Network Working Group
Request for Comments: 4901
Category: Standards Track

J. Ash, Ed.
J. Hand, Ed.
AT&T
A. Malis, Ed.
Verizon Communications

June 2007

Protocol Extensions for Header Compression over MPLS

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This specification defines how to use Multi-Protocol Label Switching (MPLS) to route Header-Compressed (HC) packets over an MPLS label switched path. HC can significantly reduce packet-header overhead and, in combination with MPLS, can also increases bandwidth efficiency and processing scalability in terms of the maximum number of simultaneous compressed flows that use HC at each router). Here we define how MPLS pseudowires are used to transport the HC context and control messages between the ingress and egress MPLS label switching routers. This is defined for a specific set of existing HC mechanisms that might be used, for example, to support voice over IP. This specification also describes extension mechanisms to allow support for future, as yet to be defined, HC protocols. In this specification, each HC protocol operates independently over a single pseudowire instance, very much as it would over a single point-to-point link.

Table of Contents

1. Introduction

Voice over IP (VoIP) typically uses the encapsulation voice/RTP/UDP/IP. When MPLS labels [RFC3031] are added, this becomes voice/RTP/UDP/IP/MPLS-labels. MPLS VPNs (e.g., [RFC4364]) use label stacking, and in the simplest case of IPv4 the total packet header is at least 48 bytes, while the voice payload is often no more than 30 bytes, for example. When IPv6 is used, the relative size of the header in comparison to the payload is even greater. The interest in header compression (HC) is to exploit the possibility of significantly reducing the overhead through various compression mechanisms, such as with enhanced compressed RTP (ECRTP) [RFC3545] and robust header compression (ROHC) [RFC3095, RFC3095bis, RFC4815], and also to increase scalability of HC. MPLS is used to route HC packets over an MPLS label switched path (LSP) without compression/decompression cycles at each router. Such an HC over MPLS capability can increase bandwidth efficiency as well as the processing scalability of the maximum number of simultaneous compressed flows that use HC at each router. Goals and requirements for HC over MPLS are discussed in [RFC4247]. The solution using MPLS pseudowire (PW) technology put forth in this document has been designed to address these goals and requirements.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Context: the state associated with a flow subject to IP header compression. While the exact nature of the context is specific to a particular HC protocol (CRTP, ECRTP, ROHC, etc.), this state typically includes:

- the values of all of the fields in all of the headers (IP, UDP, TCP, RTP, Encapsulating Security Payload (ESP), etc.) that the particular header compression protocol operates on for the last packet of the flow sent (by the compressor) or received (by the decompressor).

- the change in the value of some of the fields in the IP, UDP, TCP, etc. headers between the last two consecutive sent packets (compressor) or received packets (decompressor) of the flow. Some of the fields in the header change by a constant amount between subsequent packets in the flow most of the time. Saving the changes in these fields from packet to packet allows

verification that a constant rate of change is taking place, and to take appropriate action when a deviation from the normal changes are encountered.

For most HC protocols, a copy of the context of each compressed flow is maintained at both the compressor and the decompressor.

compressed Real-time Transport Protocol (CRTP): a particular HC protocol described in [RFC2508].

Context ID (CID): a small number, typically 8 or 16 bits, used to identify a particular flow, and the context associated with the flow. Most HC protocols in essence work by sending the CID across the link in place of the full header, along with any unexpected changes in the values in the various fields of the headers.

Enhanced Compressed Real-time Protocol (ECRTP): a particular HC protocol described in [RFC3545].

Forwarding Equivalence Class (FEC): a group of packets that are forwarded in the same manner (e.g., over the same LSP, with the same forwarding treatment)

Header Compression scheme (HC scheme): a particular method of performing HC and its associated protocol. Multiple methods of HC have been defined, including Robust Header Compression (ROHC [RFC3095, RFC3095bis]), compressed RTP (CRTP, [RFC2508]), enhanced CRTP (ECRTP, [RFC3545]), and IP Header Compression (IPHC, [RFC2507]). This document explicitly supports all of the HC schemes listed above, and is intended to be extensible to others that may be developed.

Header Compression channel (HC channel): a session established between a header compressor and a header decompressor using a single HC scheme, over which multiple individual flows may be compressed. From this perspective, every PPP link over which HC is operating defines a single HC channel, and based on this specification, every HC PW defines a single HC channel. HC PWs are bi-directional, which means that a unidirectional leg of the PW is set up in each direction. One leg of the bi-directional PW may be set up to carry only compression feedback, not header compressed traffic. An HC channel should not be confused with the individual traffic flows that may be compressed using a single Context ID. Each HC channel manages a set of unique CIDs.

IP Header Compression (IPHC): a particular HC protocol described in [RFC2507]

Label: a short fixed length physically contiguous identifier that is used to identify a FEC, usually of local significance

Label Stack: an ordered set of labels

Label Switched Path (LSP): the path through one or more LSRs at one level of the hierarchy followed by a packet in a particular forwarding equivalence class (FEC)

Label Switching Router (LSR): an MPLS node that is capable of forwarding native L3 packets

MPLS domain: a contiguous set of nodes that operate MPLS routing and forwarding and which are also in one Routing or Administrative Domain

MPLS label: a label that is carried in a packet header, and that represents the packet's FEC

MPLS node: a node that is running MPLS. An MPLS node will be aware of MPLS control protocols, will operate one or more L3 routing protocols, and will be capable of forwarding packets based on labels. An MPLS node may also optionally be capable of forwarding native L3 packets.

Multiprotocol Label Switching (MPLS): an IETF working group and the effort associated with the working group, including the technology (signaling, encapsulation, etc.) itself

Packet Switched Network (PSN): Within the context of Pseudowire PWE3, this is a network using IP or MPLS as the mechanism for packet forwarding.

Protocol Data Unit (PDU): the unit of data output to, or received from, the network by a protocol layer.

Pseudowire (PW): a mechanism that carries the essential elements of an emulated service from one provider edge router to one or more other provider edge routers over a PSN

Pseudowire Emulation Edge to Edge (PWE3): a mechanism that emulates the essential attributes of service (such as a T1 leased line or Frame Relay) over a PSN

Pseudowire PDU (PW-PDU): a PDU sent on the PW that contains all of the data and control information necessary to emulate the desired service

PSN Tunnel: a tunnel across a PSN, inside which one or more PWs can be carried

PSN Tunnel Signaling: a protocol used to set up, maintain, and tear down the underlying PSN tunnel

PW Demultiplexer: data-plane method of identifying a PW terminating at a provider edge router

Real Time Transport Protocol (RTP): a protocol for end-to-end network transport for applications transmitting real-time data, such as audio or video [RFC3550].

