A LightWeight IP Encapsulation (LIPE) Scheme
<draft-chuah-avt-lipe-02.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This document is submitted to the Audio/Video Transport Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the rem-conf@es.net mailing list.

Distribution of this memo is unlimited.

Abstract

Several application level multiplexing/compression schemes have been proposed in the IETF Audio/Video Transport (AVT) Working Group [1][2][9] to improve the transport efficiency of packet-voice traffic over IP-based networks. These approaches generally assume voice packets are encapsulated in RTP/UDP/IP by the communicating end-points (e.g., IP phones, mobile terminal, media gateways, etc.). In some transport scenarios, e.g., CDMA/GSM cellular networks, using RTP for voice packet is unnecessary as only the data transfer services provided by the IP/UDP layer, not the media control functions of RTP, are required.

This document describes a lightweight IP encapsulation scheme to multiplex low bit rate audio (or multimedia) packets into a single UDP/IP session. This document is the product of the AVT Working
Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the ietf-avt@merit.edu mailing list.
Applicability

These extensions are intended for those implementations which desire to multiplex small data packets together into one IP/UDP protocol data unit (PDU).

gb
1. Introduction

As the Internet evolves into a ubiquitous communication infrastructure, IP based technologies have also become more sophisticated. As a consequence of the maturing of the IP technology, it is now evident that the previously separate data and voice networks are converging to provide integrated communications services, including data, voice and video. While data packets can often be quite large, voice packets are in general rather small. Codecs in IP phones or IP telephony gateways compress the incoming speech samples and generate speech frames with sizes ranging from 5 to 20 bytes per speech sample. For example, G723.1 generates a 20-byte speech frame at 30 ms intervals [3]. Many codecs used in cellular environments generate speech frames of size less than 10 bytes per speech sample.

Such small speech frame sizes are subject to a large transport overhead when transferred in individual IP packets. For example a 10-byte voice packet transmitted using UDP/IP encapsulation incurs an overhead of 28 bytes (20 byte IP header, plus possibly 8 bytes UDP header), or 280%. In addition, if each audio stream uses one UDP media session, the large number of packets will create heavy packet processing load for the intermediate media gateway devices that operate above the IP-layer, even if no processing of the user flow is actually required. The available UDP port numbers may also become a limiting factor on the maximum number of voice flows as media gateways are expected to handle tens of thousand voice flows.

Although the relatively high transport overhead may not constitute a critical traffic engineering factor in transport scenarios where network resources are plentiful, the situation is undesirable for most wireless access networks. For instance, a T1/E1 link to a wireless base station would only be able to support a third of the voice traffic currently supported by their native packet-voice transport schemes. The idea of pooling multiple user flows into a more efficiently multiplexed channel is highly appealing.

Fortunately, for many applications, it is possible to multiplex traffic from a large number of flows into a single IP packet to improve efficiency and scalability without incurring much multiplexing delay. For example, when long distance telephony is offered over the Internet, the IP telephony gateways or the mobile switching centers in a cellular network provide an interface between the existing circuit switched telephone networks (such as PSTN and cellular networks like CDMA/GSM/TDMA network) and the packet switched IP data networks. The voice calls between a pair of IP telephony gateways or the mobile switching centers can be multiplexed into a single UDP session. As another example, in a CDMA based cellular network, an IP network may be used as the access network by the wireless service.
provider to connect the base stations to the mobile switching centers, part of whose function is to select the reverse direction radio frames and duplicate the forward direction radio frames for mobiles in soft handoff. In this case, the radio frames from different mobiles handled by the same base station, which can be either voice or data, can also be multiplexed into a single UDP session.

Many such applications have stringent delay requirements. In the first example above, the usual transfer delay and delay jitter requirements for voice application apply. In the second example, the duplicate radio frames in the reverse direction must be received by the mobile switching center within a small time window for the frame selection.

RTP [4] is a protocol designed to provide various real time services to the application layer with no assumption on the underlying network providing timely delivery or quality-of-service commitments. It can be used when the network is not heavily loaded and the application it supports can adapt to the varying network conditions to some extent. To improve the transport efficiency, some multiplexing schemes have been proposed within the framework of RTP [1,2].

