

---

# 무선망에서 효율적인 멀티미디어 전송을 위한 헤더압축 알고리즘 연구

윤성열\* · 박석천\*\*

A Study on Header Compression Algorithm for the Effective Multimedia Transmission  
over Wireless Network

Sung-Yeol Yun\* · Seok Cheon Park\*\*

## 요 약

MoIP는 IP위에 다양한 멀티미디어를 전송가능하게 해주는 기술인데, 기존 음성 서비스에 비해 큰 대역폭을 요구하며, 무선 환경에서의 무선 자원은 이미 한계점에 도달하고 있다. 따라서 이런 문제점을 해결하기 위한 방법으로 헤더 압축에 대한 연구가 많이 되고 있다. SCTP 프로토콜을 이용한 헤더압축은 ROHC-SCTP가 연구되었는데, ROHC-SCTP는 ROHC알고리즘과는 다른 형식의 패킷구조를 가지고 있고, SCTP의 특징을 적용하려면 헤더압축의 성능이 많이 떨어진다는 단점을 가지고 있다. 따라서 본 논문에서는 위의 문제점을 해결하기 위해 향상된 헤더압축 알고리즘을 설계하였다. 본 논문에서 제안한 알고리즘을 평가하기 위해 NS-2 시뮬레이션 환경에서 헤더압축 동작을 모델링하였다. 평가 결과 본 논문에서 설계한 알고리즘이 ROHC-SCTP 알고리즘에 비해 오버헤드율이 낮았고, 데이터의 타입이 많이 변화할 때 총 헤더의 크기가 더 작음을 확인하였다.

## ABSTRACT

MoIP is technology to transmit a variety of multimedia over IP, but compared to traditional voice services require greater bandwidth and radio resources in a wireless environment has already reached the limits. Therefore, as a way to resolve this issue for header compression is a lot of research. SCTP protocol header compression using ROHC-SCTP has been research, ROHC-SCTP packet structure of the ROHC algorithm with different types and, SCTP header compression to apply the characteristics of the poor performance of many of these have drawbacks. Therefore, in this paper to solve these problems better header compression algorithm was designed. In this paper, the proposed algorithm to evaluate the NS-2 simulation environment was modeled on the header compression operation. Evaluation results, the algorithm designed in this paper compared to ROHC-SCTP algorithms determine the overhead rate was low, the data types vary a lot better when the total header size was small.

## 키워드

무선망, 멀티미디어 전송, 헤더 압축

## Key word

Multimedia over IP, Stream Control Transmission Protocol, Header Compression

---

\* 경원대학교

\*\* 경원대학교 정교수(교신저자)

접수일자 : 2009. 08. 10

심사완료일자 : 2009. 11. 07

## I. 서 론

최근 유선 및 무선 통신 인프라의 급속한 발달은 다양한 서비스를 지원할 수 있게 개발 되었다. 음성 통신을 비롯하여 각종 데이터 및 영상전송을 통한 다양한 서비스의 구현이 가능해 짐에 따라 사용자는 더욱 고도화된 서비스를 필요로 하게 되었다. 대표적으로 MoIP (Multimedia over IP)가 있으며, 이는 IP위에 다양한 멀티미디어를 전송가능하게 해주는 기술이 [1]. 그러나 MoIP는 기존 음성 서비스보다 훨씬 더 큰 대역폭을 요구하며, 무선 환경의 무선 자원은 이미 한계점에 도달하고 있다. 무선 환경에서 중요한 이슈가 되고 있는 대역폭의 문제는 일반적으로 유선 환경에서보다 많이 발생되는데, 이런 문제점들을 해결하기 위해서는 헤더를 압축하거나 제거/생성을 하는 기술들이 있고 이런 기술들 중에 가장 이슈가 되는 기술이 헤더압축 기술이다 [2].

기존의 헤더압축 방법들은 유선망을 기반으로 설계되었기 때문에 무선망에서는 유선망보다 어려움이 확연히 높고 또한 갑자기 데이터가 전송되는 특성이 있기 때문에 이를 고려한 ROHC(RObust Header Compression) 기술이 등장하였다[3][4]. ROHC 기술은 TCP/UDP[5][6]를 기반으로 설계되었기 때문에 이를 SCTP(Stream Control Transmission Protocol)에 그대로 적용할 수 없다[7]. 따라서 무선망을 위한ROHC-SCTP가 연구되었는데, 이 프로토콜은 SCTP의 특성을 고려하여 설계된 프로토콜로써 ROHC와 유사한 방법을 채택하였다[8].