Robust Header Compression (ROHC): a particular HC protocol consisting of a framework [RFC3095bis] and a number of profiles for different protocols, e.g., for RTP, UDP, ESP [RFC3095], and IP [RFC3843]

Tunnel: a method of transparently carrying information over a network

3. Header Compression over MPLS Protocol Overview

To implement HC over MPLS, after the ingress router applies the HC algorithm to the IP packet, the compressed packet is forwarded on an MPLS LSP using MPLS labels, and then the egress router restores the uncompressed header. Any of a number of HC algorithms/protocols can be used. These algorithms have generally been designed for operation over a single point-to-point link-layer hop. MPLS PWs [RFC3985], which are used to provide emulation of many point-to-point link layer services (such as frame relay permanent virtual circuits (PVCs) and ATM PVCs) are used here to provide emulation of a single, point-to-point link layer hop over which HC traffic may be transported.

Figure 1 illustrates an HC over MPLS channel established on an LSP that traverses several LSRs, from R1/HC --> R2 --> R3 --> R4/HD, where R1/HC is the ingress router performing HC, and R4/HD is the egress router performing header decompression (HD). This example assumes that the packet flow being compressed has RTP/UDP/IP headers and is using a HC scheme such as ROHC, CRTP, or ECRTP. Compression of the RTP/UDP/IP header is performed at R1/HC, and the compressed packets are routed using MPLS labels from R1/HC to R2, to R3, and finally to R4/HD, without further decompression/recompression cycles. The RTP/UDP/IP header is decompressed at R4/HD and can be forwarded to other routers, as needed. This example assumes that the application is VoIP and that the HC algorithm operates on the RTP, UDP, and IP headers of the VoIP flows. This is an extremely common application of HC, but need not be the only one. The HC algorithms supported by the protocol extensions specified in this document may operate on TCP or IPsec ESP headers as well.

```
                    |
                    | data (e.g., voice)/RTP/UDP/IP/link layer
                    V
                  _____
                 |     |
                 |R1/HC| Header Compression (HC) Performed
                 |_____|
                    |
                    | data (e.g., voice)/compressed-header/MPLS-labels
                    V
                  _____
                 |     |
                 | R2  | Label Switching
                 |_____| (no compression/decompression)
                    |
                    | data (e.g., voice)/compressed-header/MPLS-labels
                    V
                  _____
                 |     |
                 | R3  | Label Switching
                 |_____| (no compression/decompression)
                    |
                    | data (e.g., voice)/compressed-header/MPLS-labels
                    V
                  _____
                 |     |
                 |R4/HD| Header Decompression (HD) Performed
                 |_____|
                    |
                    | data (e.g., voice)/RTP/UDP/IP/link layer
                    V
```

Figure 1: Example of HC over MPLS over Routers R1 --> R4

In the example scenario, HC therefore takes place between R1 and R4, and the MPLS LSP transports data/compressed-header/MPLS-labels instead of data/RTP/UDP/IP/MPLS-labels, often saving more than 90% of the RTP/UDP/IP overhead. Typically there are two MPLS labels (8 octets) and a link-layer HC control parameter (2 octets). The MPLS label stack and link-layer headers are not compressed. Therefore, HC over MPLS can significantly reduce the header overhead through compression mechanisms.

HC reduces the IP/UDP/RTP headers to 2-4 bytes for most packets. Half of the reduction in header size comes from the observation that half of the bytes in the IP/UDP/RTP headers remain constant over the life of the flow. After sending the uncompressed header template once, these fields may be removed from the compressed headers that

follow. The remaining compression comes from the observation that although several fields change in every packet, the difference from packet to packet is often constant or at least limited, and therefore the second-order difference is zero.

The compressor and decompressor both maintain a context for each compressed flow. The context is the session state shared between the compressor and decompressor. The details of what is included in the context may vary between HC schemes. The context at the compressor would typically include the uncompressed headers of the last packet sent on the flow, and some measure of the differences in selected header field values between the last packet transmitted and the packet(s) transmitted just before it. The context at the decompressor would include similar information about received packets. With this information, all that must be communicated across the wire is an indication of which flow a packet is associated with (the CID), and some compact encoding of the second order differences (i.e., the harder to predict differences) between packets.

MPLS PWs [RFC3985] are used to transport the HC packets between the ingress and egress MPLS LSRs. Each PW acts like a logical point-to-point link between the compressor and the decompressor. Each PW supports a single HC channel, which, from the perspective of the HC scheme operation, is similar to a single PPP link or a single frame relay PVC. One exception to this general model is that PWs carry only packets with compressed headers, and do not share the PW with uncompressed packets.

The PW architecture specifies the use of a label stack with at least 2 levels. The label at the bottom of the stack is called the PW label. The PW label acts as an identifier for a particular PW. With HC PWs, the compressor adds the label at the bottom of the stack and the decompressor removes this label. No LSRs between the compressor and decompressor inspect or modify this label. Labels higher in the stack are called the packet switch network (PSN) labels, and are used to forward the packet through the MPLS network as described in [RFC3031]. The decompressor uses the incoming MPLS PW label (the label at the bottom of the stack), along with the CID to locate the proper decompression context. Standard HC methods (e.g., ECRTP, ROHC, etc.) are used to determine the contexts. The CIDs are assigned by the HC as normal, and there would be no problem if duplicate CIDs are received at the HD for different PWs, which support different compressed channels. For example, if two different compressors, HCa and HCb, both assign the same CID to each of 2 separate flows destined to decompressor HDc, HDc can still differentiate the flows and locate the proper decompression context for each, because the tuples <PWlabel-HCa, CID> and <PWlabel-HCb, CID> are still unique.

In addition to the PW label and PSN label(s), HC over MPLS packets also carry a HC control parameter. The HC control parameter contains both a packet type field and a packet length field. The packet type field is needed because each HC scheme supported by this specification defines multiple packet types, for example, "full header" packets, which are used to initialize and/or re-synchronize the context between compressor and decompressor, vs. normal HC packets. And most of the HC schemes require that the underlying link layer protocols provide the differentiation between packet types. Similarly, one of the assumptions that is part of most of the HC schemes is that the packet length fields in the RTP/UDP/IP, etc. headers need not be explicitly sent across the network, because the IP datagram length can be implicitly determined from the lower layers. This specification assumes that, with one exception, the length of an HC IP datagram can be determined from the link layers of the packets transmitted across the MPLS network. The exception is for packets that traverse an Ethernet link. Ethernet requires padding for packets whose payload size is less than 46 bytes in length. So the HC control parameter contains a length field of 6 bits to encode the lengths of any HC packets less than 64 bytes in length.

