Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Header compression specific Convergence Sublayer**
Date Submitted	**2004-11-04**
Source(s)	Jee-young Song, Taesoo Kwon, Hyun-ho Choi, Hyu-dae Kim, Sang-wook Kwon, Howon Lee, Dong-Ho Cho KAIST Div. of EE, Dept of EECS, KAIST, Yuseong-gu, Daejeon, Korea Voice: +82-42-869-3467 Fax: +82-42-867-0550 jysong@comis.kaist.ac.kr Yong Chang, Geunhwi Lim, Hong Sung Chang, JungWon Kim, TaeWon Kim Samsung Electronics Co. Ltd. Voice: +82-31-279-3621 yongchang@samsung.com
Re:	Re: Sponsor ballot on IEEE P802.16e/D5
Abstract	Design of Header compression specific convergence sublayer
Purpose	Adoption of proposed changes into IEEE P802.16e/D5
Notice	
Release	
Patent Policy and Procedures	

Header compression-specific Convergence Sublayer

Jee-young Song, Taesoo Kwon, Hyun-ho Choi, Hyu-dae Kim, Howon Lee, Sang-wook Kwon, Dong-Ho Cho
KAIST
Yong Chang, Geunhwi Lim, Hong Sung Chang, JungWon Kim, TaeWon Kim
Samsung Electronics

1. Introduction

While several Header Compression schemes such as ROHC, ECRTP, and so on, are widely applied for efficient utilization of resources in air interface, IP-specific Convergence Sublayer defined in current standard is not compatible to header compression schemes, and the Payload Header Suppression (PHS) scheme specified in Convergence Sublayer performs less efficiently than other header compression schemes. It is needed to define a new convergence sublayer for header compression protocols. We propose a new convergence sublayer to support header compression protocols. This document describes changes suggested for 802.16e draft to support new convergence sublayer.

2. Brief summary of Header Compression

Payload Header Suppression (PHS) included in current standard also supports IP/UDP/RTP header suppression. But header compression by RObust Header Compression (ROHC) or Enhanced Compressed RTP outperforms PHS due to considering second order difference and delta encoding.

Here's an example of ROHC that shows the difference on the size of the compressed header by each compression scheme. PHS cannot suppress the field 'Sequence number' and 'Time stamp', of which the second-order difference is zero since the first-order difference is constant. In addition, PHS cannot suppress 'Payload type' even though that field is static, because PHS operates as the unit of byte and the first bit of the second byte ('Marker' bit) is not static to suppress. Compressed_RTP of ROHC compresses RTP header to 2 bytes when the second-order differences of the fields are all zero.

VER	P	X	CC	M	PAYLOAD TYPE	SEQUENCE NUMBER
TIME STAMP						
SYNCHRONIZATION SOURCE IDENTIFIER						

12-Bytes
RTP header

(a)

VER	P	X	CC	M	PAYLOAD TYPE	SEQUENCE NUMBER
TIME STAMP						
SYNCHRONIZATION SOURCE IDENTIFIER						

7-Bytes
After PHS

(b)

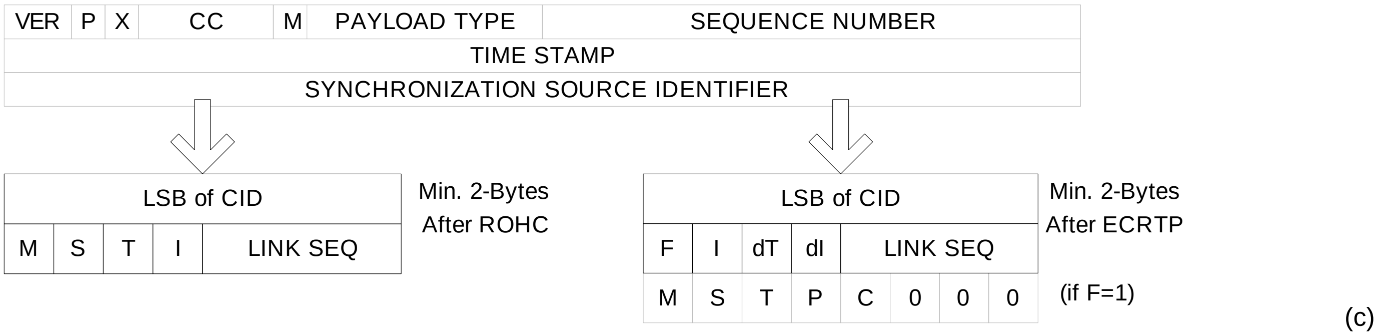

Fig. 1 RTP headers : (a) RTP full header (b) RTP header after suppression by PHS
(c) compressed header by ROHC & extended compressed header by ECRTP

PHS uses PHSM (Marker) to identify whether the marked byte shall be suppressed or transmitted. Therefore, PHS works only for the case when the first-order difference between the previous packet and the current packet is zero. ROHC compresses the fields when not only the first-order difference is zero, but the second-order difference is zero. Even though the second-order difference is not static, it compresses the fields by using of delta encoding.

In case that the first-order difference is zero, appropriate setting of PHSM enables PHS to perform as the same compression level with ROHC. However, if there exist fields that are not static, PHS that doesn’t consider the second-order difference and delta encoding suppresses less than ROHC.

Besides the performance of PHS, it is also a problem that IP-specific CS defined in current standard draft cannot support header-compressed packets. First, IP-specific CS cannot classify packets between IP and header-compression protocols. Second, although it is possible to classify packets, cannot extract the information for classifier (IP address, UDP port, DSCP, etc) from compressed header. Therefore new convergence sublayer for header compression protocol is needed.

