ROHC 압축 기법을 적용한 SCTP 헤더 압축 연구

송희옥* · 최문석* · 최성곤* · 신병철* · 이인성*

A Study on SCTP Header Compression using the ROHC Method

Hee-Ok Song* · Seong-Gon Choi** · Byung-Cheol Shin** · Insung Lee**

이 논문은 2003년도 한국과학재단 목적기초연구과제(R01-2003-000-11620-0)지원으로 수행되었습니다.

요 약

본 논문에서는 무선을 포함한 패킷 전송에서의 효율성을 위해 차세대 전송 계층 프로토콜인 SCTP에 ROHC(RObust Header Compression)를 적용한 ROHC 프로파일 7(ROHC-SCTP)을 제안한다. 제안된 ROHC-SCTP는 SNR(Sequence Number ROHC)-SCTP의 1 또는 2 바이트 길이의 새로운 필드를 SCTP 패킷 헤더에 추가하여, SCTP 멀티 스트림을 acknowledgement 스트림과 data 스트림으로 나누어 헤더 압축을 하며, 동기화 손실이 발생한 경우 context를 복구하는데 사용될 수 있도록 설계되었다. ROHC-SCTP의 제안을 통해 SCTP 헤더의 오버헤드 율을 최대 25%까지 감소할 수 있으며 대역폭 또한 최대 5% 정도 절약할 수 있다.

ABSTRACT

In this paper, we propose a new profile, ROHC(RObust Header Compression) profile 7, for SCTP with ROHC for applying robust header compression SCTP, which is a transport layer protocol. The proposed new profile 7 adds a new field of 1 or 2 byte size on the existing SCTP packet header, which can make the SCTP stream to be divided into acknowledgement stream and data stream. In addition, the classification of the stream can be used for recovering fault context. Consequently, in the case of using proposed ROHC-SCTP, it is possible to reduce the SCTP header overhead rate and also can be saved bandwidth.

키워드

SCTP, ROHC, SNR-SCTP, 헤더 압축

Ⅰ. 서 론

최근 무선을 이용한 데이터 전송방식은 가장 광범위한 통신 시장 영역으로 스마트 폰, PDA 그리고 랩탑과 같은 3G 네트워크와 단말 장치들 대부분에 IP의 응용서비스를 위한 IP와 무선의 복합적 연구가 활발히 진행되고 있다. 그러나 이런 서비스들은 기존의 음성 서비스보다 훨씬 더 많은 대역폭을 요구하여 무선 자원은 이미 한계에 도달하고 있다. 그러므로 멀티미디어 서비스에 사용되는 RTP(Real-time Transport Protocol)[1], UDP(User Datagram Protocol)[2], IP와 같은 프로토콜 헤더의 오버헤드를 압축하여 대역폭을 절약할 수 있는 헤더 압축 기술이 요구되고 있다.

또한 인터넷상에서 실시간 멀티미디어를 위한 전송 프로토콜도 기존의 TCP(Transmission Con-

* 충북대학교

접수일자 : 2004. 11. 1

trol Protocol)[3]나 UDP로부터 새로운 전송 프로토콜인 SCTP(Stream Control Transmission Protocol)[4]로 대체되고 있다. TCP 및 UDP 프로토콜은 기존의 전기통신망에서 제공되던 전화서비스를 비롯하여 새로이 개발되는 고도의 실시간 멀티미디어 응용 서비스의 제공에는 부적합하다. SCTP는 IETF에서 제정한 새로운 전송 프로토콜로서 다양한 종류의 서비스를 적용할 수 있도록 적용 범위를 확대하고 있다.

제한된 대역폭에서 문제점을 해결하기 위한 기술들이 유선망보다는 대역폭의 사용비용이 높은 무선망에서 활발히 연구되고 있다. 대표적인 기술은 Header Removal/Generation, Header Striping/Regeneration, 헤더 압축 기술[5] 등이 있으며, 여러 가지 기술 중에서 가장 이슈가 되는 기술이 헤더 압축 기술이다. 헤더 압축 연구는 유선망환경에서 TCP/IP 프로토콜 헤더 압축[6][7], RTP/UDP/IP 헤더 압축[8]이 연구되었으며, 무선망에서는 ROHC(RObust Header Compression) 헤더 압축[9]을 중심으로 연구되고 있다. ROHC는 높은 비트 에러율(BER)과 긴 왕복소요시간(RTT)과 같은 무선망에서의 단점을 극복할 수 있도록 설계된 헤더 압축 프로토콜로서 현재 TCP 스트림 관리에 대한 특정 프로파일인 ROHC+[10]와 TCP/IP에 ROHC를 완전히 적용한 ROHC-TCP[11][12]가 출현되었다. ROHC-TCP는 서비스의 품질 상에서 직접적인 이익을 가져왔으나 대역폭의 경제적 가치에 대해서는 큰 효과를 보지 못했으며 특히 나날이 고갈되어가는 IP 주소에 대해서는 전혀 이익이 없다.

SCTP는 TCP의 연결형(connection-oriented) 및 신뢰성(reliability) 특성도 갖고 있고, 그 외에 멀티호밍 및 멀티스트리밍 특성을 제공한다. SCTP의 멀티호밍 특성은 SCTP 세션이 여러 개의 IP 주소를 동시에 사용할 수 있도록 하며, 멀티스트리밍 특성은 하나의 세션을 통해 다양한 종류의 응용데이터를 보낼 수 있도록 한다[13].

이와 같은 이유로 본 논문에서는 무선 및 유선망에서의 패킷 전송이 헤더의 오버헤드 율을 줄이고 대역폭 또한 절약할 수 있도록 하기 위해 SCTP에 ROHC 압축 기법을 적용한 ROHC-SCTP를 제안한다.

ROHC 압축 기법을 적용한 SCTP 헤더 압축 방법은 ROHC에서 제정한 헤더 필드 분류법으로 SCTP 헤더필드를 분류하고, SNR-SCTP라고 하는 1 또는 2 바이트의 새로운 필드를 SCTP 헤더 필드 중에 추가하여 SCTP 스트림을 acknowledgement(ACK) 스트림과 data 스트림으로 나누어 헤더 압축을 하며, 이러한 스트림의 구분은 SCTP 스트림이 동기화를 손실했을 경우 context 복구를 위해 사용될 수 있다.

ROHC-SCTP는 최대 SCTP 대비 약 25% 정도의 오버헤드 감소효과를 볼 수 있을 뿐만 아니라 대역폭은 5% 정도 절약할 수 있는 결과를 얻을 수 있다.

이 논문의 구성은, 2장에서 ROHC 압축 기법을 적용한 SCTP 헤더 압축에 관련된 연구를 하고, 3장에서는 구체적인 적용방법을 소개하며, 4장에서는 제안된 방법에 대해 성능 분석, 5장에서는 결론을 내린다.