Many of the features of RTP are designed to provide media control information to cope with the unavailability of QoS guarantees from the underlying network at the application layer. As such guarantees become available in modern/future IP networks, some of these features become unnecessary. These features are also of limited value to non-RTP applications (e.g., most commercial wireless voice traffic). In this document, we propose to use a lightweight encapsulation scheme based on UDP/IP for multiplexing application sessions. LIPE is designed to support multimedia traffic including both voice and data. We also include some discussions on how UDP/IP header compression can be incorporated within LIPE to further improve encapsulation efficiency.

2. The LIPE Encapsulation Scheme

The Lightweight IP Encapsulation (LIPE) uses UDP/IP as the transport layer. Each LIPE encapsulated payload consists of a variable number of multimedia data packet (MDP). For each MDP, there is a multiplexing header (MH) that conveys transport and media specific information.

The format of the IP packet conveying multiple MDPs over UDP using a minimum size MH is shown in Figure 1.
The generic protocol stack for LIPE, assuming PPP in HDLC-like framing [RFC 1662], is as shown in Figure 2:

```
+----------+----------+----------+----------+----------+----------+----------+
|          | LIPE     |          | UDP      |          | IP       |          |
+----------+----------+----------+----------+----------+----------+----------+
|          |          |          |          | HDLC/PPP |          |          |
```

All LIPE packets on the same PPP link MUST use the LIPE/IP encapsulation depicted in Figure 1. Section 6 explains how IPCP is used to negotiate for LIPE.

### 2.1. Basic LIPE Frame Format

The Multiplexing Header (MH) comprises of two components: the MDP Header Extension bit (the E bit) and the MDP length field. Optional Extension Headers can be supported via the E bit. The MH format is shown in Figure 3.

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4
+-------------------------------+
| E  MDP  Extended Headers      |
+-------------------------------+
```

E bit: The Header Extension bit is the least significant bit of the MH header. It is set to one/zero to indicate the presence/absence of
an extension header longer than 2 bytes. If the E bit is set to zero, the multiplexed header contains a 7 bit MDP length and a 2 bytes extended header identifier. If the E bit is set to one, it means there are more fields after the EHI.

MDP Length: a 7 bit MDP length field. This field indicates the size of the entire MDP packet in bytes including the E bit, Length field and optional Extension Headers (if present). For Type 0 Extended Header, the actual length is as indicated in the 7 bit field. For Type 1 Extended Header, the actual length is 4 times the number expressed in the 7 bit field.

2.2. Extension Headers

Extension Headers are used to convey application specific information. They also facilitate customization of LIPE to incorporate additional transport/session level control functionality such as sequence number, voice/video quality estimator.

2.2.1. Type 0 Extended Header

The 16-bit Type 0 Extended Header (EH0) is the first field in any Extension Header. It is used to identify MDPs belonging to specific small voice/video flows. The format of an LIPE encapsulated payload with a EH0 Extension Header is shown in Figure 4(a).

```
1                   2
+-------------------++-------------------+
| +             + +     +                       |
|0+   Length    +X+ Seq +     FlowId            |
| +             + +  No +                       |
+-------------------++-------------------+
|                                               |
|                 MDPn Payload                  |
|                                               |
+-------------------++-------------------+
```

Figure 4(a) : A MH with a Type 0 EH

When X bit is clear, the following 3 bits are used as the Sequence Number. This means the FlowId field is 12 bits. This version is used mostly for applications that generate small packets e.g. voice. The 12-bit FlowId allows up to 4096 flows to be multiplexed into a single UDP/IP session.
2.2.2. Type 1 Extended Header

The 16-bit Type 1 Extended Header (EH1) is used to identify flows that generate large data packets. The format of an LIPE encapsulated payload with a Type 1 Extension Header is shown in Figure 4(b). The least significant bit of the 1st byte of EH1 is the X bit. When X is set to one, it indicates that a EOF bit and a 3-bit Seq Number fields exist. This means when X bit is set, the FlowId field is 11-bit. The EOF bit is set when there are more fragments to be transmitted. For the non-fragment case, this bit is clear.

```
  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +             + +E+Seq +                     |
|0+   Length    +X+O+ No  +   FlowId            |
| +             + +F+     +                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 4(b) : A MH with a Type 1 EHI field
```

2.2.3. Payload Identifier (PID)

If the E-bit is set, it means there is a Payload Identifier (PID) extension header following the Multiplexed Header Identifier field. The Payload Identifier field starts with a 4-bit Payload Type Identifier (PTI), a 4-bit PID Length and any additional payload specific data. The format of the PID field is illustrated in Figure 6.