그러나 ROHC-SCTP는 SCTP의 특성인 멀티스트리밍을 지원하기 위해 데이터의 변화를 감지하는 필드를 추가함으로써 ROHC에서 정의한 패킷의 구조를 유지할 수 없다. 또한 데이터가 수시로 변함에 따라 압축효율의 급격한 저하가 발생하기 때문에 헤더압축의 성능이 많이 떨어지는 단점을 가지고 있다.

따라서 본 논문에서는 무선망 환경에서 패킷 통신에서 발생하는 패킷의 헤더의 오버헤드를 줄이고 대역폭을 절약하기 위해 향상된 헤더압축 알고리즘을 설계하고 평가하였다.

## II. 관련 연구

### 2-1. 헤더압축의 기존 연구

기존에 연구되던 헤더압축은 주로 TCP/UDP의 패킷을 압축하는 방법이다. 무선에서 많이 쓰이는 ROHC를 기반으로 연구되었으며 ROHC-SCTP헤더 포맷이 제안되었다.

#### 2.1.1 ROHC

ROHC는 무선망의 특징인 높은 에러율과 Burst한 패킷에 특화된 알고리즘으로 송신측은 3가지의 압축 상태를 가진다[9]. IR 상태, FO 상태, SO 상태로 구성되는데 그 설명은 다음과 같다[10].

- IR(Initialization and Refresh) : 초기화와 STATIC, DYNAMIC 부분을 전송하는 상태, 비압축상태의 패킷이다.
- FO(First Order) : DYNAMIC에 해당하는 부분을 전송하는 상태로 변화하는 값을 전송한다.
- SO(Second Order) : 가장 효율적인 압축상태를 나타낸다. 대부분의 정보를 이미 전송했기 때문에 유저의 데이터를 전송한다.

각 상태별로 생성하는 패킷의 구조와 크기가 다른데 IR 상태에서 생성하는 패킷을 IR 패킷, FO 상태에서 생성하는 패킷을 FO 패킷, SO 상태에서 생성하는 패킷을 SO 패킷이라 부르고 각 패킷의 구조는 그림 1과 같다.

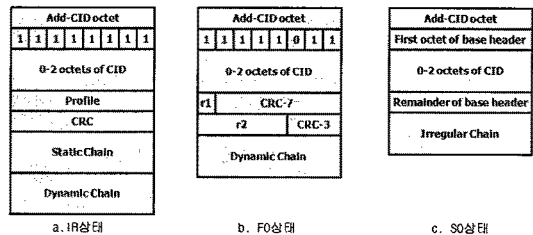


그림 1. IR 상태, FO 상태, SO 상태 패킷  
Fig. 1 IR State, FO State, SO State Packet

패킷은 SO 상태일 때 가장 헤더의 크기가 작는데, 이 때가 헤더압축 알고리즘에서 가장 압축 효율이 좋음을

나타낸다. 그림 2는 송신측에서 ROHC 압축상태의 천이도이다.

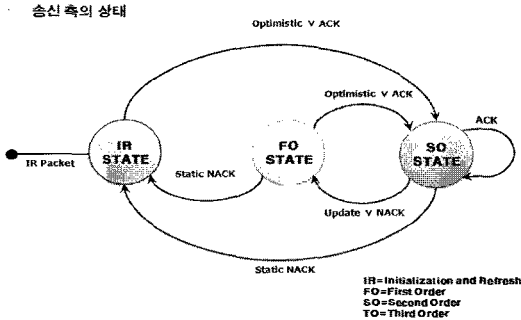


그림 2. ROHC 압축상태 천이도  
Fig. 2. ROHC Compression State Transition Diagram

이렇게 압축된 패킷은 수신측까지 전송되고 수신측은 송신측의 정보를 알고 있어야 압축했던 내용을 해석할 수 있다. 따라서 수신측도 송신측과는 별도로 3가지 상태를 가지게 된다. NC 상태, SC 상태, FC 상태로 구성되며 다음은 3가지 상태의 설명이며 그림 3은 수신측에서 ROHC 압축해제상태의 천이도이다[11].

- NC(No Context) : 압축해제의 초기 상태이다.
- SC(Static Context) : FC 상태에서 정상적인 압축해제를 할 수 없을 때 DYNAMIC 필드가 필요한 상태이다.
- FC(Full Context) : NC 상태에서 정상적으로 압축해제를 하면 FC 상태로 변경되고, 이때는 STATIC 필드와 DYNAMIC 필드를 해석하는데 필요한 모든 정보가 있는 상태이다.