II. 관련 연구 조사

이 장에서는 기존의 헤더 압축 방법들인 VJHC (혹은 CTCP)[6], IPHC[7], CRTP[8] 등의 기법을 이용한, 에러 율이 높고 실시간 지원을 요구하는 무선망에서의 헤더 압축 방법인 ROHC에 대해 소개하고, 새로운 전송 계층 프로토콜로 주목받고 있는 SCTP, 그리고 TCP 스트림에 대한 새로운 헤더 압축 방법인 ROHC+에 대해서도 간단히 소개한다.

2.1 ROHC[14]

ROHC는 기존의 헤더 압축 방법들의 경험에 비추어, 높은 비트 에러 율(BER)과 긴 왕복소요시간(RTT)이 있는 무선망에서 효율적이고 강인(robust)한 헤더 압축을 위한 방법이다.

ROHC 헤더 압축을 위해 헤더 필드는 STATIC, DYNAMIC와 INFERABLE 세 가지 클래스로 분류된다.

STATCI- 동일한 스트림에 속하는 패킷 헤더 내에서 변하지 않는 필드(예를 들면, IP 버전, 소스와 목적지 주소, 포트 번호 등);

DYNAMIC- 동일한 스트림에 속하는 패킷 헤더 내에서 가변 량에 의해 변할 수 있는 필드(예를 들면, 식별자, 체크섬 등);

INFERABLE- STATIC 필드와 DYNAMIC 필드들을 통해 추론할 수 있는 여분의 필드(예를 들면, 패킷 길이).

ROHC 압축 방법은 컴프레서와 디컴프레서에서 각각 세 가지 압축 상태인 Initialization and Refresh(IR), First Order(FO), Second Order(SO) 상태와 No Context(NC), Static Context(SC), Full Context(FC)상태에 의해 부동한 압축 레벨을 나타낸다. 컴프레서와 디컴프레서 사이에서는 context를 통하여 상태 정보를 유지한다. 최적의 상태에서 컴프레서는 SO 상태를, 디컴프레서에서는 FC 상태

를 유지하게 된다. 또한, ROHC에는 상태 외에 새로운 개념인 모드가 있는데, 여기서 모드의 개념은 네트워크에 따라 피드백의 유무와 신뢰성이나 최적화에 따라 세 가지 부동한 동작 모드인 Unidirectional mode(U-모드), Bidirectional Reliable mode(R-모드)와 Bidirectional Optimistic mode(O-모드)가 존재하는데, 이 세 가지 모드는 압축 상태 사이에서 논리적으로 상호 전환된다.

그림 1은 컴프레서에서 세 가지 압축 상태사이의 전환 과정이다. 그림에서와 같이, 압축이 최적의 상태이거나 혹은 컴프레서가 디컴프레서로부터 context 업데이트 정보에 대한 ACK를 받았을 때 IR 상태로부터 FO 상태나 혹은 SO 상태로, 또는 SO 상태에서 SO 상태로 전환한다. 압축 과정에서 주기적인 타임아웃이 있거나, 헤더 필드의 변화 패턴을 업데이트 해야 되거나, 컴프레서가 디컴프레서로부터 NACK(Negative ACK)를 받았을 때 FO 상태에서 IR 상태로 혹은 SO 상태에서 IR 상태로 전환한다.

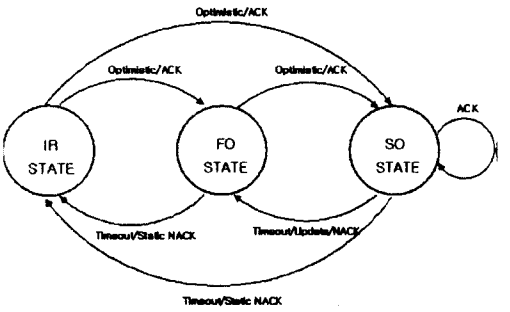


그림 1. 압축상태 사이의 전환
Fig. 1 Compressor states for all modes

또한, ROHC 프로토콜에서는 4개의 압축 프로파일이 소개되었다.

프로파일 0 (0x0000) : 압축 하지 않은(Uncompressed) 스트림(헤더 길이를 압축하지 않고 패킷을 보내는 경우);

프로파일 1 (0x0001) : RTP/UDP/IP 스트림(실시간 서비스에 적용);

프로파일 2 (0x0002) : UDP/IP 스트림(RTP는 압축하지 않는다);

프로파일 3 (0x0003) : ESP/IP 스트림(데이터 보안을 위하여 ESP 프로토콜을 사용하는 스트림).

이외에, ROHC+[10]에서 제안한 프로파일 4, 5와 draft 문서로 제출된 프로파일 6인 ROHC-TCP[11],[12]가 있다.

프로파일 4 (0x0004) : TCP/IP 스트림의 관리(management)에 대한 프로파일;

프로파일 5 (0x0005) : TCP/IP 스트림에서 손상된 TCP 조각을 재전송하는 경우를 위한 프로파일;

프로파일 6 (0x0006) : TCP/IP에 ROHC 압축 기법을 적용한 프로파일(ROHC-TCP).

2.2 SCTP[13]

SCTP는 기존의 UDP 및 TCP에 비해 많은 장점을 제공한다. SCTP는 UDP의 데이터 그램 지향성과 TCP의 순서화(sequencing) 및 신뢰성(reliability) 특징을 결합한 것으로 볼 수 있으며 멀티호밍(multi-homing) 환경에서 멀티 스트림(multi-stream) 및 메시지 지향성(message-oriented) 전달 특성을 갖는다. SCTP는 응용 계층과 네트워크 계층 사이에 위치하며, SCTP 지점 간에 응용 데이터를 APIs(Application Programming Interfaces)로 전달받아 IP 망을 통해 전송하는 기능을 수행한다. SCTP는 두 지점 간에 메시지 전송을 위해 멀티 패스(multiple path) 및 멀티 스트림(multi stream) 기능을 사용한다. 두 SCTP 지점간의 SCTP 연결을 "SCTP 결합(association)"이라 한다.

SCTP의 멀티 스트림의 특성은 하나의 세션을 통해 다양한 종류의 응용데이터를 보낼 수 있도록 한다. 세션 초기화 단계에서 송신자는 자신이 전송할 스트림의 개수를 수신자에게 통보하며, 전송 계단에서 각 스트림 별로 독립적인 순서화(ordering) 기능이 제공된다. 데이터 복구 및 재전송 과정 또한 스트림 ID 별로 수행되어, 기존의 TCP에서 문제시 되었던 HOL(Head-of-Line) 블로킹 문제를 해결한다.

