이되는 방법을 사용하는데, 이는 압축기가 충분한 신뢰(confidence)를 가졌을 때만 가능하다. 이때 Decompressor는 높은상태(higher state)에서 압축을 풀기 위한 정보를 얻게 되며, 이 경우 신뢰과정은 decompressor로 부터의 ACK에 의해서 이루어진다. 이러한 전이과정이 아래 [그림1]에 나타나 있다. Compressor 상태는 먼저 필수적으로 초기화나 재초기화의 경우에만 동작하는 완전헤더(Full Header: 이하 FH로 표기)를 전송한다. FH가 전송된 후에 곧바로 FO(First Order) 상태로 전이해서, Sequence Number 와 기타 추가적인 정보를 전송하게 된다. 그 후 가장 높은 압축상태인 SO(Second Order)단계에서 Compressor는 짧은 일련번호(Sequence Number)만 전송하게 된다.

Decompressor가 Compressor와의 동기화 손실을 감지했을 때, Compressor에게 Refresh 정보를 보내라고 요구하는 RFC2508은 Refresh 정보로 40~60 바이트의 완전헤더를 사용해야 함으로 압축효율에 부정적 영향을 미치고, 대역폭에 대한 요구가 갑자기 증가하기 때문에 현재의 무선 데이터 기술에서

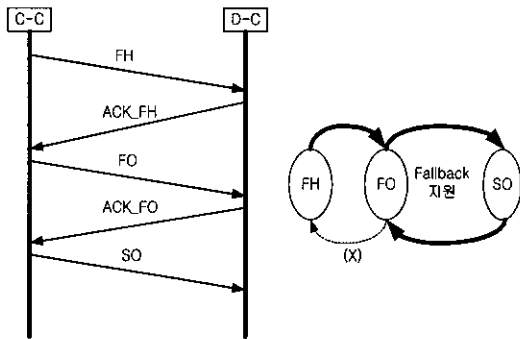


그림 1. 헤더적용기법

는 올바르게 동작하지 않는 단점이 있다. Refresh를 기다리고 있는 동안에, Decompressor는 모든 들어오는 압축헤더를 폐기해야 하고, 심지어 동기화 손실에 의한 에러전파가 되지 않은 것들도 폐기해야 하기 때문이다. 이러한 문제점에 대한 대처방안으로 ACE는 ACK을 사용해서 동기화 손실문제를 해결한다. 지금까지 논의한 기법들은 현재 IETF WG에서 논의되고 있는 것들이다. 링크에 의해 패킷이 손상되었거나, 손상되지 않았더라도 이어지는 패킷에서 연속적으로 손실이 발생하는 에러전파 문제를 줄이기 위해서 사용되는 체크섬 또는 CRC나 완전 헤더를 전송하지 않고도 버스트 에러에 대한 복구가 가능한 multiple extrapolation 등과 같은 다양한 기법을 제안하고 있지만, 긴 RTT 값을 가지는 현재의 무선링크 특성에서는 잘 동작하지 않는 단점이 있다. 또한 셀룰러 링크에서 흔히 발생하는 핸드오버에 대해서도 성능이 검증되지 않은 상태이다. 이동호스트가 IP 액세스 포인트로부터 다른 접속지점까지 이동했을 때, 새로운 네트워크 노드에서 패킷 손실에 대한 영향을 받지 않는 헤더압축 작동을 계속하기 위해서는 라우터와 이동호스트 간에 네트워크(셀) 변경에 대한 빠른 인식이 필수적이다. 이는 손실되는 패킷의 수를 감소시켜서, 대역폭의 효율을 떨어뜨리지 않게 된다.