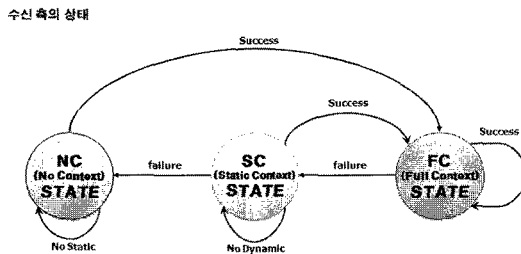


그림 3. ROHC 압축해제 상태 천이도  
Fig. 3. ROHC De-compression State Transition Diagram

2.1.2 ROHC-SCTP

ROHC-SCTP는 ROHC를 기반으로 만든 알고리즘으로 SCTP를 지원할 수 있는 방법이다. SCTP의 특징 중 Chunk라는 필드가 있는데, 이 Chunk 필드는 각 응용 프로그램이 하나의 패킷에 자신들의 데이터를 올릴 때 사용한다. ROHC-SCTP는 SCTP의 패킷 구조에 Chunk Type이 변화했을 때 이를 감지할 수 있는 SNR-SCTP(Sequence Number ROHC-SCTP) 필드를 추가하고, 변화가 감지될 여지를 남겨두는 방법을 제시하였다. 그림 4는 SCTP의 필드구조를 나타낸다[10][12].

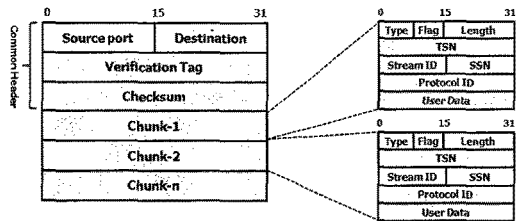


그림 4. SCTP의 필드구조  
Fig. 4. Field Structure of SCTP

SCTP의 패킷 중 Chunk Header는 Type, Flag, Length, TSN(Transmission Sequence Numbers), Stream ID, SSN(Stream Sequence Numbers), Protocol ID 필드가 있다. SCTP는 Chunk마다 독립적으로 스트리밍을 가질 수 있기 때문에 데이터 전송 중에도 수시로 Chunk가 변한다. 따라서 헤더압축을 하기 위한 Chunk 내부에 있는 헤더도 함께 변하게 된다.

SCTP에서 Chunk의 값이 변하게 되면 각 Chunk에 해당하는 필드는 DYNAMIC과 INFERRED로 구분되는데 INFERRED 필드는 전송하지 않는 필드이고 DYNAMIC에 해당하는 Chunk Type 부분이 변하게 된다. 이때, SO 상태에서 FO 상태로 천이하여 DYNAMIC 필드를 재전송하는 것으로 적용할 수 있다.

SCTP의 Chunk 값의 변화를 감지하기 위해서는 각 패킷에 그 변화를 감지할 수 있는 방법이 필요한데 SNR-SCTP 필드를 추가함으로써 이를 해결했다. 그림 5는 ROHC-SCTP의 헤더 포맷이다[8].

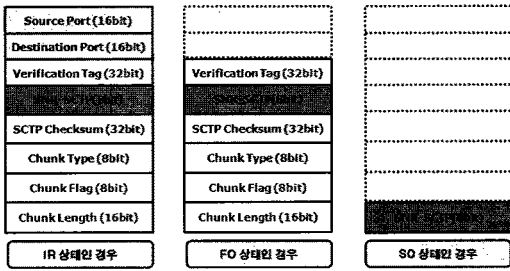


그림 5. ROHC-SCTP 헤더 포맷  
Fig. 5. Header Format of ROHC-SCTP

### 2.2 기존 연구의 문제점

ROHC-SCTP 알고리즘은 ROHC 알고리즘을 SCTP에 적용할 수 있게 설계한 헤더압축 알고리즘으로 SCTP의 특징을 적용시킬 수 있다. 하지만 SCTP의 Chunk의 변화를 감지하기 위하여 8bit의 필드를 추가하여 본래의 ROHC 패킷 크기보다 커진다는 점과, Chunk의 변화가 있을 때 에러가 발생하지 않았음에도 FO 상태로 천이하기 때문에 압축 효율이 떨어질 수 있다. 따라서 본 논문에서는 필드의 추가 없이 새로운 상태를 정의하고, 그 상태에서 발생하는 패킷을 설계 하였다. 또한 기존의 연구보다 높은 효율의 압축상태를 유지하기 때문에 발생하는 오버헤드의 비율도 낮출 수 있다.

## III. SCTP 프로토콜의 헤더압축 알고리즘 설계

제안하는 헤더압축 알고리즘은 ROHC를 기반으로 기존의 TCP 및 UDP에서 적용되던 알고리즘을 새로운 프로토콜인 SCTP에 적용가능하게 설계하였다.