SCTP는 멀티흐름 및 멀티호밍 특성 이외에도 TCP와는 다른 세션 초기화, 데이터 전송 및 세션 종료 절차를 사용한다. 초기화 과정에서 SCTP는 TCP의 three-way handshake와는 달리, four-way handshake 절차를 사용한다. 이러한 4단계 초기화 절차는 외부의 DoS(Denial of Service) 공격을 방어하는 데에 적합하다. SCTP 데이터 전송 단계에서, 각 데이터 chunk에게 TSN(Transmission Sequence Numbers) 및 SSN(Stream Sequence Numbers) 값이 부여된다. TSN은 세션에 대한 흐름제어 및 오류복구를 위해 사용되며, SSN은 수신단 SCTP에서 흐름별 순서화를 위해 사용된다. SCTP 세션 종료의 절차는 TCP의 "half-open" 문제를 해결한다. TCP에서는 4단계 절차를 통해 각 SCTP 개체별로 데이터 전송을 종료했으나, SCTP에서는 3단계 절차를 사용하여 두 개체의 데이터 종료가 통합적으로 이루어지도록 하였다.

2.3 ROHC+

ROHC+[10]는 상대적으로 높은 손실 특성을 나타내는 3G 플랫폼과 같은 무선 context에서 TCP/IP 스트림의 관리에 대한 특정 헤더 압축 프로파일을 정의했다. ROHC+는 TCP 스트림에 연관된 ACK 스트림과 data 스트림의 명확한 관리에 근거한 헤더 압축 방법이다. 또한 에러 율을 감소하기 위하여 손상된 압축 패킷에 대하여 강인한(robust) 인코딩 기술(W-LSB)과 세 가지 복구 메커니즘을 제기하였다. 결론적으로 ROHC+는 40-60 바이트 헤더를 4-6 바이트로 압축함으로서 비용과 서비스 품질 면에서 많은 이익을 가져왔다. 그러나 ROHC+를 새로운 전송 계층 프로토콜인 SCTP에 직접 적용할 수 없다. 왜냐하면, SCTP는 TCP가 가지고 있는 특성 외에도 멀티호밍이나 멀티스트리밍 같은 여러 가지 다른 특성도 포함하고 있으며, 특히 TCP는 단일 IP 주소를 사용하는 반면 SCTP는 동시에 여러 개의 IP 주소를 사용할 수 있기 때문이다.

2.4 요약

ROHC는 기존의 헤더 압축 방법들에 비해 높은 비트 에러 율(BER)과 긴 왕복소요시간(RTT)을 극복할 수 있는 무선망에서의 강인한 헤더 압축 방법이며, IP나 기타 전송 프로토콜과의 복합적 연구에 많이 쓰이고 있다. SCTP는 기존 TCP 및 UDP의 문제점을 극복하도록 설계되었으며 특히 멀티스트리밍 및 멀티호밍 특성을 제공함으로서, TCP의 HOL(Head-of-Line) 블로킹 문제를 해결하고 IP 경로 장애에 대한 복구(fail-over) 기능을 제공한다. ROHC+는 TCP 스트림에 ROHC를 적용함으로서 TCP 헤더를 10%까지 압축한 새로운 헤더 압축 방법이다. 하지만 ROHC를 SCTP에 직접 적용할 수 없다. 따라서 본 논문에서는 차세대 전송 계층 프로토콜로써 주목 받고 있는 SCTP에 적용할 수 있는 새로운 ROHC 프로파일 7을 설계함으로서, 무선망에서 전송되는 패킷 헤더를 대폭적으로 감소하여 헤더의 오버헤드를 감소하고, 대역폭도 크게 절약하는 방법을 제안한다.

III. ROHC 압축 기법을 SCTP에 적용시키는 방법

제안된 ROHC-SCTP는 ROHC 방법에 의해 채택된 헤더 필드 분류법을 이용하여 SCTP 헤더 필드를 분류한 다음 1 또는 2 바이트의 SNR-SCTP라는 새로운 필드를 SCTP 헤더 필드 중에 추가함으로서 SCTP 스트림을 ACK 스트림과 data 스트림으로 구분하여 헤더 압축을 진행하며, 이러한 구분을 통하여 SCTP 스트림에 동기화 손실이 생겼을 경우 더욱 적합한 복구 메커니즘을 제기한다. 본 장에서는 제안된 기법을 위한 헤더 필드의 분류, 추가 필드의 내용, 스트림의 사용과 복구 예 등을 소개한다.

3.1 SCTP 헤더 필드의 분류

그림 2에서와 같이 SCTP 패킷은 크게 Common Header와 Chunk로 구분된다. Common Header에서 소스 포트(16 비트)는 송신자 포트 번호이고, 목적지 포트(16 비트)는 수신자 포트 번호이다. Verification Tag 필드(32 비트)는 수신자가 SCTP 패킷을 보낸 송신자를 확인하는데 사용되며, association이 설정된 endpoint들을 유일하게 식별할 수 있다. Checksum 필드(32 비트)는 Adler-32 알고리즘을 이용한 SCTP 패킷의 무결성과 일관성들을 검사할 수 있게 한다.

각 Chunk들은 또 Chunk Header와 Chunk Data (혹은 Chunk Control)로 구성되었는데, Chunk Header에서 Type 필드는 Chunk 값 필드에 포함된 정보의 타입을 식별하며, Flag 필드는 Chunk 타입에 의해 결정되는데, 특별한 설명이 없으면 전송 상에서 0으로 설정되거나 수신 상에서 무시된다. Length 필드는 Chunk 타입, Chunk 플래그, Chunk Length와 Chunk 값 필드를 포함하는 전체 Chunk의 길이를 표시한다. Chunk Data는 TSN (Transmission Sequence Number)(32비트), Stream ID(16비트), SSN(Stream Sequence Number)(16비트), Protocol ID(32비트)와 User Data(32비트)로 구분된다.

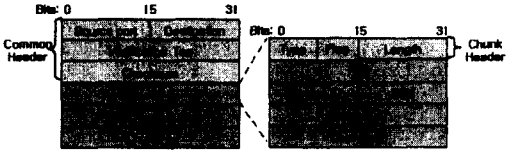


그림 2. SCTP 패킷 포맷
Fig. 2 SCTP packet format

SCTP 패킷 헤더를 압축하기 위하여 ROHC 필드 분류법을 적용하여 헤더 필드를 분류한다. 분류된 필드들은 압축과 압축해제 과정을 위한 중요한 정보인 context(임의의 context는 context 식별자인 CID에 의해 식별된다)를 사용하여 압축 또는 압축해제된다. 분류 결과는 그림 3에 나타낸다.