HC PWs are set up by the PW signaling protocol [RFC4447]. [RFC4447] actually defines a set of extensions to the MPLS label distribution protocol (LDP) [RFC3036]. As defined in [RFC4447], LDP signaling to set up, tear down, and manage PWs is performed directly between the PW endpoints, in this case, the compressor and the decompressor. PW signaling is used only to set up the PW label at the bottom of the stack, and is used independently of any other signaling that may be used to set up PSN labels. So, for example, in Figure 1, LDP PW signaling would be performed directly between R1/HC and R4/HD. Router R2 and R3 would not participate in PW signaling.

[RFC4447] provides extensions to LDP for PWs, and this document provides further extensions specific to HC. Since PWs provide a logical point-to-point connection over which HC can be run, the extensions specified in this document reuse elements of the protocols used to negotiate HC over the Point-to-Point Protocol [RFC1661]. [RFC3241] specifies how ROHC is used over PPP and [RFC3544] specifies how several other HC schemes (CRTP, ECRTP, IPHC) are used over PPP. Both of these RFCs provide configuration options for negotiating HC over PPP. The formats of these configuration options are reused here for setting up HC over PWs. When used in the PPP environment, these configuration options are used as extensions to PPP's IP Control Protocol [RFC1332] and the detailed PPP options negotiations process described in [RFC1661]. This is necessary because a PPP link may support multiple protocols, each with its own addressing scheme and options. Achieving interoperability requires a negotiation process

so that the nodes at each end of the link can agree on a set of protocols and options that both support. However, a single HC PW supports only HC traffic using a single HC scheme. So while the formats of configuration options from [RFC3241] and [RFC3544] are reused here, the detailed PPP negotiation process is not. Instead, these options are reused here just as descriptors (TLVs in the specific terminology of LDP and [RFC4447]) of basic parameters of an HC PW. These parameters are further described in Section 4. The HC configuration parameters are initially generated by the decompressor and describe what the decompressor is prepared to receive.

Most HC schemes use a feedback mechanism which requires bi-directional flow of HC packets, even if the flow of compressed IP packets is in one direction only. The basic signaling process of [RFC4447] sets up unidirectional PWs, and must be repeated in each direction in order to set up the bi-directional flow needed for HC.

Figure 1 illustrates an example data flow set up from R1/HC --> R2 --> R3 --> R4/HD, where R1/HC is the ingress router where header compression is performed, and R4/HD is the egress router where header decompression is done. Each router functions as an LSR and supports signaling of LSP/PWs. See Section 5 for a detailed example of how the flow depicted in Figure 1 is established.

All the HC schemes used here are built so that if an uncompressible packet is seen, it should just be sent uncompressed. For some types of compression (e.g., IPHC-TCP), a non-compressed path is required. For IPHC-TCP compression, uncompressible packets occur for every TCP flow. Another way that this kind of issue can occur is if MAX_HEADER is configured lower than the longest header, in which case, compression might not be possible in some cases.

The uncompressed packets associated with HC flows (e.g., uncompressed IPHC-TCP packets) can be sent through the same MPLS tunnel along with all other non-HC (non-PW) IP packets. MPLS tunnels can transport many types of packets simultaneously, including non-PW IP packets, layer 3 VPN packets, and PW (e.g., HC flow) packets. In the specification, we assume that there is a path for uncompressed traffic, and it is a compressor decision as to what would or would not go in the HC-PW.

4. Protocol Specifications

Figure 2 illustrates the PW stack reference model to support PW emulated services.

```
+-------------+                                +-------------+
|  Layer2     |                                |  Layer2     |
|  Emulated   |                                |  Emulated   |
|  Services   |         Emulated Service       |  Services   |
|             |<==============================>|             |
+-------------+                                +-------------+
|     HC      |           Pseudowire           |     HD      |
|Demultiplexer|<==============================>|Demultiplexer|
+-------------+                                +-------------+
|    PSN      |            PSN Tunnel          |    PSN      |
|   MPLS      |<==============================>|   MPLS      |
+-------------+                                +-------------+
|  Physical   |                                |  Physical   |
+-----+-------+                                +-----+-------+
```

Figure 2: Pseudowire Protocol Stack Reference Model

Each HC-HD compressed channel is mapped to a single PW and associated with 2 PW labels, one in each direction. A single PW label MUST be used for many HC flows (could be 100's or 1000's) rather than assigning a different PW label to each flow. The latter approach would involve a complex mechanism for PW label assignment, freeing up of labels after a flow terminates, etc., for potentially 1000's of simultaneous HC flows. On the other hand, the mechanism for CID assignment, freeing up, etc., is in place and there is no need to duplicate it with PW assignment/deassignment for individual HC flows.

Multiple PWs SHOULD be established in case different quality of service (QoS) requirements are needed for different compressed streams. The QoS received by the flow would be determined by the EXP bit marking in the PW label. Normally, all RTP packets would get the same EXP marking [RFC3270], equivalent to expedited forwarding (EF) treatment [RFC3246] in Diffserv. However, the protocol specified in this document applies to several different types of streams, not just RTP streams, and QoS treatment other than EF may be required for those streams.

Figure 3 shows the HC over MPLS protocol stack (with uncompressed header):

```
   Media stream
   RTP
   UDP
   IP
   HC control parameter
   MPLS label stack (at least 2 labels for this application)
   Link layer under MPLS (PPP, PoS, Ethernet)
   Physical layer (SONET/SDH, fiber, copper)


                                                        +--------------+
                                                        | Media stream |
                                                        +--------------+
                                                        \_______ ______/
                                                2-4 octets      V
                                                 +------+--------------+
                         Compressed /RTP/UDP/IP/ |header|              |
                                                 +------+--------------+
                                                 \__________ __________/
                                          2 octets          V
                                          +------+---------------------+
                     HC Control Parameter |header|                     |
                                          +------+---------------------+
                                          \______________ _____________/
                                   8 octets              V
                                   +------+----------------------------+
                       MPLS Labels |header|                            |
                                   +------+----------------------------+
                                   \_________________ _________________/
                                                     V
                            +------------------------------------------+
      Link Layer under MPLS |                                          |
                            +------------------------------------------+
                            \____________________ _____________________/
                                                 V
                     +-------------------------------------------------+
      Physical Layer |                                                 |
                     +-------------------------------------------------+

     Figure 3: Header Compression over MPLS Media Stream Transport
```

The HC control parameter MUST be used to identify the packet types for the HC scheme in use. The MPLS labels technically define two layers: the PW identifier and the MPLS tunnel identifier. The PW label MUST be used as the demultiplexer field by the HD, where the PW label appears at the bottom label of an MPLS label stack. The LSR that will be performing decompression MUST ensure that the label it distributes (e.g., via LDP) for a channel is unique. There can also

be other MPLS labels, for example, to identify an MPLS VPN. The IP/UDP/RTP headers are compressed before transmission, leaving the rest of the stack alone, as shown in Figure 3.