우선 IP의 패킷 헤더 필드를 분류하고, SCTP의 패킷 헤더 필드를 분류한 후에 압축을 위한 필드와 압축이 되지 않는 필드로 구분한다. 그리고 압축상태에서 SCTP만의 특성인 멀티 스트리밍의 특성으로 인한 Chunk Data의 변화를 감지할 수 있는 방법도 설계한다.

### 3.1 헤더압축 알고리즘 설계시 고려사항

제안하는 헤더압축 알고리즘의 고려사항은 다음과 같다.

첫째, 헤더압축 방법이 현존하는 인프라에 적용가능

해야 한다.

헤더의 압축/해제의 결과가 원래의 헤더와 동일해야 한다. 만약 헤더의 압축/압축해제 후 결과 헤더가 다를 경우 이를 폐기하고 오류를 정정하거나 새로운 데이터를 재전송 받아야 한다. 또한 기존에 인프라인 IP나 SCTP에 구현 방식과 동일해야 한다.

둘째, SCTP만의 특정 구조를 반영할 수 있어야 한다.

SCTP 패킷에서의 고유한 구조를 적용시켜야 하는데, 기존의 TCP/UDP와는 다르게 헤더의 크기가 뒤에 따라 붙는 Chunk의 개수에 따라 가변한다는 특징이 있다. 또한 SCTP의 멀티스트리밍의 특징을 이용하여 적용되어야 한다.

셋째, 성능에 대한 요구사항이 필요하다.

헤더압축의 성능은 전체 패킷에서의 낮은 오버헤드를 제공하는 것, 시스템의 운영에 큰 지연이 없어야 한다는 것이 있다.

### 3.2 헤더압축 알고리즘의 필드 분류

헤더압축을 하기 위해서는 ROHC의 필드 분류법을 적용하여 헤더 필드를 분류하여야 하며, 이렇게 분류되는 필드들은 STATIC 필드, DYNAMIC 필드, INFERABLE 필드로 구분된다. STATIC 필드는 동일 스트림에 속하는 패킷 헤더 내에서 항상 고정된 필드로써 처음에 정상적으로 송수신이 되면 생략할 수 있는 가장 압축효율이 좋은 필드이다. 다음은 DYNAMIC 필드로 동일 스트림에 속하는 패킷 헤더 내에서 변할 수 있는 필드를 뜻한다. 마지막으로 INFERABLE 필드는 STATIC 필드와 DYNAMIC 필드로 추론할 수 있는 필드를 뜻한다.

### 3.3 SCTP에서의 헤더압축 알고리즘 설계

IP 헤더 패킷은 기본 160bit의 크기를 가지고 있으며, 헤더의 구조는 그림 6과 같다.

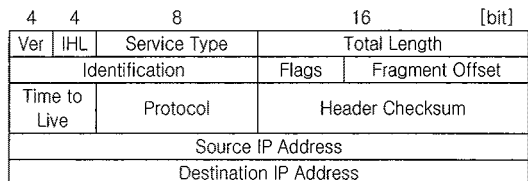


그림 6. IP 헤더 구조  
Fig. 6. IP Header Structure

IP 헤더에서 각각의 필드를 분류하면 표 1과 같다[4].

표 1. 분류된 IP 헤더와 헤더 크기  
Table 1. Classified IP Header and Header Size

분류된 필드	헤더 필드	용량(bit)	
STATIC	Version	4	88
	Header Length	4	
	Flags	3	
	Fragment offset	13	
	Source IP Address	32	
	Destination IP Address	32	
DYNAMIC	Time to Live	8	32
	Service Type	8	
	Identification	16	
INFERRED	Header Checksum	16	40
	Protocol	8	
	Total Length	16	

SCTP 헤더는 기존의 TCP/UDP와는 다르게 스트림의 개수에 따라 헤더의 크기가 변한다. 이때 스트림의 개수나 종류를 표현해주는 Chunk 필드가 있는데 Chunk 부분을 제외한 부분이 Common Header이고, 각 Chunk 내부에는 Chunk Header와 Chunk Body 부분이 있다. 분류된 SCTP 헤더와 헤더 크기는 표 2와 같다[8].