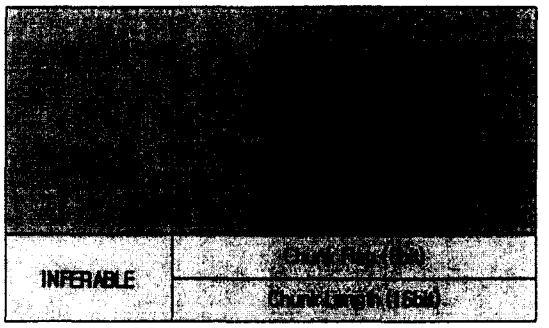


그림 3. SCTP 헤더 필드의 분류
Fig. 3 Classification of SCTP header fields

3.2 SNR-SCTP 필드의 추가

ROHC 방법에 소개된 각 압축 프로파일은 context의 모든 상태를 가장 높은 압축 상태(SO)에서 동작하게 하기 위하여, 스트림의 연속적인 패킷 사이에서 규칙적으로 변화하는 dynamic 필드들을 사용한다. SO 상태에서, 컴프레서는 스트림이 규칙적으로 변할 때만 동작한다. 컴프레서는 스트림이 전개되는 동안에 헤더 필드의 모든 변화를 감시하고 있다가, 규칙성을 발견하게 되면, 원본 dynamic 필드에서 변화된 부분과 암호화된 W-LSB (W-LSB는 오버헤드나 강인성(robustness) 수준상 높은 이익을 가지며, W-LSB 인코딩은 패킷 손실이 있을 때에도 정확하게 압축해제를 할 수 있다)를 보낸다. 컴프레서가 SO 상태에 있다가, 하나 혹은 그 이상의 dynamic 필드가 불규칙적으로 변하게 되면, SO 상태를 떠나 좀 낮은 압축 수준인 FO 상태에 들어가게 된다. FO 상태에서 컴프레서는 원본 필드의 체인지(change)와 danamic 필드들의 변화와 암호화된 W-LSB를 보낸다. 스트림이 다시 규칙적으로 변하면, 컴프레서도 다시 SO 상태에 들어간다. 만약 그렇지 않고 오랜 시간 불규칙적이면 컴프레서는 계속하여 FO 상태에 머물거나 혹은 IR 상태에 들어간다.

각각의 새로운 SCTP 스트림을 위해, 컴프레서는 SNR-SCTP 필드를 dynamic 필드로 설정하여 SCTP 헤더의 dynamic 필드들과 함께 전송한다.

SCTP 패킷을 연속적으로 보낼 때, Chunk 헤더 중의 Type은 변화가 없거나 혹은 변화가 있는 두 가지 경우가 있을 수 있는데 변화가 없는 경우에는 T=0으로 설정함으로서, SNR-SCTP 필드를 1 바이트만 추가한다. 변화가 있는 경우에는 T=1로 설정함으로서, Chunk Type 필드(1 바이트)까지 총 2 바이트의 SNR-SCTP 필드를 추가한다. 이 두 가지 경우의 포맷은 그림 4에 나타낸다. 이 그림에서와 같이, T는 1 비트를 차지하고, f(X1)는 4비트를 차지하며, 나머지 3 비트는 예약(Reserved)되어 있다. 여기서, f(X1)의 값은 다음과 같다:

0000: SCTP 스트림을 구분하지 않는다.
0001: 현재 SCTP 스트림이 ACK 스트림임을 표시한다.
0010: 현재 SCTP 스트림이 Data 스트림임을 표시한다.
0011~1111: 예약되어 있다.

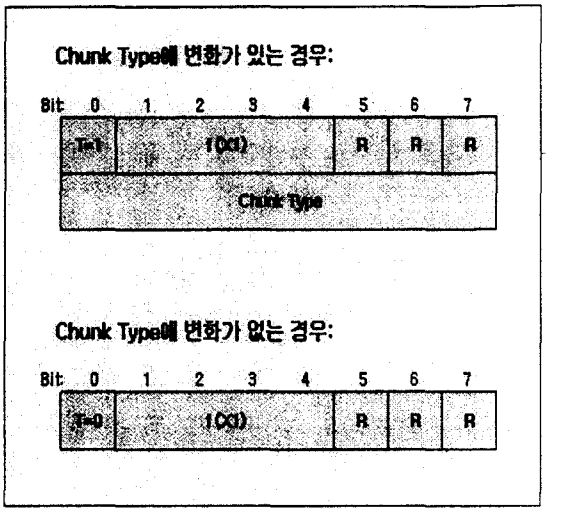
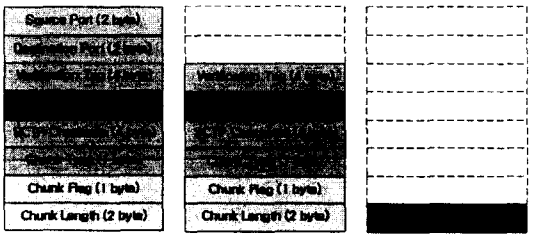


그림 4. SNR-SCTP 필드 포맷
Fig. 4 SNR-SCTP field format

3.3 ROHC-SCTP 패킷 포맷

ROHC-SCTP 패킷 헤더의 포맷은 그림 5(a)의 IR 상태, 그림 5(b)의 FO 상태, 그림 5(c)의 SO 상태로 나누어 볼 수 있다. IR 상태에서 컴프레서는 Full 헤더 정보(17 또는 18 바이트)를 디컴프레서에 전송한다. 압축이 최적의 상태로 진행되어 FO 상태에 진입하면, 컴프레서는 헤더 필드 중에서 STATIC 필드들인 소스 포트 필드와 목적지 포트 필드를 제외한 나머지 필드(13 또는 14 바이트)들을 디컴프레서에 전송한다. 다시 압축 과정이 최적의 상태로 진행되거나 또는 디컴프레서로부터 context 업데이트 정보에 대한 ACK를 받으면 컴프레서는 FO 상태로부터 SO 상태에 진입함으로서 최적의 압축 상태를 나타내게 된다. 이 경우에 SO 상태에는 SNR-SCTP 필드(1 또는 2 바이트)만 존재하게 된다. 결과적으로, SCTP 헤더 17 또는 18 바이트를 1 또는 2 바이트로 압축함으로서, 6 또는 11% 까지 압축할 수 있다는 것을 알 수 있다.



(a) IR 상태인 경우 (b) FO 상태인 경우 (c) SO 상태인 경우
(a) Case of IR state (b) Case of FO state (c) Case of SO state

그림 5. ROHC-SCTP 헤더 포맷
Fig. 5 SNR-SCTP header format

이 논문에서 제안한 ROHC-SCTP 프로파일을 위한 패킷 포맷은 다음과 같은 몇 가지 차이점 외에 ROHC 프로파일 [9]에 사용된 구조와 비슷하다.

프로파일 값은 0x0007이다.

SN RTP 필드는 SNR-SCTP 필드로 대체한다.

모든 압축된(타입 CO) 포맷들은 SNR-SCTP 추가 필드가 존재하여 SCTP 체크섬의 전송을 위해 예약된다.

M 플래그(RTP marker 비트)는 X1 플래그로 바뀐다. 만약 X1 플래그가 1로 설정되면, X1이라는 특별한 확장 필드 즉, SNR-SCTP 필드가 추가된다.

Forward 패킷 포맷 :

IR과 IR-DYN 패킷들은 ROHC[9]에 정의된 IR과 IR-DYN 패킷 포맷을 사용한다. IR 패킷 타입은 context의 static 부분과 dynamic 부분을 전달하고, IR-DYN 패킷 타입은 context의 dynamic 부분만 전달한다.

