1. Introduction

Header overhead is a key issue for communication sessions when packets have small payloads. For example, each packet in an RTP stream contains an RTP, UDP and IP header, a total of 40 bytes. When RTP is used for carrying voice data in a packet network like the Internet, this header overhead can be large since the size of the packet is relatively small. For instance, the G.723.1 codec for voice data compression with 30 ms packetizing interval generates frames of size 20 bytes only (The G.723.1 compresses a 64kbs voice stream into 5.3 kbps...
stream). If every frame is sent in an RTP packet, this means that only 33% of the total size of the packet is user data.

Several drafts for RTP streams multiplexing have been presented to the IETF Audio/Video Transport (AVT) group [Tani98][Rose98][Subb98]. These drafts presented various approaches for multiplexing audio streams between peer gateways. In these drafts, multiplexing and de-multiplexing are implemented at the gateway, which provides an interface between the Public Switch Telephone Network (PSTN) and the Internet.

All the presented drafts require the existence of a gateway between the communicating entities, and are not applicable to non-VoIP flows. Research into IP telephony is going toward an end-to-end IP telephony without going through a gateway. Although it will take some time before gateways become obsolete, the need for multiplexing small packets will always be here.

This draft proposes a light-weight data driven approach for multiplexing low bit rate RTP streams at the edge router in order to reduce the RTP/UDP/IP header overhead associated with each RTP stream. The RTP/UDP/IP header is replaced with a mini-header at the ingress router. The egress router reproduces the original packet (except for the RTP timestamp and sequence number fields) using the information in the mini-header and a mapping table.

2. RTP Streams Multiplexing Scheme

2.1 Overview

During the past 10 years, the world of telecommunication has seen unprecedented revolutions in the way people communicate with each other. Sending mails and placing calls with remote parties was never easier: "point and click" and you are connected to your party or your email is in the mail box of the recipient. At the heart of this revolution is the Internet that provides communication between local access networks.

The Internet attracted the attention of people from different fields and classes. Internet applications nowadays cover all aspects of life, such as online-shopping, tele-teaching, tele-medicine, online banking to list a few. This vast acceptance of Internet applications into our daily life imposed pressure on the bandwidth providers to keep their customers satisfied. Internet users are always asking for more bandwidth and bandwidth providers are always lacking behind the demands.

With this picture in mind, many research groups turned attention to tools and techniques to help the Internet coop with the big demand for bandwidth; compression and multiplexing are at the heart of this
direction. While compression mechanisms strive to represent the user’s data in the minimum amount of bits, multiplexing algorithms try to keep the transmission protocol overhead to a minimum.

The main driving force behind multiplexing was the reduction in the header overhead associated with headers stacked from several protocol layers. At the heart of the multiplexing approach was the assumption that at any time, there is more than one user communicating with the same remote location. Another key to the use of multiplexing is that the data of the users (referred to as payloads) are relatively small compared to the additional overhead imposed by the network to pass the data between the sender and the receiver. Voice-over-IP applications provide a typical environment where multiplexing of voice streams from different users can improve the bandwidth utilization in the IP network.

Figure 1 depicts a situation where two local access networks A and B are connected to the Internet via the access routers RA and RB respectively. All packets generated in network A and destined to network B, have to go through the edge routers RA and RB. Two users UA1 and UA2 use voice streams to communicate with remote users UB1 and UB2. Packets from UA1 and UA2 could be multiplexed at the router RA and de-multiplexed at router RB and vice versa for packets generated from users UB1 and UB2.

![Figure 1. The Internet with two access networks and two edge routers.](image)

### 2.2 Mini-Header

Even though the proposed multiplexing scheme can be implemented in any scenario similar to the one just described, we will focus on the case of multiplexing RTP streams used for VoIP applications. In a VoIP application, the voice input is sampled, digitized, compressed and framed at the devices connected to the access network (microphones...
connected to computers, IP phones, or gateway). The RTP, UDP and then IP header are added to each frame (payload) before it is sent to the edge router of the local access network.