표 2. 분류된 SCTP 헤더와 헤더 크기  
Table 2. Classified SCTP Header and Header Size

분류된 필드	헤더 필드	용량(bit)	
STATIC	Source Port	16	32
	Destination Port	16	
DYNAMIC	Verification Tag	32	72
	Checksum	32	
	Chunk Type	8	
INFERRED	Chunk Flag	8	24
	Chunk Length	16	

STATIC 필드에는 Source Port와 Destination Port가 포함된다. DYNAMIC 필드에는 Verification Tag, Checksum, Chunk Type이 포함된다. 마지막으로 INFERRED 필드에는 Chunk Flag와 Chunk Length가 포함되는데, 이 두 개의 필드는 Chunk Type 필드로 유추가 가능하다.

위에서 분류된 IP헤더와 SCTP헤더를 각각 STATIC, DYNAMIC, INFERRED 필드에 넣고, 이를 통해 ROHC 패킷과 동일한 구조를 가진 IR패킷, FO 패킷, SO 패킷에 해당하는 부분에 삽입한다. 본 논문에서는 Chunk의 변화만 업데이트 하여 전송할 수 있는 TO(Third Order) 패킷과 TO 상태를 제안한다. 제안하는 TO 상태는 그림 7과 같이 정의한다.

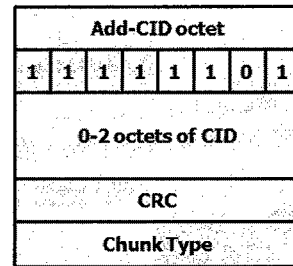


그림 7. TO 상태 패킷  
Fig. 7. TO State Packet

TO 상태의 패킷은 기존에 FO 상태의 패킷(DYNAMIC 필드를 포함하는)을 기반으로 설계하였으며, DYNAMIC 필드 전체를 재전송 하지 않고 Chunk Type 필드만 재전송 한다. 따라서 FO 패킷보다 많은 오버헤드를 감소시킬 수 있다. 그림 8은 제안하는 송신측의 압축 상태를 나타낸 것이다.

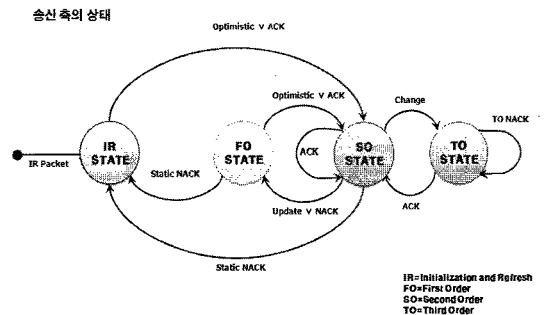


그림 8. 제안하는 송신측 헤더압축 상태 천이도  
Fig. 8. Propose Header Compression State Transition Diagram at Sender

기존의 ROHC-SCTP에서는 송신측에서 SO 상태에서 최적의 압축효율을 나타내고 있을 때 회선의 오류가 발생하지 않았음에도 빈번한 업데이트 요구로 FO 상태로

천이되기 때문에 발생하는 압축효율의 저하가 발생하지만 본 논문에서 제안하는 알고리즘은 FO 상태로 천이하는 대신, **Chunk**의 변화를 감지하는 단계인 TO 상태로 천이하고, TO 패킷을 전송한다. 이 상태에서는 기본적인 동작구조는 FO 상태와 동일하나, 보내는 패킷이 TO 패킷이라는 차이점이 있다.

TO 상태에는 수신측의 응답 메시지가 ACK일 경우 SO 패킷을 전송하고 SO 상태로 천이한다. 하지만 ACK가 아닌 TONACK 패킷을 받을 경우 TO 패킷을 재전송하고 TO 상태를 유지한다. 그림 9는 TO 상태의 동작 알고리즘이다.

```

BEGIN
WHILE SendData Exist DO
  Packet Receive from Receiver
  IF Received Packet is ACK THEN
    SO Packet Send to Receiver
  ELSE
    TO Packet Send to Receiver
  ENDIF
ENDWHILE
END
    
```

그림 9. TO 상태의 동작 알고리즘  
Fig. 9. Algorithm of TO State

수신측에서는 ROHC와 동일한 알고리즘으로 동작한다. 단, FC 상태일 때만 TO 패킷을 입력받고, 동일한 FC 상태를 유지하는 부분만 추가된다. 그림 10은 수신측에서의 압축해제상태 천이도이다.

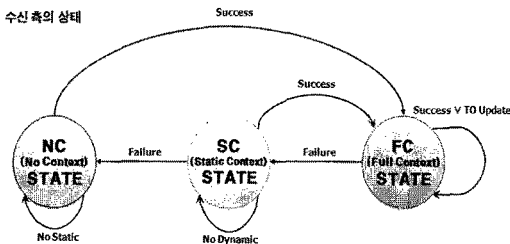


그림 10. 제안하는 수신측의 헤더압축해제 상태 천이도  
Fig. 10. Propose Header Compression State Transition Diagram at Receiver

FC 상태에서 수신측은 송신측의 TO 패킷을 전송받고, 이때 전송받은 패킷으로 헤더 정보의 업데이트를 수행한다. 성공적으로 업데이트가 되면 ACK를, 실패하면 TONACK 패킷을 송신측에 전송하고 FO 상태를 유지한다.