IR-CR 패킷들은 ROHC-CR[15]에 정의된 ROHC CR 패킷의 일반적인 포맷을 사용한다. 이 패킷 타입은 base context와 다르게 복사된 context의 static 그리고 dynamic 부분과 함께 base context에 reference를 전달한다.

압축된 패킷(CO)들은 ROHC-TCP CO[11]처럼 패킷 헤더에 불규칙적인 것들을 전달한다. 모든 CO 패킷들은 CRC을 운반하고 context를 갱신할 수 있다.

Feedback 패킷 포맷 :

프로파일 7의 feedback 패킷은 ROHC-TCP에 사용된 ACK, NACK, STATIC-NACK 이 세 가지 feedback 포맷들을 사용한다. Forward 포맷에 알맞은 고려사항은 반드시 feedback 포맷까지 동일하게 확장되어야 한다.

Feedback-1은 오직 압축해제의 성공적인 통신(ACK)에만 사용될 수 있다.

Feedback-2는 ROHC-TCP에서처럼, Acktype 값에 더욱 융통성 있게 사용할 수 있다. NACK나 STATIC-NACK에는 반드시 Feedback-2 포맷이 사용된다.

3.4 에러 검출을 위한 ACK 스트림과 data 스트림의 분류

SCTP는 데이터를 여러 개의 다른 스트림으로 나누는 것이 가능하며, 각 스트림은 그 특성에 따라 배달될 수 있고, 다른 스트림과 독자적으로 다루어 질 수 있다. 스트림은 TCP와 마찬가지로 엄격히 순서를 지켜서 전달되도록 정의할 수 있으며, 부분적으로만 순서를 지키거나, 순서와 무관하게 수신 측에 도착하자마자 응용 계층으로 배달되도록 하는 경우에 일부 데이터가 손실되거나 순서에 맞지 않게 수신 측에 도착함으로써 발생하는 헤드 오브 라인 블로킹(Head-of-Line Blocking) 문제도 각 스트림이 독립적으로 동작하므로 해당 스트림에만 영향을 줄 뿐, 다른 스트림에는 영향을 주지 않는다. 그림 6은 호스트A와 B 사이의 멀티 스트림 association을 보여준다.

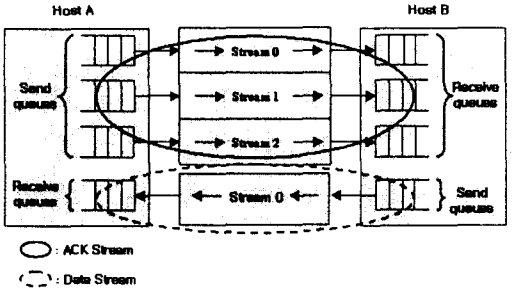


그림 6. SCTP 멀티 스트림 결합
Fig. 6 SCTP multi-stream association

이 association 동안에, 호스트 A는 호스트 B로부터 세 개의 스트림(번호는 0부터 2까지)을 요청받았고, 호스트 B는 호스트 A로부터 하나의 스트림(번호는 0)을 요청받았다. 만약 스트림 1에서 데이터가 손실되었다면, 재전송을 기다리고 있는 동안 오직 스트림 1만 수신자 측에 전송되지 못하고 블록 된다. SCTP 수신자는 다른 스트림들(스트림 0, 스트림 2) 상에서 손실이 없이 도착하는 데이터를 즉시 애플리케이션에 전달할 수 있다. 각 스트림은 이처럼 서로 영향을 주지 않기 때문에, 임의의 SCTP 연결이 확정되었을 때, 이 스트림에 속하는 패킷들이 전송될 수 있다.

더욱 효율적인 압축을 위하여, 서로 독립적으로 분리될 수 있는 SCTP의 멀티 스트림은 전체적으로 볼 때, 그림 6에서처럼 두 가지 단일방향 스트림인

ACK 스트림(호스트A -> 호스트B)과 data 스트림(호스트B -> 호스트A)으로 볼 수 있으며 개별적으로 압축된다. 호스트 B는 호스트 A로부터 압축된 ACK 스트림(0, 1, 2)을 받으면 다시 압축된 data 스트림(0)을 보낸다. 만약 ACK 스트림 상에서 에러가 발생하여 스트림 1을 수신 측에서 받지 못했다면 개별적으로 ACK 스트림 상에서 복구할 수 있으며, 만약 data 스트림 상에서 에러가 발생하여 스트림 0을 송신 측에서 받지 못한 경우에도 개별적으로 data 스트림 상에서 복구를 할 수 있다.

3.5 ACK 스트림과 Data 스트림을 이용한 context 복구 방법

압축 방법에서 가장 중요한 요구사항의 하나가 바로 패킷 손실을 처리하는 능력이다. 헤더 압축에서 context는 컴프레서와 디컴프레서 간의 중요한 정보를 주고받는 것이기 때문에 손실되었을 경우 반드시 복구해야 된다. 실제로, context 복구는 왕복소요시간(RTT)이 지난 후에만 특정된 룰이 없이 복구가 가능하다. 이런 비효율성은 대역폭의 경제적 가치와 서비스 품질 면에서 용납할 수 없는 것이다.

context 복구를 함에 있어서, 보통 두 가지 절차를 거치는데 에러 발견과 패킷 복구이다. ROHC 프로파일 1 (*RTP/UDP/IP* 스트림)에서 이 문제는 CRC와 ROHC 복구 알고리즘[9]을 함께 사용하여 해결하였다.

SCTP 멀티 스트림에서는 그림 7과 그림 8에서처럼 각각 ACK 스트림과 Data 스트림을 이용하여 패킷 손실을 복구한다.

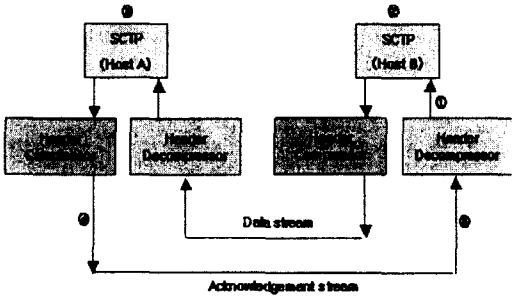


그림 7. ACK 스트림(호스트 A->B)에서 동기화 손실 경우의 context 복구 과정
Fig. 7 SCTP recovery procedure in the case of synchronization loss from host A to host B

그림 7에 대한 설명은 아래와 같다:
① 호스트 A에서 보낸 ACK 패킷은 압축해제가 정확하게 되지 않았다.
② 호스트 B는 SCTP 체크섬에 실패하여, SCTP 조각을 재전송할 것을 호스트 A에 요구한다.
③ 호스트 A는 새로운 ACK 패킷을 전송한다.
④ 호스트 A 컴프레서는 ACK 번호의 0 혹은 부수적인 변화를 발견하고, 호스트 B 디컴프레서의 context를 갱신하기 위하여, IR 혹은 IR-DYN 패킷을 전송한다.
⑤ 호스트 B 디컴프세서의 context는 성공적으로 업데이트 되었다.

