Network Working Group
Internet Draft
<draft-ash-avt-ecrtp-over-mpls-protocol-02.txt>
Expiration Date: June 2005

Jerry Ash
Bur Goode
AT&T
Jim Hand
Consultant
George Swallow
Cisco Systems, Inc.

December, 2004

# Protocol Extensions for ECRTP over MPLS

## Status of this Memo



## Copyright Notice



## Abstract

VoIP typically uses the encapsulation voice/RTP/UDP/IP. When MPLS labels are added, this becomes voice/RTP/UDP/IP/MPLS-labels. For an MPLS VPN, the packet header is at least 48 bytes, while the voice payload is often no more than 30 bytes, for example. Header compression can significantly reduce the overhead through various compression mechanisms, such as enhanced compressed RTP (ECRTP). We consider using MPLS to route ECRTP compressed packets over an MPLS LSP without compression/decompression cycles at each router. Such an ECRTP over MPLS capability can increase the bandwidth efficiency as well as processing scalability of the maximum number of simultaneous compressed flows that use header

compression at each router.  In this draft we propose to use RSVP-TE extensions to signal the header compression context and other control messages between the ingress and egress LSR.  We re-use the methods in ECRTP to determine the context.

Table of Contents



Specification of Requirements

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 [RFC2119].

## 1. Introduction

Voice over IP (VoIP) typically uses the encapsulation voice/RTP/UDP/IP. When MPLS labels [MPLS-ARCH] are added, this becomes voice/RTP/UDP/IP/MPLS-labels.  For an MPLS VPN (e.g., [MPLS-VPN], the packet header is at least 48 bytes, while the voice payload is often no more than 30 bytes, for example.  The interest in header compression is to exploit the possibility of significantly reducing the overhead through various compression mechanisms, such as with enhanced compressed RTP [ECRTP], and also to increase scalability of header compression. We consider using MPLS to route ECRTP compressed packets over an MPLS LSP (label switched path) without compression/decompression cycles at each router.  Such an ECRTP over MPLS capability can increase bandwidth efficiency as well as the processing scalability of the maximum number of simultaneous compressed flows that use header compression at each router.

To implement ECRTP over MPLS, the ingress router/gateway would have to apply the ECRTP algorithm to the IP packet, the compressed packet routed on an MPLS LSP using MPLS labels, and the compressed header would be decompressed at the egress router/gateway where the ECRTP session terminates.  Figure 1 illustrates an ECRTP over MPLS session established on an LSP that crosses several routers, from R1/HC --> R2 --> R3 -->

R4/HD, where R1/HC is the ingress router where header compression (HC) is performed, and R4/HD is the egress router where header decompression (HD) is done.  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.

```
                     _____
                    |     |
                    |R1/HC| ECRTP Header Compression (HC) Performed
                    |_____|
                       |
                       | voice/ECRTP/MPLS-labels
                       V
                     _____
                    |     |
                    | R2  |
                    |_____|
                       |
                       | voice/ECRTP/MPLS-labels
                       V
                     _____
                    |     |
                    | R3  |
                    |_____|
                       |
                       | voice/ECRTP/MPLS-labels
                       V
                     _____
                    |     |
                    |R4/HD| ECRTP Header Decompression (HD) Performed
                    |_____|
```

Figure 1. Example of ECRTP over MPLS over Routers R1 --> R4

In the example scenario, ECRTP header compression therefore takes place between R1 and R4, and the MPLS path transports voice/ECRTP/MPLS-labels instead of voice/RTP/UDP/IP/MPLS-labels, saving 36 octets per packet. The MPLS label stack and link-layer headers are not compressed. Therefore ECRTP over MPLS can significantly reduce the header overhead through compression mechanisms.

Goals and requirements for header compression over MPLS are discussed in [MPLS-HC-REQ]. The solution put forth in this document meets these Requirements. In particular, the ECRTP over MPLS solution:

a. uses existing protocols [ECRTP] to compress RTP/UDP/IP headers,
b. allows HC over an MPLS LSP, and thereby avoids hop-by-hop compression/decompression cycles,
c. minimizes incremental performance degradation due to increased delay, packet loss, and jitter, by leveraging a compression scheme [ECRTP] that is less sensitive to delay, packet loss, and jitter, as well as by eliminating multiple compression/decompression cycles as a source of performance degradation, over a range of network path characteristics,
d. uses standard protocols [RSVP-TE] to signal context identification

and control information,
e. packet reordering does not cause incorrectly decompressed packets to be forwarded from the decompressor [ECRTP].