In order to reduce the header overhead in the Internet, the RTP/UDP/IP header of each packet is replaced with a mini-header at the edge router. The mini-header is a 2-byte tag that replaces the original header at the ingress router, and helps to reconstruct the original packet at the egress router. Section 2.3 gives a complete description of all fields in the mini-header. Each of the access routers will keep a mapping table that stores the association between the mini-header and the original header. When a packet from the local access network arrives at the ingress router, the mapping table is searched for a match using the source and destination IP address and port number as search key. In case no match was found (the packet is the first packet in the stream), a mini-header is generated for the stream, and a new entry with the mini-header is added to the mapping table. To pass the association between the RTP/UDP/IP header and the mini-header to the peer router, the ingress router creates a void packet with only the mini-header and the RTP/UDP/IP header (exactly 40 bytes) with the Payload Length field in the mini-header set to zero (0). This packet will be sent before other packets from the same stream to ensure that the payload is not lost at the egress router. Another packet with the mini-header and the payload is created, and both packets are sent through the multiplexed connection to the egress router. In case a match was found in the table (previous packets of the same stream have already passed through), the RTP/UDP/IP header is replaced with a mini-header constructed using the CID stored in the mapping table, and the payload size computed from the size of the IP packet.

When the egress router receives the multiplexed packet, it reads the mini-header for each multiplexed stream. Information in the mini-header can tell whether the mini-header is followed by an RTP/UDP/IP header or by a payload and what is the size of the payload. When the packet carries a payload, the mini-header is taken out, and the system searches the mapping table using the ingress router IP address and port number, the egress router port number and the CID from the mini-header as a search key. The RTP/UDP/IP header from the mapping table is then added to the packet, and the packet is sent to its destination.

In case the mini-header was followed by the RTP/UDP/IP header, the mapping table will still be searched. In case the search was successful, the matching entry will only be refreshed by updating the Last_Time_Refreshed field. The payload type in the entry will also be updated in case it is different from the payload type stored in the mini-packet. If the search failed (first packet in the stream), a new entry for the stream is created in the mapping table.
2.3 Mini-header Format

Figure 2 shows the format of the 2-byte mini-header. Only necessary information to reconstruct the original RTP/UDP/IP header is stored in the mini-header. Following are the entries and their meanings:

- **Channel or Call ID (CID: 8 bits):** This 8-bit field can support 256 different CIDs. The CID is used to identify the stream at the egress router.

- **Extension bit (X: 1 bit):** An extension header is used for packets with length larger than 128 bytes. The extension header is 2 bytes, and it follows directly the mini-header; when it is present (the X bit is set to one), it indicates the size of the payload in the mini-packet.

- **Payload Length (PL: 7 bits):** ONLY payload size in bytes. Able to support payloads with sizes up to 128 bytes. A value zero (0) in the Payload Length indicates that ONLY the full RTP/UDP/IP header is included after the mini-header.

```
       _0_1_2_3_4_5_6_7_8_9_0_1_2_3_4_5_
         |CID   |X   |PL   |
         ---------------------------------  
     | Extension Header                    |
         ---------------------------------  

Figure 2. Format of the mini-header with the extension header
```

2.4 Mapping Tables

Figure 3 and 4 show the mapping tables at the ingress and egress routers respectively. Table 1 list all the abbreviations used in these mapping tables. The fields Source User IP Address, Source User Port Number, Destination User IP Address, Destination User Port Number, Ingress Router IP Address, Ingress Router Port Number, Egress Router IP Address, and Egress Router Port Number are self-explanatory.