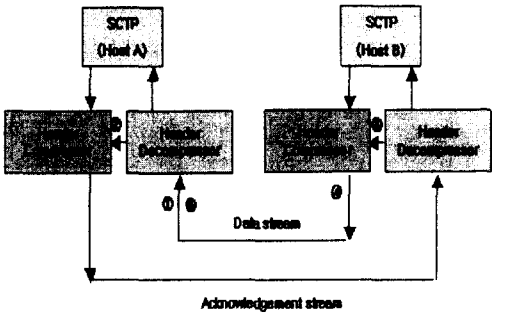


그림 8. Data 스트림(호스트 B->A)에서 동기화 손실 경우의 context 복구 과정
Fig. 8 SCTP recovery procedure in the case of synchronization loss from host B to host A

그림 8에 대한 설명은 아래와 같다:
① 데이터 스트림 패킷은 호스트 A 디컴프레서 측으로의 SCTP 체크섬에 실패한다.
② 호스트 A 디컴프레서는 손상된 조각의 필요사항과 함께 피드백 패킷을 호스트 A 컴프레서에 보내며, 이 피드백 패킷은 호스트 B 디컴프레서에 전송된다.
③ 호스트 B 디컴프레서는 호스트 B 컴프레서에 피드백 정보를 준다.
④ 호스트 B 컴프레서는 새로운 조각을 보낸다.
⑤ Context는 복구된다.

IV. 성능 분석

본 논문의 성능 분석에서는 일반적인 서비스를 위하여 오버헤드와 대역폭에 대한 성능 파라미터들을 설정하였다. 표 1은 ROHC-SCTP의 성능 분석에 활용된 주요한 파라미터들이다.

표 1. 성능 분석에 활용된 파라미터
Table. 1 Parameters utilized in performance

파라미터	의미
f_{sf}	소스 프레임 율
P	페이로드 길이
H	헤더 길이
Hc	압축된 후의 헤더 길이
Bwc	헤더 압축을 하지 않았을 때 전체 서비스에 필요한 대역폭.
Bc	헤더 압축을 했을 때 전체 서비스에 필요한 대역폭.
BH	헤더 전송에 필요한 대역폭.
Bsaving	절약할 수 있는 대역폭.

4.1 각 압축 상태에 따른 오버헤드와 페이로드의 관계 그래프

패킷 오버헤드는 일반적으로 헤더 길이와 헤더 길이에 데이터 길이를 더한 합의 비[10]로 나타낸다. 즉 다음과 같은 수식으로 표현할 수 있다.

$$\text{Packet Overhead} = H/(H+P) \quad (1)$$

$$\text{Packet Overhead\%} = 100\times H/(H+P) \quad (2)$$

헤더 압축 과정은 낮은 상태에서 높은 상태로 전환한다. 즉, IR 상태에서 가장 낮은 압축률, SO 상태에서 최적의 압축률을 나타낸다. 그러므로 각 상태에 따라 헤더 오버헤드 율이 부동할 것이다. 아래에 이 세 가지 압축 상태에 따른 오버헤드에 대해 설명한다.

1) IR 상태인 경우

이 경우에 컴프레서는 Full 헤더를 전송하기 때문에 헤더 오버헤드는 거의 100%에 도달하므로, ROHC를 적용하지 않았을 때의 SCTP 헤더와 동일하다고 볼 수 있다. 때문에 IR 상태에서 오버헤드에 대한 그래프는 큰 의미가 없게 된다.

2) FO 상태인 경우

① Chunk Type에 변화가 없는 경우(T=0)

이 경우에 SCTP 헤더는 4 바이트의 STATIC 필드들은 없어지고 추가된 SNR-SCTP 필드가 1 바이트 있으므로 SCTP 전체 헤더 길이는 13 바이트가 된다. 즉 H=13인 경우 수식 (2)에 근거하여 그림 9와 같은 그래프를 얻어낼 수 있다.

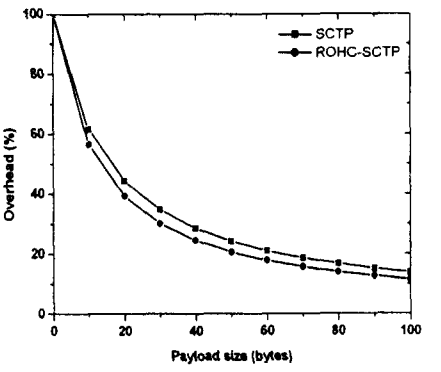


그림 9. FO 상태에서 SCTP와 ROHC-SCTP의 오버헤드 비교(T=0인 경우)
Fig.9 Overhead compression for SCTP and proposed ROHC-SCTP in FO state(case of T=0)

② Chunk Type에 변화가 있는 경우(T=1)

이 경우에 SCTP 헤더는 4 바이트의 STATIC 필드들은 없어지고 추가된 SNR-SCTP 필드가 2 바이트 있으므로 SCTP 전체 헤더 길이는 14 바이트가 된다. 즉 H=14인 경우 수식 (2)에 근거하여 그림 10과 같은 그래프를 얻어낼 수 있다.

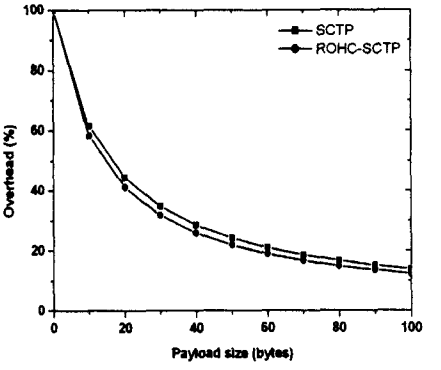


그림 10. FO 상태에서 SCTP와 ROHC-SCTP의 오버헤드 비교(T=1인 경우)
Fig.10 Overhead compression for SCTP and proposed ROHC-SCTP in FO state(case of T=1)

3) SO 상태인 경우

① Chunk Type에 변화가 없는 경우(T=0)

이 경우에 SCTP 헤더는 오직 추가된 SNR-SCTP 필드 1 바이트만 존재하므로, H=1인 경우 수식 (2)에 근거하여 그림 11과 같은 그래프를 얻어낼 수 있다.