#### IV. SCTP 프로토콜의 헤더압축 알고리즘 평가

##### 4.1 성능 평가 모델

본 논문에서 제안하는 알고리즘은 무선망에서 강한 ROHC 알고리즘을 기본으로 하여 SCTP 프로토콜에 적용할 수 있는 알고리즘을 설계하였으며, 이를 평가하기 위하여 그림 11과 같은 성능 평가 모델을 제시하고, NS2 2.31에서 시뮬레이션 하였다[13].

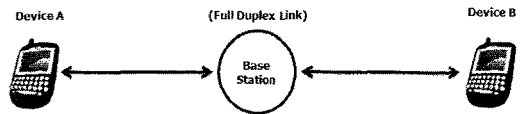


그림 11. 성능 평가 모델  
Fig. 11. Performance Evaluation Model

먼저 Device A와 Device B는 무선 노드로 Base Station에 서로 연결되어 있다. 두 개의 Device는 SCTP 프로토콜로 연결되어 있으며, 발생하는 트래픽은 FTP이다. 시뮬레이션 환경은 표 3과 같다.

표 3. 시뮬레이션 환경 정의  
Table. 3. Definition of Simulation Environment

항목	내용
Simulation Tool	Ns2 - 2.31
Flow Data	SCTP/FTP
Link	Full-Duplex-link 5Mb 300ms
Node	2 units
BaseStation	1 unit
Wireless Protocol	IEEE 802.16

제안하는 알고리즘의 성능 평가기준은 다음과 같이 정의하였다.

- 시간의 변화에 따른 오버헤드율과 총 헤더사이즈 측정 : 시간 값이 변화함에 따라 생성되는 패킷의 개수를 측정하고, 측정된 패킷의 개수를 통하여 발생하는 총 헤더필드의 크기를 계산한다. 계산된 헤더필드의 크기를 발생한 모든 패킷의 평균 오버헤드도 계산하여 결과 값을 이용한다.
- 패킷 헤더 변화율에 따른 헤더사이즈 측정 : 전송하는 패킷의 헤더가 변화함에 따라 생성되는 패킷의 개수를 측정하고, 측정된 패킷의 개수를 이용하여 발생하는 헤더필드의 크기를 이용한다.

위의 두 가지의 성능평가기준으로 ROHC-SCTP와 비교한다. 첫 번째의 성능평가기준으로 측정하기 위해 일정한 트래픽을 발생시키는 무선망에서 Device A에서 Device B로 패킷이 이동될 때 Device A에서 보내는 패킷을 기준으로 캡처하였다. 두 번째의 성능평가기준으로 측정하기 위해 패킷 타입의 변화율을 조정하면서 180초 동안 Device A에서 보내는 패킷을 캡처하였다.

**4.2 SCTP 프로토콜의 헤더압축 알고리즘 평균 오버헤드 비교**

본 논문에서는 제안한 SCTP 프로토콜의 헤더압축 알고리즘이 최소의 평균 오버헤드 값을 가져야 한다. 이는 패킷의 전송이 진행됨에 따라 점점 누적되는 전체의 헤더 크기를 이용하여 구할 수 있다. 이를 측정하기 위해 NS2 시뮬레이터를 이용하였다. 이번 시뮬레이션에서는 회선에서 에러가 나지 않는다는 가정 하에 200초 동안 발생하는 모든 패킷의 수를 측정 하였다.

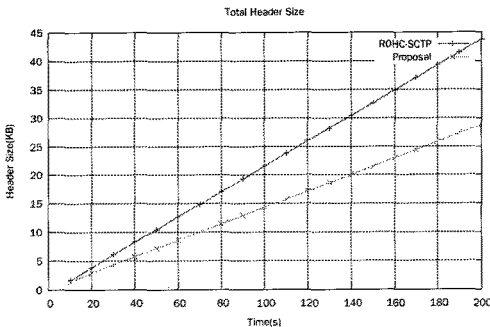


그림 12. ROHC-SCTP와 제안알고리즘을 적용하였을 때 측정된 총 헤더의 크기  
Fig. 12. Total Header Size when Apply ROHC-SCTP and Propose Algorithm

그림 12는 ROHC-SCTP와 제안한 알고리즘을 적용하였을 때 각각 측정된 패킷수와 패킷의 크기를 연산하여 산출한 결과인 총 헤더의 크기를 나타낸 그래프이다.