III. 제안된 헤더압축 및 에러복구 기법

본 연구에서는 비디오 트래픽에 대한 효율적인 헤더압축 기법을 적용하기 위해 MPEG 스트림을 사용하였다. 보통 MPEG 인코더는 비트율, 프레임율, 그리고 비디오 품질의 선택이 자유로우며, 인코딩 단계에서 선택할 수 있는 다음의 4가지 구조가 가능하다. 먼저 Still Image와 같은 Intra(I)

picture가 있고, 이전의 그림에 의존하는 Inter(P) picture가 있다. 또한 이전과 다음의 그림 둘 다에 의존하는 Bilinear(B) picture가 있는데 이는 다소 복잡한 알고리즘에 사용된다. RTP를 사용해 Video를 압축할 경우, 한 개의 패킷은 하나 이상의 그림을 포함하지 않는다. 한 개의 그림은 두 개 이상의 RTP 패킷으로 나누어 질 수도 있다. RTP Marker bit은 하나의 그림을 가지고 만들어진 패킷 중 마지막 패킷에서 1로 설정된다. 비디오 정보와 관련된 헤더압축 프로파일은 참조클럭으로 90 kHz를 사용한다. 본 연구에서는 헤더압축을 위한 비디오 프로파일을 설계하기 위해서 RTP/UDP/IP 구조를 이용하였고, TimeStamp, Sequence Number의 변화를 어떻게 표현할 것인지와, CRC를 사용하기 위한 헤더구조에 초점을 두고 연구를 진행하였다.

1. Sequence Number 압축

순서번호의 압축과정은 다음과 같다. SEQ7은 순서번호에 대한 모듈로-7 연산을 수행한 결과이며, SEQR은 7에 의해 나누어진 후의 정수부분을 의미한다. 순서번호의 압축은 $SeqNr = SEQR * 7 + SEQ7$ 수식에 의해 간략화 될 수 있다. SEQ7은 압축헤더의 첫 3비트에 표시되어 전송된다. 기존의 방법은 헤더 압축 전에 1개의 reordering을 처리하기 위해 사용하는 코드의 순서번호 변화로 [-1, 5]범위를 다루고 있다. 이 방법은 순서번호의 동기화를 잃지 않고도 연속적인 4개의 패킷 손실을 복구할 수 있다.

본 연구에서는 보다 확장된 방법으로 SEQRLSB를 이용하여 확장헤더가 전송되도록 하였다. 여기서 SEQRLSB는 SEQR의 LSB(least significant bit)를 의미한다. 확장헤더를 이용한 후에 순서번호의 변화는 $[-3, (7 * 2^{SEQR_NB}) - 4]$ 과 같은 수식을 갖게 된다. SEQR_NB는 decompressor에 전송된 SEQR 비트수이며, 이처럼 범위를 확장함으로써 다중패킷 손실에 대한 강인성을 제공할 수 있다. 순서번호를 Decompressor 측에서 해석하기 위한 절차로, S(n)을 RTP의 순서번호라 하고, S(n-1)은 가장 최근에 디코드된 순서번호라 하면, 제안된 기법에 의해 순서번호의 범위를 확장했을 때 아래와 같은 형태로 표현할 수 있다.

Decoded Sequence Number :

$$S(n) = S(n'-1) - SEQRLSB(n'-1)*7 - S(n'-1) \text{ modulo } 7 + SEQRLSB(n')*7 + S(n') \text{ modulo } 7$$

만약 이전의 순서번호에 대한 SEQR 비트의 전송이 없다면, 디코더는 순서번호를 디코드하기 위해 선행하는 순서번호 값을 계산해야 한다. 이 경우 Modulo wrap-around 검사는 순서번호의 Delta값인 [-3, (7*2SEQRLSB_NB)-4] 사이의 간격을 처리할 수 있도록 바뀐다. RTP 순서번호 Wrap-around는 16비트 크기의 순서번호가 가질 수 있는 최대값인 65535에서 다시 0으로 넘어갈 때 SEQRLSB의 값이 잘못 인식되는 것을 의미하며, 아래의 [표 1]에서 이를 보여준다. Wrap-around에서 패킷 손실이 발생하면, 순서번호의 디코드된 값이 잘못될 확률이 증가하는데, 이런 현상을 피하기 위해서 동적 패킷이나 압축 패킷의 확장을 사용해서 순서번호를 코드화 하였다.