Section 2 presents the proposed protocol extensions, and Section 3 presents the procedures.

## 2. Protocol Extensions

Extensions to MPLS signaling are needed to signal the session context IDs (SCIDs) between the ingress and egress routers on the MPLS LSP. For example, new objects need to be defined for [RSVP-TE] to signal the SCIDs between the ingress and egress routers, and [ECRTP] used to determine the FULL_HEADER packets for the context identification; these FULL HEADER packets then contain the SCID identified by using the RSVP-TE objects. These protocol extensions need coordination with other working groups (e.g., MPLS).

### 2.1 ECRTP over MPLS Header Compression & Call Flows

The goal of ECRTP header compression is to reduce the IP/UDP/RTP headers to 4 bytes for most packets, since ECRTP requires that the UDP checksums be sent. In ECRTP header compression, the first factor-of-two 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 connection. 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 and therefore the second-order difference is zero.

By maintaining both the uncompressed header and the first-order differences in the session state shared between the compressor and decompressor, all that must be communicated is an indication that the second-order difference was zero. In that case, the decompressor can reconstruct the original header without any loss of information simply by adding the first-order differences to the saved uncompressed header as each compressed packet is received. The compressed packet carries the SCID, to indicate in which session context that packet should be interpreted. Since compressed packets are assumed to be routed on a separate LSP, set up by RSVP-TE, the decompressor uses the incoming MPLS label and the SCID to locate the proper decompression context.

In Figure 1 we assume an example VoIP flow set up from R1/HC --> R2 --> R3 --> R4/HD, where R1/HC is the ingress router where header compression (HC) is performed, and R4/HD is the egress router where header decompression (HD) is done, and in the reverse direction. Each router functions as an LSR and supports RSVP-TE signaling of LSPs. A summary of the ECRTP flow setup is as follows:

**1. R1/HC sends an RSVP-TE PATH message to R4/HD, which includes a**
SCID_Request object with a 2-byte ECRTP-Flow-ID.
**2. R4/HD assigns a unique 2-byte SCID to the call and sends an RSVP-TE**
RESV message to R1/HC that includes a Header_Compression_Reply object with the ECRTP-Flow-ID and the assigned SCID.
**3. R1/HC sets the SCID in compressed packets and FULL_HEADER packets.**

**4. Compressed packets and header compression control packets**
(FULL_HEADER and CONTEXT_STATE packets) are routed on a separate LSP, set up by RSVP-TE, from non-compressed packets; FULL-HEADER packets are routed on the same R1/HC --> R4/HD LSP as the R1/HC to R4/HD compressed packets for the ECRTP flow; CONTEXT-STATE packets are routed on the same R4/HD --> R1/HC LSP as the R4/HD to R1/HC compressed packets for the ECRTP flow.

5. **compressed packets, FULL_HEADER packets, and CONTEXT_STATE packets** are routed with MPLS labels.
6. **R4/HD uses the incoming MPLS label and the SCID to locate the proper** decompression context.
7. **if needed to resync, R4/HD sends a CONTEXT_STATE packet to R1/HC with** SCID set; R1/HC resends FULL_HEADER packet with SCID set to R4/HD, etc.
8. **R4/HD frees up the SCID when the ECRTP flow disconnects (e.g., as** indicated by SIP BYE message and RSVP-TE/PATH-TEAR messages), or by refreshing the PATH state.

## 2.2 Link Layer Packet Type Field

The encodings in ECRTP use a different link layer packet type field for each of 9 ECRTP packet types. Since it is necessary to identify packet types for ECRTP packets over MPLS (e.g., FH packets and compressed packets), it is proposed in this Section to add a 4-bit packet type field in the ECRTP header to encode the 9 different packet types.