4.1. MPLS Pseudowire Setup and Signaling

PWs MUST be set up in advance for the transport of media streams using [RFC4447] control messages exchanged by the HC-HD endpoints. Furthermore, a PW type MUST be used to indicate the HC scheme being used on the PW. [RFC4447] specifies the MPLS label distribution protocol (LDP) [RFC3036] extensions to set up and maintain the PWs, and defines new LDP objects to identify and signal attributes of PWs. Any acceptable method of MPLS label distribution MAY be used for distributing the MPLS tunnel label [RFC3031]. These methods include LDP [RFC3036], RSVP-TE [RFC3209], or configuration.

To assign and distribute the PW labels, an LDP session MUST be set up between the PW endpoints using the extended discovery mechanism described in [RFC3036]. The PW label bindings are distributed using the LDP downstream unsolicited mode described in [RFC3036]. An LDP label mapping message contains a FEC object, a label object, and possible other optional objects. The FEC object indicates the meaning of the label, identifies the PW type, and identifies the PW that the PW label is bound to. See [RFC4447] for further explanation of PW signaling.

This specification defines new PW type values to be carried within the FEC object to identify HC PWs for each HC scheme. The PW type is a 15-bit parameter assigned by IANA, as specified in the [RFC4446] registry, and MUST be used to indicate the HC scheme being used on the PW. IANA has set aside the following PW type values for assignment according to the registry specified in RFC 4446, Section 3.2:

PW type	Description	Reference
0x001A	ROHC Transport Header-compressed Packets	[RFC3095bis]
0x001B	ECRTP Transport Header-compressed Packets	[RFC3545]
0x001C	IPHC Transport Header-compressed Packets	[RFC2507]
0x001D	CRTP Transport Header-compressed Packets	[RFC2508]

The HC control parameter enables distinguishing between various packets types (e.g., uncompressed, UDP compressed, RTP compressed, context-state, etc.). However, the HC control parameter indications are not unique across HC schemes, and therefore the PW type value allows the HC scheme to be identified.

4.2. Header Compression Scheme Setup, Negotiation, and Signaling

As described in the previous section, the HC PW MUST be used for compressed packets only, which is configured at PW setup. If a flow is not compressed, it MUST NOT be placed on the HC PW. HC PWs MUST be bi-directional, which means that a unidirectional leg of the PW MUST be set up in each direction. One leg of the bi-directional PW MAY be set up to carry only compression feedback, not header compressed traffic. The same PW type MUST be used for PW signaling in both directions.

HC scheme parameters MAY be manually configured, but if so, manual configuration MUST be done in both directions. If HC scheme parameters are signaled, the Interface Parameters Sub-TLV MUST be used on any unidirectional legs of a PW that will carry HC traffic. For a unidirectional leg of a PW that will carry only compression feedback, the components of the Interface Parameters Sub-TLV described below are not relevant and MUST NOT be used.

The PW HC approach relies on the PW/MPLS layer to convey HC channel configuration information. The Interface Parameters Sub-TLV [IANA, RFC4447] must be used to signal HC channel setup and specify HC parameters. That is, the configuration options specified in [RFC3241, RFC3544] are reused in this specification to specify PW-specific parameters, and to configure the HC and HD ports at the edges of the PW so that they have the necessary capabilities to interoperate with each other.

Pseudowire Interface Parameter Sub-TLV type values are specified in [RFC4446]. IANA has set aside the following Pseudowire Interface Parameter Sub-TLV type values according to the registry specified in RFC 4446, Section 3.3:

Parameter	ID Length	Description	Reference
0x0D	up to 256 bytes	ROHC over MPLS configuration RFC 3241	RFC 4901
0x0F	up to 256 bytes	CRTP/ECRTP/IPHC HC over MPLS configuration RFC 3544	RFC 4901

TLVs identified in [RFC3241] and [RFC3544] MUST be encapsulated in the PW Interface Parameters Sub-TLV and used to negotiate header compression session setup and parameter negotiation for their respective protocols. The TLVs supported in this manner MUST include the following:

- o Configuration Option Format, RTP-Compression Suboption, Enhanced RTP-Compression Suboption, TCP/non-TCP Compression Suboptions, as specified in [RFC3544]

- o Configuration Option Format, PROFILES Suboption, as specified in [RFC3241]

These TLVs are now specified in the following sections.

4.2.1. Configuration Option Format [RFC3544]

Both the network control protocol for IPv4, IPCP [RFC1332] and the IPv6 Network Control Protocol (NCP), IPV6CP [RFC2472] may be used to negotiate IP HC parameters for their respective controlled protocols. The format of the configuration option is the same for both IPCP and IPV6CP. This configuration option MUST be included for ECRTP, CRTP and IPHC PW types and MUST NOT be included for ROHC PW types. A decompressor MUST reject this option (if misconfigured) for ROHC PW types and send an explicit error message to the compressor [RFC3544].

Description

This NCP configuration option is used to negotiate parameters for IP HC. Successful negotiation of parameters enables the use of Protocol Identifiers FULL_HEADER, COMPRESSED_TCP, COMPRESSED_TCP_NODELTA, COMPRESSED_NON_TCP, and CONTEXT_STATE as specified in [RFC2507]. The option format is summarized below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |    IP-Compression-Protocol    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            TCP_SPACE          |          NON_TCP_SPACE        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         F_MAX_PERIOD          |            F_MAX_TIME         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           MAX_HEADER          |            suboptions...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
2

Length
>= 14

The length may be increased if the presence of additional parameters is indicated by additional suboptions.

IP-Compression-Protocol
0061 (hex)

TCP_SPACE
The TCP_SPACE field is two octets and indicates the maximum value of a context identifier in the space of context identifiers allocated for TCP.

Suggested value: 15

TCP_SPACE must be at least 0 and at most 255 (the value 0 implies having one context). This field is not used for CRTP (PW type 0x001B) and ECRTP (PW type 0x001B) PWs. For these PW types, it should be set to its suggested value by the sender and ignored by the receiver.

