Several application level compression/multiplexing solutions 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 RTP/UDP/IP encapsulated by the communicating end-points (e.g., IP phones, mobile terminal, media gateways, etc.). In some transport scenarios, using RTP/UDP/IP encapsulation for voice packet is unnecessary as only the data transfer service provided by the IP layer is required, not the media control functions.

This document describes a lightweight IP encapsulation scheme to multiplex low bit rate audio (or multimedia) packets into a single UDP/IP session or IP session. The decision to multiplexing audio
packets at the UDP or IP layer is left as a balance between data transport efficiency and implementation complexity among the communicating end-points across a routed IP network.

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

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

Table of Contents
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 service, including data, voice and video. While data packets can often be quite large, voice packets are in general rather small. Codecs at the IP telephony gateway which, compress the incoming speech samples, generate packets with sizes ranging from 5 to 20 bytes per speech sample. For example, G723.1 generates a 20 bytes speech packet at 30 ms intervals [3]. Many codecs used in cellular environments generate packets of size less than 10 bytes per speech sample.

Such small size packets are subjected to a large transport overhead if 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 required. The available UDP port numbers may also become a limiting factor on the maximum number of sessions.

Although the relatively high transport overhead may not constitute a critical traffic engineering factor in transport scenarios where network bandwidth is plentiful, the situation is quite different for many wireless access networks where T1/E1 links are used to carry packets from many voice users. There, the concept of pooling multiple user flows into a more efficiently multiplexed channel is highly appealing.

Fortunately, for many applications, it is possible to multiplex a large number of sessions 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 specific delay requirements. In the first example above, the usual transfer delay and delay jitter requirements for voice application applies. 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 done to provide even more savings.

2. The Encapsulation Scheme

The Lightweight IP Encapsulation (LIPE) uses either UDP/IP or 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 protocol and media specific information.

The format of an IP packet conveying multiple MDPs over UDP using a minimum size MH is shown in Figure 1 (a).
MH: Multiplexing Header
MDP: Multimedia Data Packet
(Length expressed in bytes)

Figure 1 (a): Lightweight IP/UDP Encapsulation Scheme

The format of an IP packet conveying multiple MDPs without UDP is
and using a minimum size MH shown in Figure 1 (b).
Note that a 1-byte tunnel-identifier (TID)
is included when the LIPE PDUs are conveyed directly over IP.

Figure 1 (b): Lightweight IP Encapsulation Scheme

The generic protocol stack for LIPE, assuming PPP in HDLC-like framing [RFC 1662], is as shown in Figure 2:

Figure 2: Protocol Stacks for LIPE

We assume that all LIPE packets on the same PPP link are encapsulated
either as in Figure 1 (a) or as in Figure 1 (b), but not both simultaneously. Section 6 explains how IPCP is used to negotiate for
either type of encapsulation format.
2.1. Basic

The Multiplexing Header (MH) comprises of two components: the 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.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +                  +                   |
| E+   Length    + Extension            |
| +                  +                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Multiplexing Header Format

E bit (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/absense of an extension header. If the E-bit is set to one, the first header extension MUST be a UserID header. Length: 7 bit length field. This field indicates the size of the entire MDP packet in bytes, including the E bit, Length field and optional extended headers (if they exist).

2.2. Extension

Extension headers are used to convey user specific information. It also facilitates the customization of LIPE to provide additional control information e.g. sequence number, voice/video quality estimator.

2.2.1. User

The 16-bit User Identifier is the first field in any Extension Header. It is used to identify MDPs belonging to specific user flows. The format of a LIPE encapsulated payload with a UserID extension header is shown in Figure 4. The least significant bit of the 1st byte of UserID is the X-bit. When set to one, it indicates that the extension header is longer than 2 bytes. Thus, effectively addressing range of the UserID field is 15-bit long.

```
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    +X+               UserId |
| +                  +                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
The 15-bit UserID allows up to 32768 flows to be multiplexed into a single UDP/IP session. The seemingly high number for the UserID is chosen for various reasons. First, many applications may be alive, but not active, for quite some time. These "dormant" applications will still hold on to the user identifiers. Therefore, the number of active applications that can actually contribute to the multiplexed channel may be much smaller. Second, in a mobile environment, when a mobile is handed off from one base station to another, the application on this mobile will have to be multiplexed into another UDP session in the new base station. Assuming each mobile takes up one identifier, if the UserID space is large enough, it is possible to make the UserID unique to the frame handler in the mobile switching center (or Radio Network Controller). Consequently during soft handoff between base stations controlled by the same frame handler (as shown in Figure 4), there is no need to reassign UserID thus saving signaling cost.

Figure 4 : A MH with a UserID field

Figure 5: Using the same UserID in the softhandoff scenario

Note that a service provider may decide to further subdivide the 15 bit UserID field into a user identifier field (e.g. 10 bits) and a
base station identifier field (e.g. 5 bits). Such subdivision is vendor-specific and can be done transparently (as long as the two peers understand the formats).

### 2.2.2. Payload

If the X-bit in the UserId field is set, it means there is a Payload Identifier (PID) extension header following the UserId 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.