표 1. 순서번호 wrap-around

SeqNr	SEQ7	SEQRLSB
65532	5	...01
65533	6	...01
65534	0	...10
65535	1	...10
0	0	...00
.....
11	4	...01

2. TimeStamp(TS)의 효율적인 압축

압축효율을 위해 현재의 RTP TS변화는 고정된 값의 배수로 변화도록 설계하였다. 이는 현재의 Picture Clock 주파수(PCF)와 같은 값이다. 같은 그림내에서는 TS의 차이가 0이 된다. 그리고 B를 뒤따르는 B뿐 아니라 원래의 Intra(I)나 Inter(P) 그림에 대해서는 TS의 값이 PCF의 배수가 되거나 0보다 큰 값이 될 것이다. H.261, H.263 ver 1의 PCF는 29.97Hz 으로, 이는 두 코드화된 그림 사이의 증가가 3003이 된다는 의미이다. 또한 H.263 ver 2, MPEG-4의 PCF는 각각 25Hz, 30Hz가 되는데, 이 값은 각각 3600, 3000배의 TS에서의 증가와 같은 값이다. 패킷 손실시에도 TS의 증가는 PCF의 배수이므로 이런 성질은 PCF 배수인 TS의 증가분을 아는데 충분하므로 압축에서 사용하면 이득을 얻을 수 있다. 상용용 코덱이나, 이에 따르지 않는 코덱들도 지금까지 설명한 비디오 프로파일의 TS값을 확장헤더를 사용해서 보낼 수 있다. B-picture 가 사용되는 경우, 두 그림 사이에서 TS가 감소하게

되는 경우가 발생한다. 이러한 경우에 대처하기 위해서 다음과 같은 방법을 사용하였다.

$$TS = TSQ * PCTSI + TSR$$

TS = RTP timestamp
 PCTSI = PiCTure clock interval in TS
 (ex. 90000/25Hz = 3600)
 TSQ = TimeStamp Quotient
 TSR = TimeStamp Residual(=TS modulo PCTSI)

만약 두 패킷 사이의 TS의 변화가 PCTSI의 배수이면, TSR은 같은 값이 유지되면서, TSQ만 바뀔 것이다. TSQ를 인코드하기 위해 5 LSB비트가 기본압축 패킷에서 전송된다. TS값이 감소되는 경우를 처리하기 위해서 TSQ 비트는 [-6, 25]의 범위와 일치하는 Delta TSQ의 범위를 처리하는 방법을 이용한다. 이 방법을 확장했을 경우 [-10, 2^{N+5}-11]을 사용할 것을 제안한다. 여기서 N은 확장부분에 전송된 TSQ의 비트수 이다. 간단한 예로, 압축헤더에 5 개의 TSQ 비트가 사용되면 [-6, 25]를 처리하게 되고, 여기에 확장헤더를 사용하여 추가의 2비트 TSQ 를 만들면 [-10, 117]를 처리할 수 있다.

PCTSI값은 동적 패킷이나 압축 패킷의 확장부분에 있는 TS Delta영역을 사용하여 전송하면 된다. 만약 TS가 PCTSI의 배수로 바뀌지 않는다면 압축 패킷을 사용하여 여러개의 TS LSB를 확장부분에 넣어서 전송할 수 있다. TS_LSB_NB를 TS_LSB영역의 크기라 놓으면, TS가 폐기되는 것을 방지하기 위해서 TS LSB는 마지막으로 전송된 TS로 부터 [2¹⁶, 2^{TS_LSB_NB} - (2¹⁶ - 1)]의 범위를 처리해야 한다.

3. TimeStamp의 효율적인 압축해제 기법

TS의 경우는 순서번호에서 사용하고 있는 modulo 7을 사용하는 대신에 아래의 수식을 사용하였다.

$$Decoded TS: T(n) = T(n'-1) - TSQLSB(n'-1) * TS_delta + TSQLSB(n') * TS_delta$$

여기서 T(n)은 현재의 TS이고, TSQLSB 는 TSQ의 LSB의 수를 의미한다. 만약 TSQLSB(n'-1)과 TSQLSB(n')가 같은 수(크기)의 TSQ비트로 수행되지 않는다면 디코더는 TSQLSB(n')와 같은 길이가

되도록 TS LSB(n'-1)을 다시 계산해야 하고, 만약 TS LSB(n')가 TS LSB(n'-1)-7 보다 작거나 같다면 T(n)는 2^{NB}로 증가된다. 여기서 NB는 TS LSB(n') 전송에 사용된 비트 수를 의미한다.