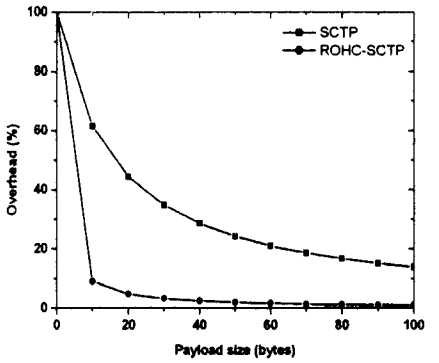


그림 11. SO 상태에서 SCTP와 ROHC-SCTP의 오버헤드 비교(T=0인 경우)

Fig.11 Overhead compression for SCTP and proposed ROHC-SCTP in SO state(case of T=0)

② Chunk Type에 변화가 있는 경우(T=1)

이 경우에 SCTP 헤더는 오직 추가된 SNR-SCTP 필드 2 바이트만 존재하므로, H=2인 경우 수식 (2)에 근거하여 그림 12와 같은 그래프를 얻어낼 수 있다.

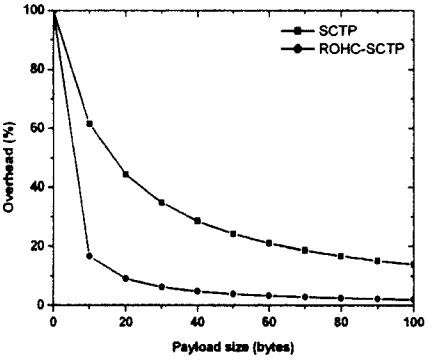


그림 12. SO 상태에서 SCTP와 ROHC-SCTP의 오버헤드 비교(T=1인 경우)

Fig.12 Overhead compression for SCTP and proposed ROHC-SCTP in SO state(case of T=1)

그림 9, 그림 10, 그림 11, 그림 12를 비교하여 볼 때, FO 상태에서는 ROHC-SCTP가 SCTP보다 평균 2.5%의 오버헤드가 더 감소되었으며, SO 상태에서는 평균 25%의 오버헤드가 더 감소되었음을 알 수 있다.

4.2 각 압축 상태에 따른 대역폭과 페이로드의 관계 그래프

패킷 오버헤드는 또한 대역폭(BW)의 효율 상에서도 같은 효과를 나타낸다. 수식으로 표시하면 아래와 같다.

$$BW\ Occupancy = H\times f_{sf}/[(H+P)\times f_{sf}]=B_H/B_{wc} \quad (3)$$

$$BW\ Occupancy\% = 100\times B_H/B_{wc} \quad (4)$$

즉, 오직 헤더 전송에만 필요한 대역폭(B_H)이 헤더 압축이 없을 때 전체 서비스에 필요한 대역폭(B_{wc})에서 차지하는 비율로 대역폭의 점유율(Bw Occupancy%)을 표시할 수 있다.

헤더 압축 방법의 효율성을 나타낼 수 있는 척도는 오버헤드 외에, 대역폭도 중요한 척도이다. 헤더 압축률이 높을수록 절약할 수 있는 대역폭도 더 많다. 이런 관계를 수식으로 표시하면 아래와 같다.

$$B_{saving}=B_{wc}-B_c=(H+P)\times f_{sf}-(H_c+P)\times f_{sf}=(H-H_c)\times f_{sf} \quad (5)$$

수식 (5)에서 보다시피, 헤더 압축률이 높을수록 (즉, 압축된 후의 헤더 길이 H_c가 작아질수록) 절약할 수 있는 대역폭(B_{saving})은 점점 커진다. 절약할 수 있는 대역폭(B_{saving})이 상대적으로 헤더압축을 하지 않았을 때 전체 서비스에 필요한 대역폭(B_{wc}) 가운데서 차지하는 비율을 다음과 같은 수식 (6)으로 나타낼 수 있다.

$$B_{saving\ relative}\% = 100\times B_{saving}/B_{wc} \quad (6)$$

수식 (2), (3)으로부터 Packet Overhead와 BW Occupancy가 같다는 것을 알 수 있다. 그러므로 수식 (4), (5), (6)으로부터 수식 (7)을 얻어낼 수 있다.

$$B_{saving\ relative}\% = 100\times(H-H_c)\times Overhead/B_H \quad (7)$$

수식 (7)에 의해, 아래에 세 가지 압축 상태에 따라 상대적으로 절약할 수 있는 대역폭에 대해 설명한다.

1) IR 상태인 경우

이 경우에 컴프레서는 Full 헤더를 전송하기 때문에 헤더 오버헤드는 거의 100%에 도달하는데, 이런 경우에 대역폭을 거의 절약할 수 없으므로 상

대적으로 절약할 수 있는 대역폭에 대한 그래프도 큰 의미가 없게 된다.

2) FO 상태인 경우

① Chunk Type에 변화가 없는 경우(T=0)

이 경우에 SCTP 헤더는 4 바이트의 STATIC 필드들은 없어지고 추가된 SNR-SCTP 필드가 1 바이트 있으므로 압축된 후의 SCTP 헤더 길이는 13 바이트가 된다. 즉 H=16, Hc=13인 경우 수식 (7)에 근거하여 그림 13과 같은 그래프를 얻어낼 수 있다.

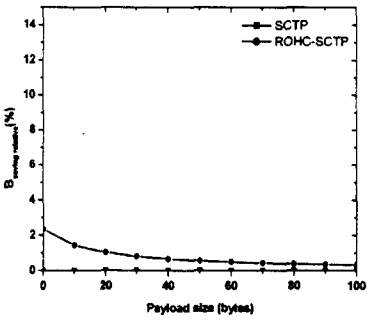


그림 13. FO 상태에서 SCTP와 ROHC-SCTP의 대역폭 비교(T=0인 경우)

Fig.13 Bandwidth compression for SCTP and proposed ROHC-SCTP in FO state(case of T=0)

① Chunk Type에 변화가 있는 경우(T=1)

이 경우에 SCTP 헤더는 4 바이트의 STATIC 필드들은 없어지고 추가된 SNR-SCTP 필드가 2 바이트 있으므로 압축된 후의 SCTP 헤더 길이는 14 바이트가 된다. 즉 H=16, Hc=14인 경우 수식 (7)에 근거하여 그림 14와 같은 그래프를 얻어낼 수 있다.

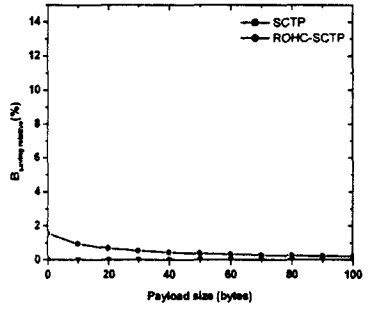


그림 14. FO 상태에서 SCTP와 ROHC-SCTP의 대역폭 비교(T=1인 경우)

Fig.14 Bandwidth compression for SCTP and proposed ROHC-SCTP in FO state(case of T=1)

3) SO 상태인 경우

① Chunk Type에 변화가 없는 경우(T=0)

이 경우에 SCTP 헤더는 오직 추가된 SNR-SCTP 필드 1 바이트만 존재하므로, H=16, Hc=1인 경우 수식 (7)에 근거하여 그림 15와 같은 그래프를 얻어낼 수 있다.