그림 12에서 ROHC-SCTP와 제안한 알고리즘을 비교 하였으며, 시간이 지남에 따라 기존의 알고리즘은 SO-FO를 왕복하면서 패킷을 생성하고, 제안한 알고리즘은 SO-TO를 왕복하면서 패킷을 생성하므로 그들의 차이만큼 그래프 상에서 차이가 발생한다.

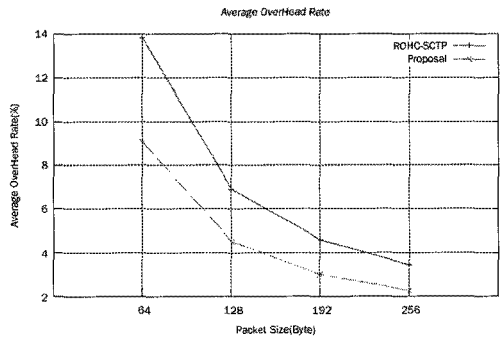


그림 13. ROHC-SCTP와 제안 알고리즘의 평균 오버헤드 비율  
Fig. 13. Average Overhead Rate of ROHC-SCTP and Propose Algorithm

그림 13은 ROHC-SCTP와 제안한 알고리즘의 평균 오버헤드 비율을 나타낸 그래프이다. 패킷 크기를 64~256Byte로 설정하여 각각의 평균 오버헤드 비율을 계산 하였다. 결과 값을 보면 패킷의 크기가 64Byte일때 제안한 방법이 기존의 방법에 비해 평균 5% 오버헤드가 감소 되었음을 알 수 있다.

**4.3 패킷 타입의 변화에 따른 헤더압축 알고리즘 성능 비교**

본 논문에서 제안하는 알고리즘의 또 다른 평가 방법으로 패킷의 타입의 변화에 따른 총 헤더의 크기를 측정 할 수 있다. ROHC-SCTP는 패킷의 타입이 변화하면 헤더압축의 최고 상태인 SO 상태에 머물러 있지 않고 FO 상태로 천이되게 되는데, 이때 전체적인 압축효율이 저 하된다.

이를 NS2를 통해 시뮬레이션하여, 측정된 두 가지 알고리즘의 총 패킷으로 헤더크기를 계산할 수 있다. 그 결과 값을 그래프로 표현하면 그림 14와 같다.

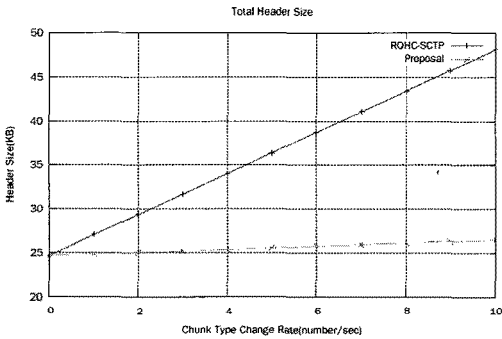


그림 14. 패킷 헤더의 변화에 따른 헤더 크기 비교  
Fig. 14. Header Size comparison on Packet Header Change

측정 결과에 따르면 패킷의 변화율이 초당 2번씩 변화하면 평균 24%의 총헤더 크기를 줄일 수 있고, 초당 10번씩 변화하면 평균 55%의 총헤더 크기를 줄일 수 있다. 패킷의 변화율이 증가함에 따라 ROHC-SCTP는 헤더의 크기가 급격하게 증가하지만, 제안한 알고리즘은 헤더 크기의 변화가 많지 않음을 확인할 수 있다.

## V. 결론

최근 무선인프라의 확산으로 한정된 대역폭에서 많은 데이터를 전송하고자 헤더압축 알고리즘의 연구가 활발히 진행되고 있다. 그중 무선 환경을 고려하여 설계된 ROHC 헤더압축 알고리즘은 TCP 및 UDP를 기반으로 제안되었다. 하지만 MoIP로의 기술이 진화함에 따라 무선환경의 전송 프로토콜인 SCTP가 각광받으면서 이 프로토콜의 헤더압축 알고리즘이 연구되고 있다. 특히 SCTP의 특징을 적용할 수 있으면서 헤더압축을 하기 위한 ROHC-SCTP가 연구되었으며, 기존의 ROHC 패킷에 헤더의 변화를 감지하는 8비트의 필드를 추가하여 적용한 것이 특징이다.