NON_TCP_SPACE
The NON_TCP_SPACE field is two octets and indicates the maximum value of a context identifier in the space of context identifiers allocated for non-TCP. These context identifiers are carried in COMPRESSED_NON_TCP, COMPRESSED_UDP and COMPRESSED_RTP packet headers.

Suggested value: 15

NON_TCP_SPACE must be at least 0 and at most 65535 (the value 0 implies having one context).

F_MAX_PERIOD
Maximum interval between full headers. No more than F_MAX_PERIOD COMPRESSED_NON_TCP headers may be sent between FULL_HEADER headers.

Suggested value: 256

A value of zero implies infinity, i.e., there is no limit to the number of consecutive COMPRESSED_NON_TCP headers. This field is not used for CRTP (PW type 0x001B) and ECRTP (PW type 0x001B) PWs. For these PW types, it should be set to its suggested value by the sender and ignored by the receiver.

F_MAX_TIME

Maximum time interval between full headers. COMPRESSED_NON_TCP headers may not be sent more than F_MAX_TIME seconds after sending the last FULL_HEADER header.

Suggested value: 5 seconds

A value of zero implies infinity. This field is not used for CRTP (PW type 0x001B) and ECRTP (PW type 0x001B) PWs. For these PW types, it should be set to its suggested value by the sender and ignored by the receiver.

MAX_HEADER

The largest header size in octets that may be compressed.

Suggested value: 168 octets

The value of MAX_HEADER should be large enough so that at least the outer network layer header can be compressed. To increase compression efficiency MAX_HEADER should be set to a value large enough to cover common combinations of network and transport layer headers.

suboptions

The suboptions field consists of zero or more suboptions. Each suboption consists of a type field, a length field and zero or more parameter octets, as defined by the suboption type. The value of the length field indicates the length of the suboption in its entirety, including the lengths of the type and length fields.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Parameters...|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2.2. RTP-Compression Suboption [RFC3544]

The RTP-Compression suboption is included in the NCP IP-Compression-Protocol option for IPHC if IP/UDP/RTP compression is to be enabled. This suboption MUST be included for CRTP PWs (0x001C) and MUST NOT be included for other PW types.

Inclusion of the RTP-Compression suboption enables use of additional Protocol Identifiers COMPRESSED_RTP and COMPRESSED_UDP along with additional forms of CONTEXT_STATE as specified in [RFC2508].

Description

Enables the use of Protocol Identifiers COMPRESSED_RTP, COMPRESSED_UDP, and CONTEXT_STATE as specified in [RFC2508].

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
1

Length
2

4.2.3. Enhanced RTP-Compression Suboption [RFC3544]

To use the enhanced RTP HC defined in [RFC3545], a new suboption 2 is added. Suboption 2 is negotiated instead of, not in addition to, suboption 1. This suboption MUST be included for ECRTP PWs (0x001B) and MUST NOT be included for other PW types.

Note that suboption 1 refers to the RTP-Compression Suboption, as specified in Section 4.2.2, and suboption 2 refers to the Enhanced RTP-Compression Suboption, as specified in Section 4.2.3. These suboptions MUST NOT occur together. If they do (e.g., if misconfigured), a decompressor MUST reject this option and send an explicit error message to the compressor [RFC3544].

Description

Enables the use of Protocol Identifiers COMPRESSED_RTP and CONTEXT_STATE as specified in [RFC2508]. In addition, it enables the use of [RFC3545] compliant compression including the use of Protocol Identifier COMPRESSED_UDP with additional flags and use of the C flag with the FULL_HEADER Protocol Identifier to indicate use of HDRCKSUM with COMPRESSED_RTP and COMPRESSED_UDP packets.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
2

Length
2

4.2.4. Negotiating Header Compression for Only TCP or Only Non-TCP Packets [RFC3544]

In [RFC3544] it was not possible to negotiate only TCP HC or only non-TCP HC because a value of 0 in the TCP_SPACE or the NON_TCP_SPACE fields actually means that 1 context is negotiated.

A new suboption 3 is added to allow specifying that the number of contexts for TCP_SPACE or NON_TCP_SPACE is zero, disabling use of the corresponding compression. This suboption MUST be included for IPHC PWs (0x001C) and MUST NOT be included for other PW types.

Description

Enable HC for only TCP or only non-TCP packets.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |   Parameter   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
3

Length
3

Parameter

The parameter is 1 byte with one of the following values:

1 = the number of contexts for TCP_SPACE is 0
2 = the number of contexts for NON_TCP_SPACE is 0

This suboption overrides the values that were previously assigned to TCP_SPACE and NON_TCP_SPACE in the IP HC option.

If suboption 3 is included multiple times with parameter 1 and 2, compression is disabled for all packets.

4.2.5. Configuration Option Format [RFC3241]

Both the network control protocol for IPv4, IPCP [RFC1332] and the IPv6 NCP, IPV6CP [RFC2472] may be used to negotiate IP HC parameters for their respective controlled protocols. The format of the configuration option is the same for both IPCP and IPV6CP. This configuration option MUST be included for ROHC PW types and MUST NOT be included for ECRTP, CRTP, and IPHC PW types. A decompressor MUST reject this option (if misconfigured) for ECRTP, CRTP, and IPHC PW types, and send an explicit error message to the compressor [RFC3544].

Description

This NCP configuration option is used to negotiate parameters for ROHC. The option format is summarized below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |    IP-Compression-Protocol    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            MAX_CID            |             MRRU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           MAX_HEADER          |          suboptions...        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
   2

Length
   >= 10

   The length may be increased if the presence of additional parameters is indicated by additional suboptions.

IP-Compression-Protocol
   0003 (hex)

MAX_CID
   The MAX_CID field is two octets and indicates the maximum value of a context identifier.

   Suggested value: 15

   MAX_CID must be at least 0 and at most 16383 (The value 0 implies having one context).

MRRU
The MRRU field is two octets and indicates the maximum reconstructed reception unit (see [RFC3095bis], Section 5.1.2).

Suggested value: 0

MAX_HEADER
The largest header size in octets that may be compressed.

Suggested value: 168 octets

The value of MAX_HEADER should be large enough so that at least the outer network layer header can be compressed. To increase compression efficiency MAX_HEADER should be set to a value large enough to cover common combinations of network and transport layer headers.