[cRTP-ENCAP] uses a separate link-layer packet type defined for header compression. Using a separate link-layer packet type for every packet type used in header compression avoids the need to add extra bits to the compression header to identify the packet type. However, this practice does not extend well to MPLS encapsulation conventions [MPLS-ENCAP], in which a separate link-layer packet type translates into a separate LSP for each packet type. So for ECRTP over MPLS VPNs, each packet type defined in ECRTP MUST have prepended to it a packet type field. This field adds 1 octet to the header, and is encoded as follows (most significant bit is 0):

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0 0|  Packet Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Defined by PPP Data Link Layer|
|         Protocol              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

    SCID_Packet_Type designation:
    00000000 (first byte)

    "Packet Type" encoding:
    0: Reserved
    1: FULL_HEADER
    2: COMPRESSED_TCP
    3: COMPRESSED_TCP_NODELTA

4: COMPRESSED_NON_TCP
5: COMPRESSED_RTP_8
6: COMPRESSED_RTP_16
7: COMPRESSED_UDP_8
8: COMPRESSED_UDP_16
9: CONTEXT_STATE
10 - 255: RESERVED

As discussed in [ECMP-AVOID], since this MPLS payload type in not IP, the first nibble is set to 0000 to avoid being mistaken for IP.  This is also consistent with the proposed encoding of the PWE3 control word [PWE3-CNTL-WORD].

### 2.3 Header Compression Object Formats

A new L3PID (ethertype), XXXX, is defined in [RSVP-TE] for ECRTP over MPLS LSPs.  This is needed to define the type of traffic used in RSVP-TE Label Request Objects.  An SCID_Request object and Header_Compression_Reply object are defined in this section.  R1/HC creates an LSP to R4/HD by creating an RSVP-TE PATH message that contains:

a. Label_Request object with the L3PID for ECRTP over MPLS (XXXX - TBD),
b. an SCID_Request object.

R1/HC will receive a RESV message containing a Label object and a Header_Compression_Reply object.  R1/HC uses the label and SCID to send compressed traffic to R2/HD.