```
  0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +             + + PID +                       |
| PTI   + LNGTH + PID Payload                  |
| +             +                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 5 : Format of the PID field
```

Thus, a MH with a Type 0 Multiplexed Header and the PID extended header will look like Figure 6.

```
  0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +             + + +                             |
|1+   Length    +O+ Seq +   FlowId              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Chuah, et al.            expires June 2001
2.3. Examples of LIPE Encapsulated Payloads

In this section, we show some specific LIPE examples:

2.3.1. Carrying raw IS95 voice packets

In this scenario, the E bit of the first MH header is set to zero to indicate that there is no PID field. Type 0 extended header is used.

![Figure 7(a): Carrying raw IS95 voice packets](image-url)

2.3.2. Carrying UMTS Interactive Data packets.

![Figure 7(b): Carrying UMTS Interactive Data packets](image-url)

Note that one can use the PID Type to indicate different wireless access technologies e.g. PID Type = 1 indicates IS95 network, PID Type = 2 indicates UMTS network.
In this scenario, Type 1 Extended Header is used where the E bit is set to one and the X bit is set to one. The FlowId is used to identify different user flows. The payload is the frame protocol (FP) PDU described in [TS25.413].

Note that in our encapsulation scheme, no field is provided in the header for error checking purposes. We rely on the link layer error detection capability (e.g., CRC field in HDLC or ATM/AAL5) and possibly the additional UDP checksum on LIPE/UDP/IP to provide for the overall LIPE payload error detection. We believe this level of protection is sufficient.

2.4. LIPE Signaling Channel

LIPE end-points need a mechanism to specify UDP destination port numbers for data transfer and negotiate FlowId. The LIPE signaling channel is designed to serve such purposes.

A specified Destination Port number is used to indicate the LIPE Signaling Channel. The format of the LIPE signaling message over this channel is illustrated in Figure 8.

```
+-+-+-+-+-+-+-+  +  +  +  +
|  IP  +  UDP  +  Type  +  LNG  +  Control Msg Payload |
| (20B) +  (8B) +(4bits)+(4bits)+                      |
+-+-+-+-+-+-+-+  +  +  +  +
```

Figure 8: LIPE Signaling Channel Message Format for IP/UDP Encapsulation

Here, the UDP Port Number xxx (TBD) identifies the LIPE Signaling Channel, the 4-bit LIPE Signaling Channel Type field is used to identify the LIPE Control Message Type, and the 4-bit Length (LNG) field identifies the length of Control Msg Payload expressed in bytes.

Seven types of control messages are currently defined:

Type = 0 Tunnel Set Up Request
Type = 1 Tunnel Set Up Response
Type = 2 Tunnel Teardown Request
Type = 3 Flow Set Up Request
Type = 4 Flow Set Up Response
Type = 5 Flow Teardown Request
Type = 6 Flow Handoff Request
Type = 7 Flow Handoff Response
All LIPE signaling messages are UDP/IP messages. We also assume that all LIPE signaling messages are retransmitted for a maximum of MAX_RETRY times. LIPE peers which send a LIPE signaling message will wait for a response and times out after MAX_Response_Time. When time out occurs, the LIPE sending peer will retransmit the LIPE signaling message for a maximum of MAX_RETRY times before giving up.

2.4.1. LIPE Signaling Message Exchange

2.4.1.1. LIPE Tunnel Set Up/Teardown Procedure

Two LIPE peers will first exchange tunnel set up request/response messages. After this exchange, the LIPE peers know which UDP port number to use for transporting LIPE data packets.

2.4.1.1.1. Tunnel Set Up Request Message

For Tunnel Set Up Request message, there is a 4 bit transaction identifier. This field is not incremented when a request is retransmitted. If necessary, later we can add QoS Parameter as an extension to this message to do QoS negotiation for the tunnel. When the tunnel is no longer needed, the LIPE peer sends a Tunnel Teardown message. This message can only be sent when there is no more active flow being
supported in the tunnel.

2.4.1.1.2. Tunnel Set Up Response Message

```
+-----------------+-----------------+-----------------+
|     Typ=1       |      Trid       |
|    +   +         +   +                           |
+-----------------+-----------------+-----------------+
|      Lngth       +                           |
+-----------------+-----------------+-----------------+
|                      Error Code                |
+-----------------+-----------------+-----------------+
```

Figure 10 (b): Tunnel Set Up Response Message

The Tunnel Set Up Response Message will contain the error codes.
Error Code = 0 means the tunnel set up is successful.