The payload type for each stream is also stored in the table (Payload Type) to accommodate for adaptive applications. Adaptive applications can change the coding scheme during the life time of one session, depending on several factors, such as the user’s request or the status of the network. At the ingress router, the payload type of the incoming packet is compared to the payload type stored in the mapping table (the same payload that is stored in the mapping table at the egress router). In case the payload types are different, the whole RTP/UDP/IP header is sent to the egress router (the payload type of each packet is included in the RTP header). This also triggers the change of the payload type in the mapping table at the egress router and the destination user.
The Channel IDentifier (CID) of each stream is assigned at the ingress router. When a new stream arrived at the ingress router, the IP address of the egress router is identified. In case there exist already a multiplexed channel between the two routers, with a free CID, this CID is assigned to the new stream; otherwise, the ingress router signals the egress router to open a new multiplexing channel. (See section 3 for more information on control signaling)

Because of the limited space for the CID, CIDs for terminated streams have to be reclaimed and re-used by new streams. In order to reduce the control signaling overhead between peer routers, we added the Last_Time_Refreshed (LTR) field to the mapping table at the ingress and egress routers. When a packet is received at the ingress router, the current time of the system is compared to the value of the Last_Time_Refreshed field from the mapping table; in case the difference is larger than a certain constant value, Delta, the association between the RTP/UDP/IP header and the mini-header is refreshed by sending a packet containing only the RTP/UDP/IP header and the mini-header. This also triggers the egress router to update the corresponding entry in the mapping table to the current time of the system. To reclaim CIDs, ingress and egress routers scan their routing tables periodically and remove entries with the time difference between the current system’s time and Last_Time_Refreshed larger than a certain constant value, Alpha. In order to allow the ingress routers to refresh the entries for all the on-going streams, this value Alpha must be larger than Delta.

The constant Delta should be small enough to allow CIDs reuse and to avoid sending packets to an already terminated session, but it should be large enough to increase the time interval between consecutive packets containing the whole RTP/UDP/IP header with the mini-header.

The field Last Packet Reproduced Sequence Number is included in the mapping table at the egress router to help reproduce the sequence number field in the RTP header of the packet. Section 2.7 talks with more details about this issue.
<table>
<thead>
<tr>
<th>Abbreviation used</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP</td>
<td>Source User IP Address</td>
</tr>
<tr>
<td>Source Port#</td>
<td>Source User Port Number</td>
</tr>
<tr>
<td>Destination IP</td>
<td>Destination User IP Address</td>
</tr>
<tr>
<td>Destination Port#</td>
<td>Destination User Port Number</td>
</tr>
<tr>
<td>PT</td>
<td>Payload Type</td>
</tr>
<tr>
<td>IRouter IP</td>
<td>Ingress Router IP Address</td>
</tr>
<tr>
<td>ERouter IP</td>
<td>Egress Router IP Address</td>
</tr>
<tr>
<td>ERouter Port#</td>
<td>Egress Router Port Number</td>
</tr>
<tr>
<td>CID</td>
<td>Channel Identifier</td>
</tr>
<tr>
<td>LTR</td>
<td>Last_Time_Refreshed</td>
</tr>
<tr>
<td>LPR Time.</td>
<td>Last Packet Reproduced Timestamp</td>
</tr>
<tr>
<td>LPR Seq#</td>
<td>Last Packet Reproduced Sequence Number</td>
</tr>
</tbody>
</table>

Table 1. Abbreviations used in the mapping tables

<---- Search Key ---->

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>PT</th>
<th>IRouter</th>
<th>ERouter</th>
<th>CID</th>
<th>LTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Port#</td>
<td>IP</td>
<td>Port#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>----</td>
<td>---------</td>
<td>---------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Mapping table of the ingress router.