#### 2.3.1 SCID_Request Object

The Class for Header Compression Objects is of the form 10bbbbbb (need an official number from IANA).  The form 10bbbbbb allows intermediate nodes which do not understand header compression to silently drop the compression object.  This ensures that an LSP either exists between the source and the destination or that header compression is disabled.

```
      Class = Header Compression Object, Type = 1

       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |Num ECRTP-Flows |              ECRTP-Flow-IDs                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |          ECRTP-Flow-IDs Continued            |       PAD      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

#### 2.3.2 Header_Compression_Reply Object

The presence of this object in a RESV message indicates that the receiver will act as a decompressor for packets sent on this LSP which contain one of the listed SCIDs.  Over the life of an RSVP-TE session SCIDs may be added and deleted simply by refreshing the PATH state with the updated set of objects  This object provides synchronization between the sender and receiver as to which SCIDs may be used.

Class = Header Compression Object, Type = 2

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Num SCIDs   |                   SCIDs                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              SCIDs Continued                 |      PAD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        ECRTP-Flow-IDs                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          ECRTP-Flow-IDs Continued            |      PAD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

## 3. ECRTP over MPLS Procedures

### 3.1 Control Plane Procedures

The MPLS control-plane uses RSVP-TE to a) establish LSPs and label bindings between each GW-GW pair, b) to establish and control ECRTP over MPLS, and c) to provide resource reservation and bandwidth allocation for the varying number of calls on a GW-GW pair. The following procedures apply only to unicast sessions (extension to multicast is for further study) and discuss processing at the source, intermediate and destination nodes.

ECRTP over MPLS is always initiated by the originator of the PATH message, which we refer to as the source.  Note that the initiator of the RSVP-TE session may or may not be the ultimate source of the compressed flow.  For instance a Cable Modem Termination System (CMTS) in a packet cable environment might serve as the compressor for a ECRTP flow across an MPLS backbone. An ingress router can determine which flows to do header compression by applying procedures discussed in RFC 3006.

The source requests SCID assignments from the decompressor and the decompressor assigns the SCIDs.

For ECRTP header compression, the compressor and decompressor follow the procedures in [ECRTP], including the sending of FULL-HEADER packets, compressed packets, CONTEXT_STATE packets, etc.

Compressed packets and header compression control packets (FULL_HEADER and CONTEXT_STATE packets) are routed on a separate LSP, set up by RSVP-TE, from non-compressed packets.  FULL-HEADER packets are routed on the same R1/HC --> R4/HD LSP as the R1/HC to R4/HD compressed packets for the ECRTP flow.  CONTEXT-STATE packets are routed on the same R4/HD --> R1/HC LSP as the R4/HD to R1/HC compressed packets for the ECRTP flow.

The SCID-Request Object is included in an RSVP-TE PATH message. This object MUST not be included if a LABEL_REQUEST object is not also included in the PATH message.

Intermediate nodes must act to ensure that an LSP exists from source to destination. Thus if an intermediate node forwards a PATH message without a label request, the node MUST drop the HC Object from the PATH message. The HC object class is set to a value which indicates to nodes in the PATH which do not understand the object that they are to silently drop the object. This has the effect of allowing the RSVP-TE session while disabling header compression. This ensures that a HC unaware node will not inadvertently allow a discontinuity in the LSP.

If the destination node receives a PATH message with HC objects and is willing to act as a decompressor for this session and these ECRTP-Flow-IDs, it includes the SCIDs in a HC_REPLY object in the corresponding RESV message.

### 3.2 Data Plane Procedures

The source node compresses the header by removing the header and forming the compressed header, which is prepended to the remainder of the packet. The SCID and the MPLS header are then prepended and the packet is sent. Note that the compressor MUST not use a SCID until it has received a RESV message which contains a HC_REPLY with the SCID listed.

The destination node removes the MPLS header and the compressed header. The node prepends the header template to the packet and then uses the operands to populate the variable fields of the header with the values sent in the compressed header.

For ECRTP header compression, the compressor and decompressor follow the procedures in [ECRTP], including the sending of FULL-HEADER packets, compressed packets, CONTEXT_STATE packets, etc.

## 4. Security Considerations

These procedures do not change the trust model of [RSVP] and [RSVP-TE]. As such no additional security risks are posed.

## 5. Acknowledgements

Thanks to Curtis Villamizar and Stewart Bryant for helpful suggestions.

## 6. IANA Considerations

As discussed in Section 3.2, a new L3PID (ethertype), XXXX, needs to be assigned for ECRTP over MPLS LSPs.

## 7. Normative References

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

[MPLS-HC-REQ] Ash, G., Goode, B., Hand, J., "Requirements for Header Compression over MPLS", work in progress.

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

[RSVP] Braden, R. et al., "Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification", RFC 2205, September 1997.

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

## 8. Informative References

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

[cRTP-ENCAP] Engan, M., Casner, S., Bormann, C., "IP Header Compression over PPP", RFC 2509, February 1999.

[ECMP-AVOID] Swallow, G., et. al., "Avoiding Equal Cost Multipath Treatment in MPLS Networks," work in progress.

[MPLS-ARCH] Rosen, E., et. al., "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.

[MPLS-ENCAP] Rosen, E., et. al., "MPLS Label Stack Encoding", RFC 3032, January 2001.

[MPLS-VPN] Rosen, E., Rekhter, Y., "BGP/MPLS VPNs", RFC 2547, March 1999.

[PWE3-CNTL-WORD] Bryant, S., et. al., "PSE3 Control Word for use over an MPLS PSN," work in progress.

## 9. Intellectual Property Considerations

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.  Information on the procedures with respect to rights in RFC documents can be found in BCP **78 and BCP 79.**

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

## 10. Authors' Addresses

Jerry Ash
AT&T
Room MT D5-2A01
**200 Laurel Avenue**
Middletown, NJ 07748, USA
Phone: +1 732-420-4578
Email: gash@att.com

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

Jim Hand
Consultant
Phone : +1 732-532-3020
E-mail: James.Hand@mail1.monmouth.army.mil

George Swallow
Cisco Systems, Inc.
**250 Apollo Drive Chelmsford, MA 01824**
Phone: +1 978-497-8143
Email: swallow@cisco.com