순서번호처럼 TS도 중복(wrap-around) 문제가 발생하게 되는데 이의 해결은 다음과 같다. RTP TS는 PCTSI에 의해 나뉘지고, TSQ의 5비트 LSB는 11111에서 00000까지 바뀌지 않을 것이기 때문에, 중복되는 동안 TSQ의 범위는 더 작아진다. 본 연구에서는 RTP TS의 정확한 Decompression을 위해서 DYNAMIC패킷이나 COMPRESSED 패킷의 확장 부분에서 전송되는 것을 고려하였으며, 이경우 적어도 21비트 크기의 TS LSB가 사용되는 것을 제안하였다. 헤더압축 세션동안 TS Delta가 바뀌었을 때 새로운 TS delta와 현재의 TS값이 모두 Decompressor에 전송되어야 한다.

4. STATIC, DYNAMIC 압축헤더 설계

본 연구에서는 STATIC(11100), DYNAMIC(1111*), COMPRESSED(나머지) 등 3가지 타입의 패킷을 설계하였다. 먼저 유료부하가 없는 STATIC 구조는 [그림 2]와 같다. 단지 한개의 STATIC패킷이 각각의 경우에 전송되는데, 만약 Decompressor가 DYNAMIC이나 STATIC 패킷 없는 헤더압축을 받는다면, Decompressor는 STATIC패킷을 요구해야 한다. 두번째 패킷은 DYNAMIC 정보패킷(Information Packet)이다. 원래의 헤더에서 모든 바뀌는 부분을 포함하는 헤더인데, 원래의 COMPRESSED 패킷처럼 페이로드를 전송한다. 이 패킷은 우선 Decompressor 컨텍스트(Context)를 설정하기 위해 처음에 전송된 STATIC 패킷 후에 전송된다. 또한 이 패킷은 COMPRESSED 패킷에 인코드 될 수 없는 방법으로 헤더영역이 바뀌었을 때 사용된다. 아래의 [그림 3]은 동적 패킷의 구조이다.

0	1	2	3	4	5	6	7
Context Identifier(CID)							
1	1	1	0	0	F	P	E
SA(4 bytes)							
DA(4 bytes)							
SP							
DP							
SSRC(4 bytes)							
Header Compression CRC							

* F(Fragment), P(padding), E(RTP Extension)

그림 2. IPv4 STATIC 구조

0	1	2	3	4	5	6	7
Context Identifier(CID)							
1	1	1	1	CSRC counter			
TimeStamp Delta							
Traffic Class							
Hop Limit							
UDP checksum							
M	Payload Type						
Sequence Number							
TS(4)							
CSRC List(0~15*4)							
Header Compression CRC							
.... Payload							

그림 3. 동적 패킷의 구조

5. 압축헤더의 확장

헤더영역에서 덜 불규칙적인 변화는 기본적인 헤더에 추가하여 확장헤더를 요구한다. 처음 3비트는 확장부분의 타입을 위해서 사용된다. TS Delta, TS LSB, TSQ, SEQR, TSC, Bit mask등이 확장부분을 이용하여 전송할 수 있는 것들이다. TSQ의 추가적인 LSB는 TS범위를 증가시키기 위해서 사용된다. 기본헤더의 비트는 항상 LSB이다. SEQR 영역은 SN 나머지를 표현하는 LSB값 이다. TSQ LSB와 TS LSB가 같은 헤더 내에 존재한다면, TSQ LSB는 무시되고 TS LSB만 고려가 된다. TSC는 큰 TS Delta 영역 대신에 COMPRESSED 패킷의 확장부분에서 사용된다. 아래의 [표 2]는 TS Delta를 위한 TSC 값에 대한 정의를 보인다.