<---- Search Key ---->

<table>
<thead>
<tr>
<th>IRouter</th>
<th>ERouter</th>
<th>CID</th>
<th>RTP/UDP/IP</th>
<th>PT</th>
<th>LTR</th>
<th>LPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Port#</td>
<td>CID</td>
<td>Header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-----</td>
<td>------------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Mapping table of the egress router
2.5 Payloads Arrangement in One IP Packet

Figure 5 shows the format of one IP packet containing several multiplexed RTP packets. The packet is addressed to the edge router B. The RTP/UDP/IP header for streams 1 and 2 have already been communicated to B, that is why they are omitted from the packet. In the case of stream 3, this is either the first packet of the stream or a refreshment packet for the entry in the mapping table at the router B. When the mini-header is read, the de-multiplexer can tell from the Payload Length field in the mini-header that the RTP/UDP/IP header is inserted after the mini-header. The RTP/UDP/IP header or the packet payload is always inserted after the mini-header.

```
<table>
<thead>
<tr>
<th>IP Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr. B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UDP Header</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RTP Header</th>
</tr>
</thead>
</table>

    ------
    | M 1 |
    |-----|
    | Payload 1 |

    ------
    | M 2 |
    |-----|
    | Payload 2 |

    *****| 0 |
    |-----|
    | RTP/UDP/IP |

Packets
For       ---->
Stream 3

    ------
    | M 3 |
    |-----|
    | Payload 3 |

    :
    :
```

Figure 5. IP packets with multiplexed streams.
2.6 Waiting Timer

Since packets arrive at the ingress router at various times, there is a variation in the waiting time between the packets; packets that arrive first undergo a longer waiting time than other packets arriving later but still multiplexed in the same packet. There are also situations when it will be long time before enough packets for multiplexing arrive at the edge router; the worst case can happen when there is only one stream, and packets from this single stream have to aggregate to be multiplexed into a single packet. Because waiting time is crucial for voice applications like VoIP, there should be an upper limit on the waiting time for the packets. We suggest using a timer to control the waiting time at each out-bound queue. The timer is set with the arrival of the first packet into the queue, and the queue is flushed out either when there are enough packets to send a "complete" packet or when the timer expires. The timer should be set to a value that is large enough to accumulate as much packets as possible to make multiplexing pay-off, but also small enough in order to keep the waiting time as small as possible.

2.7 Timestamp and Sequence Number in the RTP Header

The RTP header has two important fields that are used by real-time applications: sequence number and timestamp. The sequence number is used by the application to detect packet loss and to restore the order of the packets. The timestamp is used to remove packet jitter introduced in the network and to provide synchronous playout between numerous sources.

Since the RTP header is replaced with a mini-header at the ingress router, the original information about sequence number and timestamp are not transmitted with the packet all the way to the receiver application. To resolve this issue, we decided to use the system’s time at the ingress router as the timestamp for all the streams while the egress router takes care of regenerating the sequence numbers.

Regenerating the sequence number of the packets is much simpler than the timestamp. Since the first packet and the refresh packet of each stream have the sequence in the RTP header, the egress router can use this value as an initial value for the stream. This value is stored in the mapping table, and for each subsequent packet, the egress router will only increment the sequence number. As for the timestamp, we recommend using the timestamp of the ingress router since it is closer to the source and the loss in the delay value would be minimum. Only the information about the delay occurred in the local access network would be lost. Using RTP between peer routers allows the egress router to extract the timestamp of the ingress router from the RTP header. This timestamp is used for all the mini-packet with in that single RTP packet.
3. Exchange Multiplexing Capability

Before sending multiplexed packets, the ingress router has to determine whether the egress router has multiplexing capability for this proposed scheme or not, and in case it has, what is the port number used to received multiplexed packets. To communicate this information, we propose an easy and simple control signaling protocol. The next section presents the protocol and the type of exchanged messages.

3.1 Simple Exchange Multiplexing Capability Protocol (SEMCP)

The proposed protocol is supposed to provide the router with the ability to determine whether its peer router has multiplexing capability for the proposed scheme, and what port number is used for receiving multiplexed packets. This is done by sending an SEMCP request message to a well known port at the peer router. This port number can be universal or it can be customer tailored.

In case the peer router has multiplexing capability, it will reply to the request message with a response message containing a designated port where the multiplexed packets are supposed to be sent.