2.4.1.1.3. Tunnel TearDown Message

```
+-----------------+-----------------+-----------------+
|     Typ=2       |      Trid       |
|    +   +         +   +                           |
+-----------------+-----------------+-----------------+
|      Lngth       +                           |
+-----------------+-----------------+-----------------+
|      UDP Dest Port No |
+-----------------+-----------------+-----------------+
```

Figure 10 (c): Tunnel Teardown Message

The tunnel teardown message can only be sent when no more flows exist in the tunnel.

2.4.1.2. LIPE Flow SetUp/Teardown Procedure

```
LIPE Peer 1     LIPE Peer 2
|                  |
| Flow Set Up Request |------------------>
| Flow Set Up Response |<-----------------|
```

Figure 11: Flow Set Up Request Procedure
The Flow Set Up Request message can only be sent after the sending peer is sured that there exists at least a tunnel between itself and the receiving peer. The LIPE peer which receives a Flow Setup Request message will respond with a FlowSetup Response Message.

To teardown a flow and reclaim a flow identifier, a LIPE peer will send a Flow Teardown message.

### 2.4.1.2.1. Flow Set Up Request Message

```
1 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|       +       +               +
Typ=3  + Trid  + Lngth          +
|       +       +               +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|    RAB Id                    +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|    Dnlink FlowId             +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 12 (a): Flow Set Up Request Message

### 2.4.1.2.2. Flow Set Up Response Message

```
1 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|       +       +               +
Typ=4  + Trid  + Lngth          +
|       +       +               +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|    Error Code                +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|    RABId                     +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|    Uplink FlowId             +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 12 (b): Flow Set Up Response Message

Note that only if the Flow Set Up is successful will the Flow Set Up Response message contains RABId and a corresponding Uplink FlowId.
2.4.1.2.3.  Flow TearDown Message

```

+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
| Typ=5 + Trid + Lngth + |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
| RAB Id +      |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
| Dnlink + FlowId + |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
| Uplink + FlowId + |
```

Figure 12 (c): Flow Teardown Message

2.4.1.3.  LIPE Soft Handoff Procedure

```

    LIPE1     LIPE 2
  Peer       Peer

  Flow ^ | Flow
  Handoff | Handoff
  Response | v Request

  LIPE4     LIPE 3
  Peer       Peer
```

Figure 13: Soft Handoff scenario

For some scenarios where a flow needs to be extended to another LIPE pairs, one LIPE peer can send a soft handoff request message to a new LIPE peer to initiate soft handoff. The soft handoff request message will contain the existing RABID. In response, the new LIPE peer allocates a new flow identifier. The new LIPE peer (LIPE 3 Peer) can then initiate a flow set up request to LIPE4 peer (provided a LIPE tunnel already exists between LIPE 3 and LIPE 4).

2.4.1.3.1.  Soft Handoff Request Message

```

+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
| Typ=6 + Trid + Lngth + |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+---------------+

```
3. QoS

Besides the traditional best-effort service, other services such as integrated service (including controlled load service and guaranteed service) and differentiated service have been defined. These services, by reserving certain network resources such as bandwidth, can provide the traffic with certain guarantees such as delay and loss.

LIFE is compatible with both the DifServ and IntServ QoS models. To support multiple QoS classes, the DSCP bits of the IP header can be used to request a particular PHB e.g., for high quality voice the IP packet with multiplexed audio frames can be marked with the EF code point; for low quality voice, the IP packet can be marked with one of the appropriate AF code points.

LIFE specific QoS requirements may also be conveyed on an end-point specific basis via the TunnelID or FlowId field.

4. Multiplexing Policy

Given the link MTU L_max, a UDP/IP packet can carry payload of up to
L_max - H_ip - H_udp, where H_ip is the IP header length (20 bytes without option) and H_udp is the UDP header length (8 bytes). To limit the multiplexing delay, a multiplexing timer with a lifetime of T_mux is used. H_mh is the multiplexing header length. The encapsulation policy is as follows:

a) If the total size of the received radio frames plus that of that of their H_mh exceeds L_max - H_ip - H_udp, send all the MDP frames except the most recently received one (no fragmentation of MDP) in one UDP packet, and restart the multiplexing timer. The newly received MDP is held for multiplexing with upcoming MDPs.

b) If the multiplexing timer expires, send the accumulated MDPs in one UDP packet and restart the encapsulation timer.