표 2. TS Delta에 따른 TSC값의 변화

TSC	TimeStamp Delta
00	3000(30Hz)
01	3003(29.97Hz)
10	3600(25Hz)
11	other

6. 핸드오버시 효율적인 압축상태 복구기법

무선 인터넷에서 Mobile IP기술의 효율적인 지원은 인터넷과 무선 네트워크의 통합에 중요한 역할을 한다. 노드가 한 네트워크에서 다른 네트워크로 이동했을 때, Mobile IP는 새로운 네트워크로 데이터그램을 정확하게 전달하기 위해서 사용된다. Home Network에 있는 HA(home agent)는 데이터그램을 FA(foreign agent)로 터널링하고, MH의 새로운 위치는 등록신호를 사용해서 HA에 알려지게

된다. HA와 FA는 주기적으로 에이전트 광고메시지를 전송하는 기능을 가지며, 등록요청은 FA를 경유하여 MH로 전송되는 등록응답에 의해서 종료된다. [그림 4]에서 이러한 제안된 기법에서의 메시지교환 과정을 보여준다.

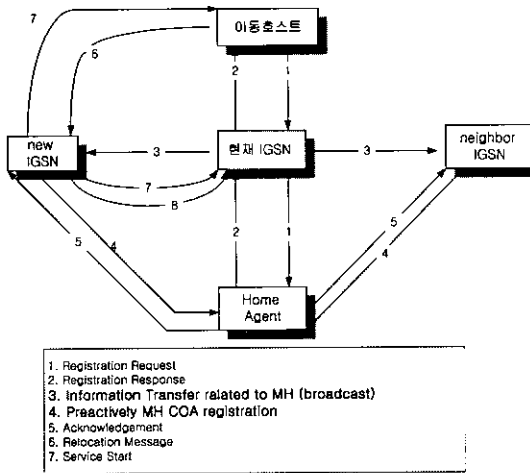


그림 4. 제안된 기법에서의 메시지교환

Mobile IP 기술을 3GPP에 통합할 경우, FA기능을 가질 수 있는 IGSNs (Internet GPRS support nodes)라 불리는 CN nodes가 사용되고 있다. PPP 세션설정 동안에, MH는 IGSN으로부터 COA를 얻고 이 주소를 자신의 홈 IP 주소 대신에 사용한다. 연결이 설정되면, IGSN은 에이전트 광고메시지를 전송하게 되고, MH는 자신의 COA를 얻고, HA에게 등록요청을 한다. HA는 홈 네트워크에 도착하는 데이터그램을 가로채서 COA값이 등록되어 있는 네트워크로 터널링하게 된다. 핸드오프 동안 MH가 다른 IGSN에 의해 서비스되는 영역으로 이동하게 되면, MH는 새로운 IGSN을 가진 링크계층 연결을 설정한 후에, 다시 Mobile IP 등록절차를 수행한다. HA에게 자신의 새로운 COA를 등록한 후에, HA는 CN으로부터 새로운 IGSNnew으로 데이터그램을 터널링하고, 다시 새로운 IGSNnew은 MH에게 데이터그램을 전달하게 되는 과정을 거친다. 기존의 IGSN은 계속해서 패킷을 MH에게 전송하지만, MH가 이미 이동하였기 때문에 패킷을 수신할 수 없다. 이때 패킷이 폐기될 것이고, 새로운 IGSNnew이 압축상태를 복구해서 MH가 서비스를 받는데 걸리는 시간 지연은 실시간 정보의 전송에 큰 장애가 된다. 이런 구조에서는 압축상태의 손실 때문에 다중 패킷 손실을 피할 수 없다. 이를 해결하기 위해서 [그림 5]와 같은 절차를 제안하였다.

림 5]와 같은 절차를 제안하였다.

먼저 MH가 핸드오버 동작을 일으키기 전에, MH는 링크계층 연결을 위해서 과정 1,2를 수행하고, IGSN과 HA 사이에 등록과정이 6번과정 까지 수행이 된다. 그리고 나서 MH는 서비스를 받게 되고, 다시 IGSN은 MH와 관련된 압축 프로파일에 대한 정보를 인접한 IGSN에게 알린다. 인접한 IGSN들은 MH와 관련된 COA 및 압축정보를 얻었기 때문에, 아직 서비스 되지 않은 MH를 위한 COA 값을 할당하여 MH의 HA와 등록과정을 수행하게 된다. 이렇게 되면 MH가 새로운 노드로 이동을 하더라도 등록과정이 생략되며, MH의 Relocation Message에 의한 서비스 시작을 알려줌으로써(ACK) 바로 헤더압축을 포함한 패킷 서비스가 가능해진다.