그러나 ROHC-SCTP는 기존의 ROHC가 정의하고 있는 패킷의 구조를 변경하였기 때문에 표준에 맞지 않는다. 또한 SCTP의 특징인 멀티 스트리밍을 고려하였기 때문에 통신 중에 수시로 변하는 패킷 헤더의 변화를 감지하기 위하여 전체적인 헤더압축 알고리즘의 성능 저하를 가져온다.

따라서 본 논문에서는 위의 문제점을 해결하기 위해 향상된 헤더압축 알고리즘을 설계하고 평가하였다.

본 논문에서 제안한 알고리즘을 평가하기 위해 NS-2 시뮬레이터를 이용하여 시뮬레이션 환경을 구축하였다. 헤더압축 동작을 모델링하여 환경변수로 시간 값과 패킷 타입의 변화율을 설정하고, 오버헤드율과 총 헤더 사이즈를 비교평가 하였다. 평가 결과 본 논문에서 설계한 알고리즘이 기존의 알고리즘에 비해 평균 5% 오버헤드율이 낮음을 확인하였다. 또한 데이터의 타입의 변화가 높아지면 기존 알고리즘보다 총 헤더의 크기가 더 작음을 확인하였다.

본 연구는 향후 SCTP의 헤더압축 알고리즘을 적용하기 위한 연구에 기초 연구 자료로 활용될 수 있을 것으로 사료된다.

## 참고 문헌

- [1] J. Postel, "Internet Protocol," IETF RFC 791, Sep. 1981.
- [2] F. Fitzek, S. Hendrata, P. Seeling, M. Reisslein, "Header Compression Schemes for Wireless Internet Access," in A. Salkintzis, and A. Poularikas (eds.), *Wireless Internet: Technologies and Applications*, CRC Press, pp. 7~8, 2003.
- [3] G. Pelletier, K. Sandlund, "Robust Header Compression Version 2 (ROHCv2) : Profiles for RTP, UDP, IP, ESP and UDP-Lite," IETF, RFC 5225, April 2008.
- [4] 김경신, 강문식, "차세대 IP체제에서 효율적인 데이터 전송을 위한 확장된 ROHC 알고리즘", 한국컴퓨터정보학회 논문집, 제10권, 제5호, pp.187-197, 2005.
- [5] J. Postel, "User Datagram Protocol," IETF RFC 768, Aug. 1980.
- [6] W. Boulevard, "Transmission Control Protocol," IETF RFC 793, Sep. 1981.
- [7] L-E. Jonsson, P. Kremer, "ROHC Implementer's Guide", RFC-draft <draft-ietf-rohc-rtp-implguide-08.txt>, Oct. 2004.
- [8] 신병철, 송희옥, "ROHC 압축 기법을 이용한 SCTP 헤더압축", *Journal of the Research Institute for Computer and Information Communication*, Vol.14, No.2, pp. 8~14, Oct. 2006.



- [9] IETF, RFC 3095, "RObust Header Compression (ROHC):Framework and four profiles: RTP, UDP, ESP, and uncompressed," July, 2001.
- [10] Stewart R., et al., "Stream Control Transmission Protocol," IETF RFC 2960, Oct. 2000.
- [11] IETF, RFC5225, "Robust Header Compression Version 2 (ROHCv2) : Profiles for RTP, UDP, IP, ESP and UDP-Lite," April 2008.
- [12] 김수희, 정인상, 정인환, "ROHC(RObust Header Compression) 알고리즘을 이용한 PoV(Push-to-Talk over VoIP) 설계", 한국정보과학회 가을 학술발표논문집, Vol.33, No.2, pp. 324~329, Oct. 2006.
- [13] The Network Simulator ns-2: Tips and Statistical Data for Running Large Simulations in NS; <http://www.isi.edu/nsnam/ns/ns-largesim.html>.

### 저자소개



윤성열(Sung-Yeol Yun)

2007년 한국교육개발원 학사  
2009년 경원대 전자계산학과 석사  
2009년 ~ 현재 경원대  
전자계산학과 박사과정

※관심분야: 음성통신, 네트워크 시큐리티, RFID/  
USN



박석천(Seok-Cheon Park)

1977년 고려대 전자공학과 학사  
1982년 고려대 컴퓨터공학 석사  
1989년 고려대 컴퓨터공학 박사  
1979년 ~ 1985년 금성통신연구소

1991년 ~ 1992년 UC, Irvine Post Doc.  
1988년 ~ 현재 경원대학교 컴퓨터공학과 정교수  
※관심분야: 차세대 인터넷, 멀티미디어 통신, 네트워크 시큐리티, 액티브 네트워크