NOTE: The four ROHC profiles defined in RFC 3095 do not provide for a MAX_HEADER parameter. The parameter MAX_HEADER defined by this document is therefore without consequence in these profiles because the maximum compressible header size is unspecified. Other profiles (e.g., ones based on RFC 2507) can make use of the parameter by explicitly referencing it.

suboptions
The suboptions field consists of zero or more suboptions. Each suboption consists of a type field, a length field, and zero or more parameter octets, as defined by the suboption type. The value of the length field indicates the length of the suboption in its entirety, including the lengths of the type and length fields.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Parameters...|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2.6. PROFILES Suboption [RFC3241]

The set of profiles to be enabled is subject to negotiation. Most initial implementations of ROHC implement profiles 0x0000 to 0x0003. This option MUST be supplied.

Description

Define the set of profiles supported by the decompressor.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Profiles...  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type
   1

Length
   2n+2

Value
   n octet-pairs in ascending order, each octet-pair specifying a ROHC profile supported.

HC flow identification is being done now in many ways. Since there are multiple possible approaches to the problem, no specific method is specified in this document.

4.3. Encapsulation of Header Compressed Packets

The HC control parameter is used to identify the packet types for IPHC [RFC2507], CRTP [RFC2508], and ECRTP [RFC3545], as shown in Figure 4:

```
                     1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0|Pkt Typ|  Length   |Res|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: HC Control Parameter

where:

"Packet Type" encoding:
0: ROHC Small-CIDs
1: ROHC Large-CIDs
2: FULL_HEADER
3: COMPRESSED_TCP
4: COMPRESSED_TCP_NODELTA
5: COMPRESSED_NON_TCP
6: COMPRESSED_RTP_8
7: COMPRESSED_RTP_16
8: COMPRESSED_UDP_8
9: COMPRESSED_UDP_16
10: CONTEXT_STATE

11-15: Not yet assigned. (See Section 8, "IANA Considerations", for discussion of the registration rules.)

As discussed in [ECMP-AVOID], since this MPLS payload type is not IP, the first nibble is set to 0000 to avoid being mistaken for IP. This is also consistent with the encoding of the PW MPLS control word (PWMCW) described in [RFC4385]; however, the HC control parameter is not intended to be a PWMCW.

Note that ROHC [RFC3095, RFC3095bis] provides its own packet type within the protocol; however, the HC control parameter MUST still be used to avoid the problems identified above. Since the "Packet Type" will be there anyway, it is used to indicate ROHC CID size, in the same way as with PPP.

The HC control parameter length field is ONLY used for short packets because padding may be appended by the Ethernet Data Link Layer. If the length is greater than or equal to 64 octets, the length field MUST be set to zero. If the MPLS payload is less than 64 bytes, then the length field MUST be set to the length of the PW payload plus the length of the HC control parameter. Note that the last 2 bits in the HC control parameter are reserved.

4.4. Packet Reordering

Packet reordering for ROHC is discussed in [RFC4224], which is a useful source of information. In case of lossy links and other reasons for reordering, implementation adaptations are needed to allow all the schemes to be used in this case. Although CRTP is viewed as having risks for a number of PW environments due to reordering and loss, it is still the protocol of choice in many cases. CRTP was designed for reliable point to point links with short delays. It does not perform well over links with a high rate of packet loss, packet reordering, and long delays. In such cases, ECRTP [RFC3545] may be considered to increase robustness to both packet loss and misordering between the compressor and the decompressor. This is achieved by repeating updates and sending of absolute (uncompressed) values in addition to delta values for selected context parameters. IPHC should use TCP_NODELTA, ECRTP should send absolute values, ROHC should be adapted as discussed in [RFC4224]. An evaluation and simulation of ECRTP and ROHC reordering is given in [REORDER-EVAL].

5. HC Pseudowire Setup Example

This example will trace the setup of an MPLS PW supporting bi-directional ECRTP [RFC3545] traffic. The example assumes the topology shown in Figure 1. The PW will be set up between LSRs R1/HC and R4/HD. LSRs R2 and R3 have no direct involvement in the signaling for this PW, other than to transport the signaling traffic.

For this example, it is assumed that R1/HC has already obtained the IP address of R4/HD used for LDP signaling, and vice versa, that both R1/HC and R4/HD have been configured with the same 32-bit PW ID, as described in Section 5.2 of [RFC4447], and that R1/HC has been configured to initiate the LDP discovery process. Furthermore, we assume that R1/HC has been configured to receive a maximum of 200 simultaneous ECRTP flows from R4/HD, and R4/HD has been configured to receive a maximum of 255 ECRTP flows from R1/HC.

Assuming that there is no existing LDP session between R1/HC and R4/HD, the PW signaling must start by setting up an LDP session between them. As described earlier in this document, LDP extended discovery is used between HC over MPLS LSRs. Since R1/HC has been configured to initiate extended discovery, it will send LDP Targeted Hello messages to R4/HD's IP address at UDP port 646. The Targeted Hello messages sent by R1/HC will have the "R" bit set in the Common Hello Parameters TLV, requesting R4/HD to send Targeted Hello messages back to R1/HC. Since R4/HD has been configured to set up an HC PW with R1/HD, R4/HD will do as requested and send LDP Targeted Hello messages as unicast UDP packets to UDP port 646 of R1/HC's IP address.

When R1/HC receives a Targeted Hello message from R4/HD, it may begin establishing an LDP session to R4/HD. It starts this by initiating a TCP connection on port 646 to R4/HD's signaling IP address. After successful TCP connection establishment, R1/HC sends an LDP Initialization message to R4/HD with the following characteristics:

When R1/HC receives a Targeted Hello message from R4/HD, it may begin establishing an LDP session to R4/HD. The procedure described in Section 2.5.2 of [RFC3036] is used to determine which LSR is the active LSR and which is the passive LSR. Assume that R1/HC has the numerically higher IP address and therefore takes the active role. R1/HC starts by initiating a TCP connection on port 646 to R4/HD's signaling IP address. After successful TCP connection establishment, R1/HC sends an LDP Initialization message to R4/HD with the following characteristics:

o Common Session Parameters TLV:
  - A bit = 0 (Downstream Unsolicited Mode)
  - D bit = 0 (Loop Detection Disabled)
  - PVLim = 0 (required when D bit = 0)
  - Receive LDP identifier (taken from R4/HD's Hello message)
    > 4 octets LSR identifier (typically an IP address with IPv4)
    > 2 octet Label space identifier (typically 0)
o No Optional Parameters TLV

Following the LDP session initialization state machine of Section 2.5.4 of [RFC3036], R4/HD would send a similar Initialization message to R1/HD. The primary difference would be that R4/HD would use the LDP identifier it received in R1/HC's Hello message(s) as the Receive LDP identifier. Assuming that all other fields in the Common Session Parameters TLV were acceptable to both sides, R1/HC would send an LDP Keepalive message to R4/HD, R4/HD would send a LDP Keepalive message to R1/HC, and the LDP session would become operational.

At this point, either R1/HC or R4/HD may send LDP Label Mapping messages to configure the PW. The Label Mapping message sent by a particular router advertises the label that should be used at the bottom of the MPLS label stack for all packets sent to that router and associated with the particular PW. The Label Mapping message sent from R1/HC to R4/HD would have the following characteristics:

o FEC TLV
  - FEC Element type 0x80 (PWid FEC Element, as defined in [RFC4447]
  - Control Parameter bit = 1 (Control Parameter present)
  - PW type = 0x001B (ECRTP [RFC3545])
  - Group ID as chosen by R1/HC
  - PW ID = the configured value for this PW, which must be the same as that sent in the Label Mapping message by R4/HD
  - Interface Parameter Sub-TLVs
    > Interface MTU sub-TLV (Type 0x01)
    > CRTP/ECRTP/IPHC HC over MPLS configuration sub-TLV (Type 0x0F)
      + Type = 2 (From RFC 3544)
      + Length = 16
      + TCP_SPACE = Don't Care (leave at suggested value = 15)
      + NON_TCP_SPACE = 200 (configured on R1)
      + F_MAX_PERIOD = Don't Care (leave at suggested value = 256)
      + F_MAX_TIME = Don't Care (leave at suggested value = 5 seconds)
      + MAX_HEADER = 168 (Suggested Value)
      + Enhanced RTP-Compression Suboption
        & Type = 2
        & Length = 2
o Label TLV - contains label selected by R1, Lr1
o No Optional Parameters

The Label Mapping message sent from R4/HD to R1/HC would be almost identical to the one sent in the opposite direction, with the following exceptions:

o R4/HD could select a different Group ID
o The Value of NON_TCP_SPACE in the CRTP/ECRTP/IPHC HC over MPLS configuration sub-TLV would be 255 instead of 200, as configured on R4/HD
o R4/HD would choose its own value for the Label TLV, Lr4

As soon as either R1/HC or R4/HD has both transmitted and received Label Mapping Messages with the same PW Type and PW ID, that HC endpoint considers the PW established. R1/HC could send ECRTP packets using the label it received in the Label Mapping Message from R4/HD, Lr4, and could identify received ECRTP packets by the label it had sent to R4/HD, Lr1. And vice versa.

In this case, assume that R1/HC has an IPv4 RTP flow to send to R4/HD that it wishes to compress using the ECRTP PW just set up. The RTP flow is G.729 media with 20 bytes of payload in each RTP packet. In this particular case, the IPv4 identifier changes by a small constant value between consecutive packets in the stream. In the RTP layer of the flow, the Contributing Source Identifiers count is 0. R1/HC decides to use 8-bit Context Identifiers for the compressed flow. Also, R1/HC determines that compression in this particular flow should be able to recover from the loss of 2 consecutive packets without requiring re-synchronization of the context (i.e., the "N" value from [RFC3545] is 2).

The first 3 (N + 1) packets of this flow would be sent as FULL_HEADER packets. The MPLS and PW headers at the beginning of these packets would be formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                  XX                   |  XX |0|        XX     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                 Lr4                   |  XX |1|        >0     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       |Pkt Typ|  Length   |Res|
|0 0 0 0|   2   |     62    |0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            ^
            |
             -- 2 == FULL_HEADER
```