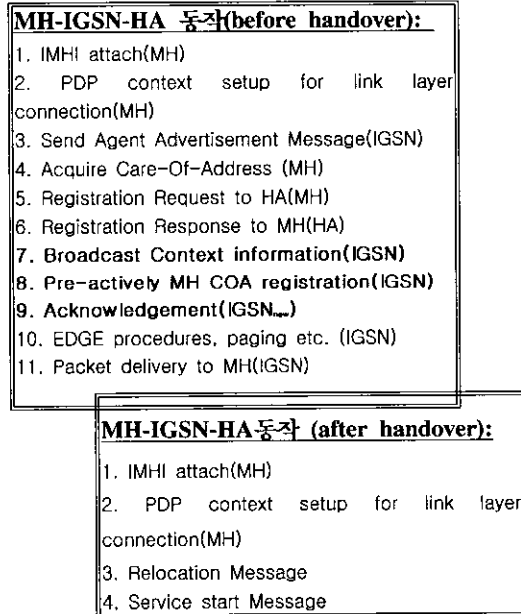


그림 5. 제안된 링크연결 과정

IV. 시뮬레이션 및 성능분석

1. 시뮬레이션 모델 및 파라미터

제안된 기법의 타당성을 검증하기 위하여 다음과 같은 기본 가정을 하였다. 첫째, Forward 채널 상에서 전송된 패킷은 손실될 수 있고, 패킷 에러의 특별한 형태가 없다. 둘째, 에러검출기법을 이용하여 손상된 정보가 전달되지 않고, 에러검출은 링크에 의해서 제공된다. 셋째, 패킷의 순서는 Compressor와 Decompressor 사이에서 유지된다. 즉 수신기는

항상 전송측에서 보낸 순서대로 패킷을 받는다. 넷째, 피드백 채널 상에서 엄격한 지연이나 에러요구 사항이 없다. ACK은 손실 및 지연될 수 있으며, 에러와 지연의 특성은 시간에 따라 변동될 수 있다. 다섯째, 패킷순서는 피드백채널 상에서 유지된다. 여섯째, ACE는 RTT의 어떤 특별한 분포를 가정하지 않는다. 동적으로RTT에 대한 변화에 적응할 수 있으며, 헤더압축은 인터리빙과 채널코딩에 영향을 미치지 않는다.

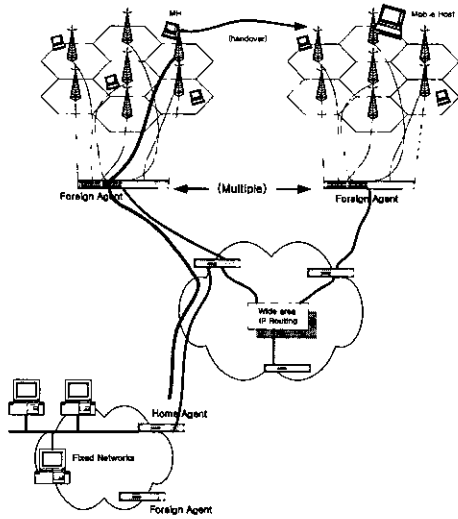


그림 6. 시뮬레이션 모델

제안된 헤더압축상태의 빠른 복구과정을 분석하기 위해서 전술한 가정하에 NIST network simulator를 사용하여 시뮬레이션을 수행하였다. 링크 레벨 시뮬레이션은 WCDMA 채널모델을 사용하였다. 시뮬레이션 모델은 [그림 6]과 같으며, 사용된 파라미터는 다음과 같다.