5. UDP/IP Header Compression

Note that if IP headers are not required to do routing (say the underlying network is either ATM or MPLS), one can either remove or compress the UDP/IP header. That will increase further the bandwidth efficiency of using the LIPE scheme in a radio access network where the BSs have IP interfaces that run over ATM/MPLS networks.

When we map a certain IP tunnel into a particular MPLS/ATM connection, we need to ensure that the quality of service provided by the MPLS/ATM connection matches with the DSCP indicated in the IP header.

6. Security

This draft does not impose additional security considerations beyond those that apply to PPP and header-compression schemes over PPP.

7. Summary

LIPE is designed to support multimedia traffic when certain resource guarantees are available from the underlying network. It is based on UDP/IP or IP; hence is lightweight compared with other proposals based on RTP [1,2]. As IP based networks become more and more sophisticated and offer various levels of resource guarantees [5], this scheme is more suitable to the modern/future IP architecture compared with RTP based schemes.

A multiplexing policy is also outlined for LIPE.
8. References


[3] ITU-T Recommendation G.723.1 "Dual Rate Speech Coder for Multimedia Communications Transmitting At 5.3 and 6.3 Kbps", 1995


[9] T. Koren etc, Enhancements to IP/UDP/RTP Header Compression, work in progress, draft-koren-avt-crtp-enhance-01.txt


9. Intellectual Property Considerations

Lucent Technologies Inc. may own intellectual property in some on some of the technologies disclosed in this document. The patent licensing policy of Lucent Technologies Inc. with respect to any patents or patent applications relating to this submission is stated in the March 1, 1999, letter to the IETF from Dr Roger E. Stricker, Intellectual Property Vice President, Lucent Technologies, Inc. This letter is on file in the offices of the IETF Secretariat.
10. Acknowledgements

The authors wish to thank S. Abraham for useful discussions.

11. Authors’ Addresses

Mooi C. Chuah
Bell Laboratories
Lucent Technologies
101, Crawfords Corner Road,
Holmdel, NJ 07733
chuah@bell-labs.com
(732)-949-7206

Enrique J. Hernandez-Valencia
Bell Laboratories
Lucent Technologies
101, Crawfords Corner Road,
Holmdel, NJ 07733
enrique@bell-labs.com
(732)-949-6153
# Table of Contents

1. Introduction .................................................. 4  
2. The LIPE Encapsulation Scheme ............................... 5  
  2.1. Basic LIPE Frame Format .................................. 6  
  2.2. Extension Headers ......................................... 7  
    2.2.1. Type 0 Extended Header ............................. 7  
    2.2.2. Type 1 Extended Header ............................. 8  
    2.2.3. Payload Identifier (PID) ........................... 8  
  2.3. Examples of LIPE Encapsulated Payloads .................. 9  
    2.3.1. Carrying raw IS95 voice packets .................... 9  
    2.3.2. carrying UMTS Interactive Data packets .............. 9  
  2.4. LIPE Signaling Channel .................................. 10  
    2.4.1. LIPE Signaling Message Exchange .................... 11  
      2.4.1.1. LIPE Tunnel Set Up/Teardown Procedure .......... 11  
      2.4.1.1.1. Tunnel Set Up Request Message ................ 11  
      2.4.1.1.2. Tunnel Set Up Response Message ............... 12  
      2.4.1.1.3. Tunnel Tear Down Message .................... 12  
      2.4.1.2. LIPE Flow Set Up/Teardown Procedure ............... 12  
        2.4.1.2.1. Flow Set Up Request Message ................ 13  
        2.4.1.2.2. Flow Set Up Response Message ............... 13  
        2.4.1.2.3. Flow Tear Down Message .................... 14  
      2.4.1.3. LIPE Soft Handoff Procedure .................... 14  
        2.4.1.3.1. Soft Handoff Request Message ................ 14  
        2.4.1.3.2. Soft Handoff Response Message ............... 15  
  3. QoS ........................................................ 15  
  4. Multiplexing Policy ........................................ 15  
  5. UDP/IP Header Compression .................................. 16  
  6. Security ................................................... 16  
  7. Summary .................................................... 16  
  8. References ................................................ 17  
  9. Intellectual Property Considerations ....................... 17  
 10. Acknowledgements ........................................... 18  
 11. Authors’ Addresses ........................................ 18  

Chuah, et al.  expires June 2001  [Page 1]