```
+---------------------------------+
|       +       +                               |
| Type  + LNGTH +     Header Information     |
|       +       +                               |
+---------------------------------+
```

Figure 6: Format of the PID field

Thus, a MH with 2-byte UserId and the PID extended header will look like Figure 7.

```
+---------------------------------+
| +             + +                             |
|1+   Length    +1+           UserId            |
| +             + +                             |
+---------------------------------+

|  PID  +  PID  +     PID       +   Data        |
| Type  + LnG=2 +   Payload     +   Payload     |
|  1    +       +               +               |
+---------------------------------+
```

Figure 7: A MH with a UserId field and a PID field

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.

### 2.3. Examples

In this section, we show some specific LIPE examples:

In this example, it is assumed that each mobile terminal generates compressed RTP/UDP/IP packets. The MH header is only 1-byte long with
the E bit set to zero indicating that there is no extended header. The UserID is not used since the context ID [8] within the compressed RTP/UDP/IP header can uniquely identify the user.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +   Compressed         | +   Compressed         |
| 0+ Length 1 + RTP/UDP/IP | 0+ Length 2 + RTP/UDP/IP |
| +   PDU 1               | +   PDU 2               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8(a): Carrying Compressed RTP/UDP/IP packets

2.3.1. Carrying IS95 voice packets

In this scenario, the E bit of the first MH header is set to one to indicate that UserID exists. The X-bit is set to one indicating that there is a PID field.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| +   Length 1 + UserID |
| +   PDU 1               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PID + PID + PTI +Timing +Quality+ Seq |
| Type + LnG=3 + Info + # |
| 1 + + + + Ind + |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Data |
| Payload |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8(b): Carrying IS95 voice packets

In the example, PID Type =1 indicates that this is IS95 voice packets and the additional extended header is 3 bytes long. The additional information carried in the extended header is the IS95 packet type (PTI), some timing information, voice quality indicator, and sequence numbers.
2.3.2. carrying UMTS conversational/streaming packets

```
+-----------------------------+
| 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 |
+-----------------------------+
```

```
| 1+ Length +0+ UserId |
```

```
+-----------------------------+
| FP PDU                     |
+-----------------------------+
```

![Figure 8(c): Carrying UMTS conversational/streaming packets](image)

In this scenario, the E bit is set to one and the X bit is set to zero. The UserID is used to identify the 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. Instead we rely on the IP or UDP checksum to provide for the overall IP or UDP payload error detection. We believe this level of protection is sufficient since the IP packet carrying multiplexed audio (or multimedia) frames are not carried over the air.

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.

To support multiple QoS classes, we suggest using the DSCP bits of the IP header e.g. for high quality voice, we can mark the IP packet with multiplexed audio frames with EF code point; for low quality voice, we can mark it with one of the appropriate AF code points.

4. Multiplexing Policy

Given the link MTU $L_{\text{max}}$, a UDP/IP packet can carry payload of up to $L_{\text{max}} - H_{\text{ip}} - H_{\text{udp}}$, where $H_{\text{ip}}$ 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 (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 UDP, or (IP+TID), 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. Comparison of the LIPE scheme with other existing proposals

In this section, we compare 3 approaches for carrying multiplexed audio packets in terms of the overhead incurred. The 3 approaches considered are cUDP/PPPMux, tCRTP, and LIPE. We assume that PPP/HDLC is the Layer 2 technology used.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Protocol Details</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach 1</td>
<td>PPP/PPPMux cUDP</td>
<td>4 bytes PPP ID</td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td>cUDP/IP</td>
</tr>
<tr>
<td></td>
<td>PFF, length per stream</td>
<td>1 byte</td>
</tr>
<tr>
<td>Approach 2</td>
<td>PPP/HDLC tCRTP</td>
<td>4 bytes cIP</td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td>cUDP/IP</td>
</tr>
<tr>
<td></td>
<td>PFF, length per stream</td>
<td>1 byte</td>
</tr>
<tr>
<td></td>
<td>cUDP/IP</td>
<td>1 byte PPPMuxID</td>
</tr>
<tr>
<td></td>
<td>cUDP/IP</td>
<td>1 byte PPPID</td>
</tr>
</tbody>
</table>

Chuah, et al. expires December 2000
Approach 3 LIPE PPP/HDLC  4 bytes cUDP/IP  3 byte
   per stream length 1 byte
         UID  0-2 bytes
         payload

We see that for approach 3, the per stream overhead is 1-3 bytes
while for approaches 1 & 2, the per stream overhead is 4 bytes.

7. PPP

A new LCP Configuration Option is used to request LIPE operation for
the PPP link. A summary of the LIPE Configuration Option format for
the Link Control Protocol (LCP) is shown below. The fields are
transmitted from left to right.

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

Type
TBD

Length
3
```