PPP link layer connection
Packet rate = 100 packets/s
Video code rates = 384 kb/s
320 bit frames
Header profile = 1~2byte header
Round Trip Time = 120ms
HA = 1
FA=4
MH=20
Cell # = 14

2. 패킷 손실률 분석

FEC, RF 변조 및 다중액세스 기법에 대한 효과

는 제외하고 핸드오프만을 고려하였다. RTP/UDP/IP Agent기능을 가진 이동호스트는 비디오 트래픽을 발생시키고, Compressor까지 전송되어 온 패킷은 압축 초기화 과정을 거친 후에 RTP 프로토콜이 지원하는 세션 내에서 2바이트 크기의 헤더를 만들어 전송하게 되고, 이를 수신한 이동호스트에서 압축헤더를 복구하게 된다. [그림 7]은 비트에러율이 주어졌을 때, 제안된 압축기법과 기존의 방법과의 패킷 손실률을 비교한 것이다. 주어진 BERs에서 제안된 기법은 패킷 손실률이 5% 이내로 제한됨을 확인하였다. 기존의 [RFC-2508]에서 제안된 기법의 경우는 패킷 손실률이 최대 30~35%까지 발생하므로, 제안된 기법이 패킷 손실률에서 기존의 방법에 비해 보다 좋은 성능을 보임을 확인하였다.

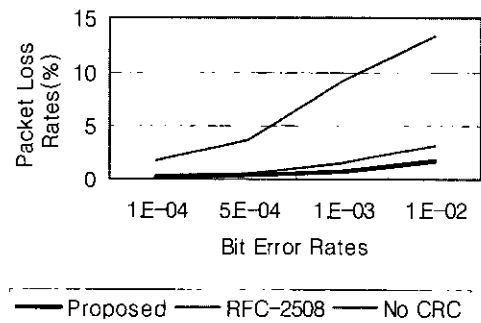


그림 7. BERs에 따른 패킷손실률의 변화

3. 평균 헤더크기 분석

압축헤더는 최대 1바이트까지 압축할 수 있지만, 핸드오프시 압축상태가 손실되어 이를 복구하기 위해서 다시 완전헤더를 전송하게 되면 2바이트의 헤더크기가 40바이트까지 늘어나게 된다. 또한, 평균 헤더크기에 있어서 [RFC-2508]의 경우는 BERs이 커짐에 따라 4바이트까지의 헤더 증가가 되지만, 제안된 기법은 평균 1~2 바이트 크기에서 안정적인 성능을 보인다. [그림 8]은 평균헤더크기에 대한 성능을 비교해 놓은 것이다. [RFC-2508]의 경우는 twice알고리즘의 사용에도 불구하고, 압축상태의 빠른 복구가 에러발생이 높은 무선 링크 상에서는 잘 작동되지 않기 때문이다. 하지만 제안된 기법은 셀 내에서 이동호스트가 다른 셀로 이동하더라도, 미리 ISGN간에 설정된 압축상태에 대한 정보와 사전등록과정 때문에 패킷손실률을 줄일 수 있어서 대역폭 효율적인 헤더압축이 가능하다.

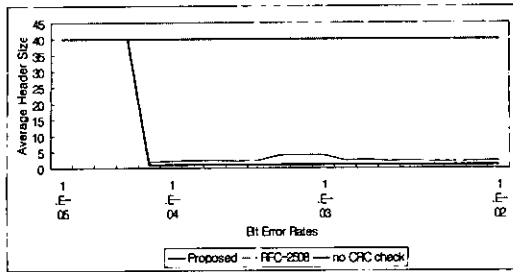


그림 8. BERs에 대한 평균헤더의 크기

4. Context 재설정 시간 분석

핸드오프 동작에 의해 이동호스트가 위치를 변경했을 때 새로운 IGSN으로 부터 서비스를 받는데 걸리는 시간을 분석한 결과 평균 130ms정도, 최대 250ms의 시간이 소요되는 것을 확인할 수 있었다. [그림 9]는 지연에 의한 Context 재설정 시간을 비교한 것이다. 20ms 프레임이 7개 정도 손실될 수 있는 시간이지만 우리가 설계한 비디오 헤더압축 프로파일에서는 최대 25까지의 패킷 손실을 복구할 수 있기 때문에 문제가 되지 않는다. 이동호스트가 비디오 스트림에 대한 RTP SN이나 TS 차이 값을 예측함으로써 25개 이상의 연속된 패킷을 복구할 수 있는 기법을 사용하였다. MPEG-4 비디오의 경우, 종점간 지연값이 최대 400ms 이기 때문에, 핸드오프 재설정 시간인 130ms는 양호한 값이 된다.