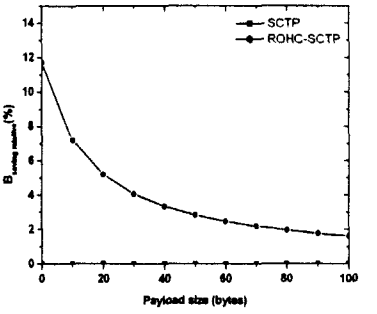


그림 15. SO 상태에서 SCTP와 ROHC-SCTP의 대역폭 비교(T=0인 경우)

Fig.15 Bandwidth compression for SCTP and proposed ROHC-SCTP in SO state(case of T=0)

① Chunk Type에 변화가 있는 경우(T=1)

이 경우에 SCTP 헤더는 오직 추가된 SNR-SCTP 필드 2 바이트만 존재하므로, H=16, Hc=2인 경우 수식 (7)에 근거하여 그림 16과 같은 그래프를 얻어낼 수 있다.

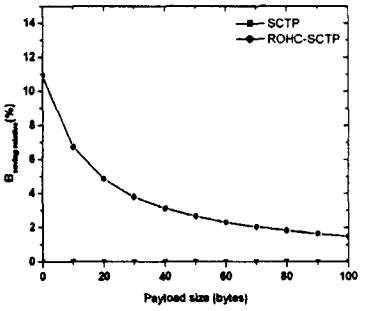


그림 16. SO 상태에서 SCTP와 ROHC-SCTP의 대역폭 비교(T=1인 경우)

Fig.16 Bandwidth compression for SCTP and proposed ROHC-SCTP in SO state(case of T=1)

그림 13, 그림 14, 그림 15, 그림 16을 비교하여 볼 때, SCTP는 헤더 압축이 없으므로 대역폭을 거의 절약할 수 없고, 제안된 ROHC-SCTP는 FO 상태에서는 평균 0.7% 정도의 대역폭을 절약할 수

있고 SO 상태에서는 평균 5% 정도 절약할 수 있다는 것을 알 수 있다.

V. 결 론

이 논문은 기존의 전송 계층 프로토콜인 TCP나 UDP보다 많은 장점을 가지고 있는 새로운 전송 계층 프로토콜 SCTP에 ROHC 압축 기법을 적용하여, 높은 에러 율(BER)과 긴 왕복소요시간(RTT)을 갖고 있는 무선망에서 헤더 압축을 통하여 대역폭을 크게 절약할 수 있는 새로운 ROHC 프로파일 7을 제안했다. 제안된 프로파일은 SNR-SCTP 라는 1 또는 2 바이트 길이의 필드를 SCTP 패킷 헤더 중에 추가함으로서 SCTP 스트림을 ACK 스트림과 data 스트림으로 나누어 헤더 압축을 한다. 또한 이러한 스트림의 구분은 동기화 손실이 발생했을 경우, 각 흐름별로 동기화 손실에 대한 복구를 할 수 있다. 결과적으로, 제안된 ROHC-SCTP 는 최적의 압축 상태인 경우 SCTP 보다 평균 25% 정도의 오버헤드를 더 감소할 수 있으며, 대역폭 또한 5% 정도 절약할 수 있다.

참고문헌

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real time applications, RFC 1889," January 1996.

[2] J. Postel, "User Datagram Protocol", IETF RFC 768, Aug. 1980.

[3] W. Boulevard, "Transmission Control Protocol", IETF RFC 793, Sep. 1981.

[4] R. Stewart et al., "Stream Control Transmission Protocol", IETF RFC 2960, Oct. 2000.

[5] TSG-SA-WG2, TSG-GERAN and TSG-RAN, "Header compression for optimized voice bearers," Joint Meeting, 3GPP, Aug. 2001.

[6] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links", RFC 1144, Feb. 1990.

[7] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression" RFC 2507, Feb. 1999.

[8] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, Feb. 1999.

[9] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, "RObust Header Compression: ROHC: Framework and four profiles: RTP, UDP, ESP, and uncompressed," RFC 3095, Tech. Rep., July 2001.

[10] G. Boggia, P. Camarda, and V.G. Squeo, "ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems," in Proceedings of the IEEE International Conference on Communications (ICC), vol. 5, 2002, pp. 3271-3278.

[11] G. Pelletier, L-E. Jonsson, M. West, R. Price, K. Sandlund, "Robust Header Compression(ROHC): A Profile for TCP/IP (ROHC-TCP)" <draft-ietf-rohc-tcp-07.txt>, Jul. 2004.

[12] G. Pelletier, Q. Zhang, L.-E. Jonsson, H. Liao, and M. West, "RObust Header Compression(ROHC): TCP/IP Profile (ROHC-TCP)," Nov. 2002, RFC-draft, work in progress.

[13] 고석주, "SCTP 표준기술 분석 및 전망", 전자통신동향분석 제18권 제 3호, 2003년 6월.

[14] F. Fitzek, S. Hendrata, P. Seeling, M. Reisslein, "Header Compression Schemes for Wireless Internet Access", Jul. 2003.

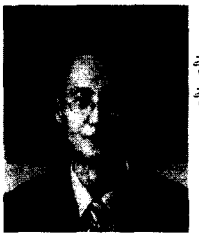
[15] Pelletier, G., "Robust Header Compression (ROHC): Context Replication for ROHC profiles", Internet Draft (work in progress), <draft-ietf-rohc-context-replication-03.txt>, July 2004.

저자 소개

송희옥(Hee-Ok Song)

전자공학과 학사
전파공학과 대학원 재학중

※관심분야 : 컴퓨터 네트워크, 유무선 헤더 압축



최문석(Mun Suk Choi)

충북대 전자계산학과 석사 졸업
충북대 전파공학과 박사과정

※관심분야 : 로봇 주행, Network QoS, Network Security, Multimedia 통신

최성곤(Seong-Gon Choi)

ETRI, 정보보호연구단, 선임연구원
충북대학교 정보통신공학과 교수 (전임강사)
BcN 표준전략협의회 위원 (TTA, Mobility 분야).
ITU-T, NGN FG, Editor(Mobility 분야) 활동중.
TTA 선정 국제 표준화 전문가 활동중.

※관심분야 : 초고속 통신망 분야, Network에서의 Mobility 기술 연구, 4세대 이동통신 기술, BcN 네트워크 엔지니어링 (Mobility, QoS/SLA, 프로토콜), 네트워크 프로토콜 및 성능 분석, 휴대인터넷

신병철(Byung-Cheol Shin)

KAIST 전자공학과 석사
KAIST 전자공학과 박사
KAIST 교수
충북대학교 교수

※관심분야 : 멀티미디어통신, 무선랜

이인성(Insung Lee)

한국통신 연구개발단 전임연구원
Texas A&M Univ., Dept. of Electrical Eng.
ETRI 이동통신기술연구단 선임연구원
충북대학교 전기전자공학부 교수

※관심분야 : 이동/위성통신 시스템, 신호처리