where XX signifies either
a. value determined by the MPLS routing layer
b. don't care

Immediately following the above header would come the FULL_HEADER packet as defined in [RFC3545], which basically consists of the IP/UDP/RTP header, with the IP and UDP length field replaced by values encoding the CID, sequence number, and "generation", as defined in [RFC3545]. The length field value of 62 comprises:

- o 2 bytes of HC control parameter (included in the above diagram)
- o 20 bytes of the IP header portion of the RFC 3545 FULL_HEADER
- o 8 bytes of the UDP header portion of the RFC 3545 FULL_HEADER
- o 12 bytes of the RTP header portion of the RFC 3545 FULL_HEADER
- o 20 bytes of G.729 payload

The next 3 RTP packets from this flow would be sent as COMPRESSED_UDP_8, to establish the absolute and delta values of the IPv4 identifier and RTP timestamp fields. These packets would use the same ECRTP CID as the previous 3 FULL_HEADER packets. The MPLS and PW headers at the beginning of these packets would be formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                  XX                   |  XX |0|        XX     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                 Lr4                   |  XX |1|        >0     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       |Pkt Typ|  Length   |Res|
|0 0 0 0|   8   |     36    |0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            ^
            |
             -- 8 == COMPRESSED_UDP_8
```

There is no change in the MPLS label stack between the FULL_HEADER packets and the COMPRESSED_UDP packets. The HC control parameter changes to reflect another ECRTP packet type following the control parameter, and a change of packet length. The length changes because the new packet type more compactly encodes the headers. The length field value of 36 comprises:

o 2 bytes of HC control parameter (included in the above diagram)
o 1 byte of CID
o 2 bytes of COMPRESSED_UDP fields that are not octet-aligned:
  - 4 bits of COMPRESSED_UDP flags
  - 4 bits of sequence number
  - 5 bits of COMPRESSED UDP extension flags
  - 3 bits MUST_BE_ZERO
o 2 bytes of UDP checksum or HDRCKSUM
o 1 byte of delta IPv4 ID
o 2 bytes of delta RTP timestamp (changes by 160 in this case, differential encoding will encode as 2 bytes)
o 2 bytes of absolute IPv4 ID
o 4 bytes of absolute RTP timestamp
o 20 bytes of G.729 payload

After the context for the IPv4 ID and RTP timestamp is initialized. Subsequent packets on this flow, at least until the end of the talk spurt or until there is some other unexpected change in the IP/UDP/RTP headers, may be sent as COMPRESSED_RTP_8 packets. Again, the same MPLS stack would be used for these packets, and the same value of the CID would be used in this case as for the packets described above. The MPLS and PW headers at the beginning of these packets would be formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                  XX                   |  XX |0|        XX     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | Exp |S|       TTL     |
|                 Lr4                   |  XX |1|        >0     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       |Pkt Typ|  Length   |Res|
|0 0 0 0|   6   |     26    |0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            ^
            |
             -- 6 == COMPRESSED_RTP_8
```

The HC control parameter again changes to reflect another ECRTP packet type following the control parameter, and shorter length associated with an even more compact encoding of headers. The length field value of 26 comprises:

- o 2 bytes of HC control parameter (included in the above diagram)
- o 1 byte of CID
- o 1 byte COMPRESSED_UDP fields that are not octet-aligned:
  - 4 bits of COMPRESSED_RTP flags
  - 4 bits of sequence number
- o 2 bytes of UDP checksum or HDRCKSUM
- o 20 bytes of G.729 payload

Additional flows in the same direction may be compressed using the same basic encapsulation, including the same PW label. The CID that is part of the HC protocol is used to differentiate flows. For traffic in the opposite direction, the primary change would be the PW label, Lr4, used in the example above would be replaced by the label Lr1 that R1/HC provides to R4/HD.