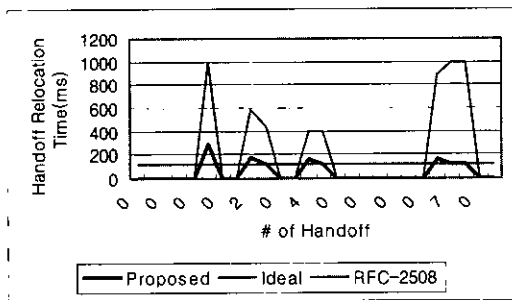


그림 9. 지연에 의한 Context 재 설정시간

5. User Payload에 따른 대역폭 효율

UDP를 사용한 경우 사용자 유료부하에 따른 대역폭 변화에 대한 실험을 하였다. 2%의 FER을 가정하고 2바이트 UDP/IP 헤더를 사용하여, 사용자 데이터그램의 크기를 변화 시켰을 때 사용자 데이터에 대한 오버헤드 비율과 이용할 수 있는 전체 대역폭에 대해서 [그림 10]과 같은 결과를 얻을수

있었다.

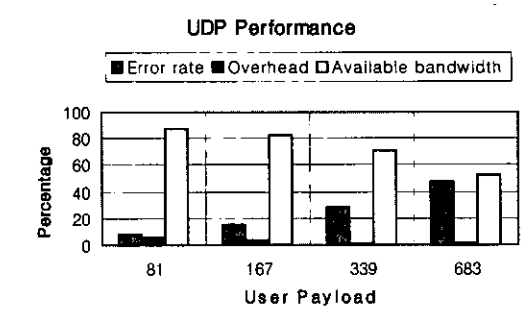


그림 10. User Payload에 따른 대역폭 효율

81 바이트의 사용자 유료부하가 약 87%로 최대 대역폭 이용율을 가짐을 볼 수 있었다. 81 바이트의 데이터그램에 대해서 에러율은 4개의 링크계층 프레임이 사용되었기 때문에 $(1-0.984)$ 인 0.07763을 보였다. 오버헤드는 $5/86$ 인 0.05814이며 시스템이 이용할 수 있는 전체 대역폭은 20ms 당 하나의 320 비트 프레임인 16kb/s 이기 때문에, 사용자 데이터율은 $(1-0.07763) \cdot (1-0.05814) \cdot 8.6$ 은 7.47kb/s 가 되는데, 이는 대역폭 효율이 약 87%가 됨을 의미한다. 사용자 페이로드가 작아질수록 비트 에러율이 줄어들다는 것을 알 수 있었고, 더 적은 사용자 페이로드는 헤더에 대한 오버헤드를 증가시키기 때문에 최대 1 바이트까지의 헤더압축으로 오버헤드에 대한 영향을 감소시킬 수 있음을 확인하였다.

V. 결론

IP 이동 멀티미디어 네트워크를 구현하기 위해서는 지금까지 논의한 제반사항들에 대한 해결이 선행되어야 한다. 비디오 패킷의 스펙트럼 효율적인 전송은 셀룰러 액세스 네트워크에서 매우 중요하므로, 대역폭의 효율적인 사용과 원하는 서비스품질은 효율적인 헤더압축 기법과 함께 제공되어야 한다.

제안된 기법은 셀룰러 IP 환경에서 핸드오프시 최대 250ms 내에 압축상태가 복구되는 것을 확인하였으며, 이는 이동호스트가 핸드오프 하기 전에 미리 주변의 FA간에 캐시화된 정보와 사전 등록과정을 이용하였기 때문이다. 셀룰러 시스템에서 비디오의 효율적인 전송을 용이하게 하기 위해 적용한 CRC 기법으로 인해 지역에서의 손실복구가 가능하였다. 그러나 이는 비트에러를 검출하기 위한 패이