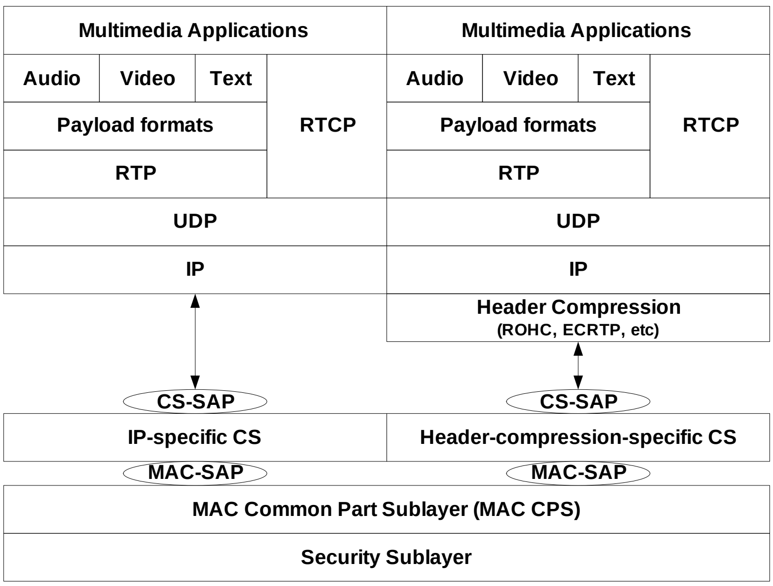

Fig. 2 Protocol Stack for IP-specific CS and Header-compression-specific CS

3. Operations for header-compression-specific packet convergence sublayer

A. Operation example for ROHC packets (compressed_UDP and/or compressed_RTP)

Header compression-specific CS extracts IP address, UDP port, IP DSCP, and ROHC Context ID from the FULL-HEADER packet at the beginning of a session. By using this information, classifier in CS maps packets from upper layer to appropriate service flow and connection ID. After getting classifier information, when ROHC packets such as compressed-UDP or compressed-RTP arrive at the CS layer, Header compression-specific CS extracts ROHC Context ID from the ROHC header to map the packet to its Connection ID of MAC layer. Header compression-specific CS updates classifier information at every arrival of FULL-HEADER packet. Header compression-CS doesn't support PHS. That means PHSI for Header compression-specific payload is always set zero.

B. Operation example for ECRTP packets (enhanced version of compressed_UDP)

Enhanced Compressed RTP (ECRTP) is based on the IP/UDP/RTP header compression defined in ROHC. ECRTP specifies the extensions to the compressed_UDP packet, in which another byte of flag is added. Basic operation of ECRTP is similar to ROHC. The difference between two header compression schemes is transparent to the header-compression convergence sublayer, so the operation of header-compression convergence sublayer for ECRTP is the same as defined in section 3.A.

4. Proposed Text Changes

In page 29, line 22, Modify the text to read:

5.2.6.2 IP classifiers
IP classifiers operate on the fields of the IP header and the transport protocols (UDP and RTP). The parameters (11.13.19.3.4.2, 11.13.19.3.4.7, 11.13.19.3.4.16, 11.13.19.3.4.17) may be used in IP classifiers.

In page 29, line 27, Add a new section as shown below:

5.2.7 Header-compression-specific part
This CS shall be applied when the compressed RTP/UDP/IP packets are carried over the IEEE Std 802.16 network.

5.2.7.1 Header-compression CS PDU format
The format of the Header-compression CS PDU shall be as shown in Figure 18. Payload Header Suppression shall not be applied for Header-compressed-packet.

PHSI=0	**Compressed header + payload**

Figure 18 Header-compression CS PDU format without header suppression

5.2.7.2 Header-compression classifiers
Header-compression classifiers operate on the fields of the header compression protocols, IP, UDP and RTP headers. The parameters (11.13.19.3.4.2, 11.13.19.3.4.7, 11.13.19.3.4.16, 11.13.19.3.4.17, 11.13.19.3.4.18, 11.13.19.3.4.19) may be used in Header-compression classifiers.

[Change the table in section 11.13.19.1]

Type	Length	Value	Scope
[145/146].28	1	0: No CS 1: Packet, IPv4 2: Packet, IPv6 3: Packet, 802.3/Ethernet 4: Packet, 802.1Q VLAN	DSA-REQ

		5: Packet, IPv4 over 802.3/Ethernet 6: Packet, IPv6 over 802.3/Ethernet 7: Packet, IPv4 over 802.3/Ethernet 8: Packet, IPv6 over 802.3/Ethernet 9: ATM 10: Packet, IPv4 with Header Compression 11: Packet, IPv6 with Header Compression 12~255: reserved	

[Change the table in section 11.13.19.2]

cst	CS
99	ATM
100	Packet, IPv4
101	Packet, IPv6
102	Packet, 802.3/Ethernet
103	Packet, 802.1Q VLAN
104	Packet, IPv4 over 802.3/Ethernet
105	Packet, IPv6 over 802.3/Ethernet
106	Packet, IPv4 over 802.3/Ethernet
107	Packet, IPv6 over 802.3/Ethernet
108	Packet, IPv4 with Header Compression
109	Packet, IPv6 with Header Compression

In page 720, line 14, Add a new section as shown below:

11.13.19.3.4.18 Session Context ID for Header-compression protocol (8-bit)

The values of the field specify the 8-bit context ID for Header-compression protocol.

Type	Length	Value
[145/146].cst.3.16	1	0~255: Session Context ID

11.13.19.3.4.19 Session Context ID for Header-compression protocol (16-bit)

The values of the field specify the 16-bit context ID for Header-compression protocol.

Type	Length	Value
[145/146].cst.3.17	2	0~65535: Session Context ID