6. Security Considerations

MPLS PW security considerations in general are discussed in [RFC3985] and [RFC4447], and those considerations also apply to this document. This document specifies an encapsulation and not the protocols that may be used to carry the encapsulated packets across the PSN, or the protocols being encapsulated. Each such protocol may have its own set of security issues, but those issues are not affected by the encapsulations specified herein.

The security considerations of the supported HC protocols [RFC2507, RFC2508, RFC3095, RFC3095bis, RFC3545] all apply to this document as well.

7. Acknowledgements

The authors appreciate valuable inputs and suggestions from Loa Andersson, Scott Brim, Stewart Bryant, Spencer Dawkins, Adrian Farrel, Victoria Fineberg, Eric Gray, Allison Mankin, Luca Martini, Colin Perkins, Kristofer Sandlund, Yaakov Stein, George Swallow, Mark Townsley, Curtis Villamizar, and Magnus Westerlund.

8. IANA Considerations

As discussed in Section 4.1, PW type values have been assigned by IANA, as follows:

```
0x001A  ROHC Transport Header-compressed Packets    [RFC3095bis]
0x001B  ECRTP Transport Header-compressed Packets   [RFC3545]
0x001C  IPHC Transport Header-compressed Packets    [RFC2507]
0x001D  CRTP Transport Header-compressed Packets    [RFC2508]
```

Procedures for registering new PW type values are given in [RFC4446].

As discussed in Section 4.2, Pseudowire Interface Parameter Sub-TLV type values have been specified by IANA, as follows:

Parameter ID	Length	Description	Reference
0x0D	up to 256 bytes	ROHC over MPLS configuration RFC 3241	RFC 4901
0x0F	up to 256 bytes	CRTP/ECRTP/IPHC HC over MPLS configuration RFC 3544	RFC 4901

As discussed in Section 4.3, IANA has defined a new registry, "Header Compression Over MPLS HC Control Parameter Packet Type". This is a four-bit value. Packet Types 0 through 10 are defined in Section 4.3 of this document. Packet Types 11 to 15 are to be assigned by IANA using the "Expert Review" policy defined in [RFC2434].

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[RFC3241] Bormann, C., "Robust Header Compression (ROHC) over PPP", RFC 3241, April 2002.

[RFC3544] Engan, M., Casner, S., Bormann, C., and T. Koren, "IP Header Compression over PPP", RFC 3544, July 2003.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

10. Informative References

[ECMP-AVOID] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", Work in Progress, February 2007.

[REORDER-EVAL] Knutsson, C., "Evaluation and Implementation of Header Compression Algorithm ECRTP", http://epubl.luth.se/1402-1617/2004/286/LTU-EX-04286-SE.pdf.

[RFC1332] McGregor, G., "The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2472] Haskin, D. and E. Allen, "IP Version 6 over PPP", RFC 2472, December 1998.

[RFC2507] Degermark, M., Nordgren, B., and S. Pink, "IP Header Compression", RFC 2507, February 1999.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC3095] Bormann, C., et al., "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[RFC3095bis] Jonsson, L-E. Pelletier, G., and K. Sandlund, "The RObust Header Compression (ROHC) Framework", Work in Progress, November 2006.

[RFC3209] Awduche, D., et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, December 2001.

[RFC3544] Koren, T., et al., "IP Header Compression over PPP," RFC 3544, July 2003.

[RFC3545] Koren, T., et al., "Compressing IP/UDP/RTP Headers on Links with High Delay, Packet Loss, and Reordering," RFC 3545, July 2003.

[RFC3246] Davie, B., et al., "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246, March 2002.

[RFC3270] Le Faucheur, F., et al., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," RFC 3270, May 2002.

[RFC3550] Schulzrinne, H., et al., "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.

[RFC3843] Jonsson, L-E. and G. Pelletier, "RObust Header Compression (ROHC): A Compression Profile for IP", RFC 3843, June 2004.

[RFC3985] Bryant, S., Pate, P., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture," RFC 3985, March 2005.

[RFC4224] Pelletier, G., et al., "RObust Header Compression (ROHC): ROHC over Channels that can Reorder Packets," RFC 4224, January 2006.

[RFC4247] Ash, G., Goode, B., Hand, J., "Requirements for Header Compression over MPLS", RFC 4247, November 2005.

[RFC4364] Rosen, E., Rekhter, Y., "BGP/MPLS IP Virtual Private Networks (VPN)s", RFC 4364, February 2006.

[RFC4385] Bryant, S., et al., "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN," RFC 4385, February 2006.

[RFC4446] Martini, L., et al., "IANA Allocations for Pseudo Wire Edge To Edge Emulation (PWE3)," RFC 4446, April 2006.

[RFC4815] Jonsson, L-E., Sandlund, K., Pelletier, G., and P. Kremer, "RObust Header Compression (ROHC): Corrections and Clarifications to RFC 3095", RFC 4815, February 2007.

11. Contributors

Besides the editors listed below, the following people contributed to the document:

Bur Goode
AT&T
Phone: +1 203-341-8705
EMail: bgoode@att.com

Lars-Erik Jonsson
Optand 737
SE-831 92 Ostersund, Sweden
Phone: +46 70 365 20 58
EMail: lars-erik@lejonsson.com

Raymond Zhang
Infonet Services Corporation
2160 E. Grand Ave. El Segundo, CA 90025 USA
EMail: zhangr@bt.infonet.com

Editors' Addresses

Jerry Ash
AT&T
Email: gash5107@yahoo.com

Jim Hand
AT&T
Room MT A2-1A03
200 Laurel Avenue
Middletown, NJ 07748, USA
Phone: +1 732-420-3017
EMail: jameshand@att.com

Andrew G. Malis
Verizon Communications
40 Sylvan Road
Waltham, MA 02451 USA
EMail: andrew.g.malis@verizon.com

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.