The new LCP option is used only as a hint to the peer that LIPE
operation is preferred by the sender. Support of LIPE operations is
negotiated in each direction independently. Acknowledgement of the
LIPE LCP Option (TBD) does not obligate a peer to transmit LIPE
frames. Non LIPE-speakers SHOULD instead send LCP Configure-Reject
for the option.

If either LCP Configure-Nak or LCP Configure-Reject is received for
this option, then the next transmitted LCP Configure-Request MUST NOT
include this option. If the received LCP Configure-Request message
does not contain a LIPE LCP option, an implementation MUST NOT send
an unsolicited Configure-Nak for the option.

(An implementation of LIPE that is already in LIPE framing mode and
receives this option in an LCP Configure-Request message MAY, both
for clarity and for convergence reasons, elect to send LCP
Configure-Ack. It MUST NOT restart LCP nor change framing modes in
this case.)
The size of a LIPE encapsulated frame MUST NOT exceed the maximum receive unit (MRU) size negotiated during LCP [10].

8. Negotiating usage of LIPE

When the layer 2 protocol is PPP, the usage of various LIPE configuration options can be negotiated via the IP Compression Protocol option of IPCP:

IPCP Option 2: IP configuration protocol
  Network Protocol: TBD indicates LIPE
  sub-option 1 enables use of IP session
  sub-option 2 enables use of UDP/IP session

9. Security

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

10. 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.

The 15-bit UserID field in LIPE facilitates its usage in the third generation wireless system where handoffs may occur during the lifetime of a session. By having a larger user space, we can greatly reduce the signalling overhead due to identifier re-negotiation during a handoff.

A multiplexing policy is also outlined for LIPE.

11. References


IP Telephony Gateway, work in progress, draft-ietf-avt-mux-rtp-00.txt, Aug, 1998

[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


12. Intellectual Property Considerations

Lucent Technologies Inc. may own intellectual property in 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.

13. Acknowledgements

14. Authors’ Addresses

Mooi C. Chuah
Bell Laboratories
Lucent Technologies
101, Crawfords Corner Road,
# Table of Contents

1. Introduction .................................................. 4  
2. The Encapsulation Scheme ...................................... 5  
   2.1. Basic ..................................................... 7  
   2.2. Extension ................................................ 7  
      2.2.1. User .................................................. 7  
      2.2.2. Payload .............................................. 9  
   2.3. Examples .................................................. 9  
      2.3.1. Carrying .............................................. 10  
      2.3.2. carrying .............................................. 11  
3. QoS .......................................................... 11  
4. Multiplexing Policy .......................................... 11  
5. UDP/IP Header Compression .................................... 12  
6. Comparison of the LIPE scheme with other existing proposals  
7. PPP .......................................................... 13  
8. Negotiating usage of LIPE .................................... 14  
9. Security ....................................................... 14  
10. Summary ....................................................... 14  
11. References .................................................... 14  
12. Intellectual Property Considerations .......................... 15  
13. Acknowledgements ............................................ 15  
14. Authors’ Addresses .......................................... 15  

TO-HERE