If the peer router can not support multiplexing, the message will not be answered. The sender router will have no other choice but to send raw data to its peer router.

A message in the SEMCP protocol has a simple layout; it has three components: variant_and_version, msg_identifier (message identifier) and data. The variant_and_version field indicates the variant from the simple scheme (See section 4) and the version of the used protocol. The 4 leftmost bits are used to identify the variant and 4 rightmost bits are used for the version. The msg_identifier is an 8-bit field used to classify the type of the message, and the data field is a 2-byte field used to convey additional required data. Table 2 gives a list of reserved msg_identifiers and their meaning. The structure of the SEMCP looks as follow:

SEMCP_msg = Struct {
    variant_and_version BYTE;
    msg_identifier BYTE;
    data[2] BYTE;
}
Figures 6 and 7 show two scenarios of control signaling exchange between two routers. In the first case (figure 6) the edge router B does not support the new multiplexing scheme while in the second (figure 7) it does.

Figure 6. Edge router B has no multiplexing capability

Figure 7. Edge router B has multiplexing capability
4. Variant of the Simple Scheme

The proposed scheme suffers from the drawback that the RTP timestamp and sequence number that are received at the receiver side are not the same as the ones that were sent at the sender side. A receiver application that depends on this information might behave in a different way from what is expected. A variant of the proposed scheme would be to expand the mini-header to include the RTP timestamp and sequence number from the original message. This variant incurs some extra space in the mini-header but it achieves complete consistency with the original streams. Figure 8 shows how a mini-header for this variant would look like.

![Figure 8. A variant of the simple mini-header](image)

An ingress router may indicate its preference to a variant in the SEMCP request message. The egress router might agree to use the variant suggested by the ingress router, or it might suggest using another one in case it was not able to support the suggested one. An egress router might also be able to support both variants over different port numbers, depending on the requirements of the applications.

5. Flow Identification

An important issue with the implementation of the proposed scheme is the flow identification. The ingress router should be able to identify all packets that belong to a certain flow, especially flows with small packets. Typically, flow identification is done by recognizing some combination of source IP address and port number, destination IP address and port number, and protocol type. RFC-1953 defines two flow types or classes [Newm96]. Packets in Class-2 flow have the same IP source and destination addresses. Packets in Class-1 flow have also the same UDP or TCP port numbers.
RTP flows or streams carrying IP telephony packets could be easily identified if there was a fixed port for receiving IP telephony traffic. One may also use other simple flow identification algorithms to identify the flows of constant and small packets, which may or may not carry the voice over IP traffic [Lin97].

6. Conclusion

A light-weight data driven multiplexing scheme is proposed. This scheme can be used whenever the payload size is relatively small compared to the header information. The scheme increases the bandwidth efficiency by substituting the header with a mini-header, and merging several packets into a single one. A simple control signaling protocol is also proposed to exchange simple control signals between peer entities. A variant of the here proposed multiplexing scheme could be used in the case that the end-to-end significance of the RTP time-stamp and sequence number information must be conveyed reliably from the source to the sink. In this case, an expanded mini-header could be used which includes, in addition to the information described above, the RTP time-stamp and sequence number of the original packet. This requires, however, 6 more octets per mini-packet.

7. Authors’ Addresses

Gregor v. Bochmann
University of Ottawa
Colonel By Hall
161 Louis Pasteur, Rm. A519
Ottawa, On K1N-6N5, Canada
E-mail: bochmann@site.uottawa.ca
Tel. (613) 562 5800 ext. 6205
Fax. (613) 562-5175

Gang Luo,
Nortel Networks,
PO. Box 3511, Station C, Ottawa ON K1Y 4H7, Canada
E-mail: gluo@nortelnetworks.com
Tel: (613) 765 5969

Khalil M. El-Khatib
University of Ottawa
Colonel By Hall
161 Louis Pasteur, Rm. A519
Ottawa, On K1N-6N5, Canada
E-mail: elkhatib@site.uottawa.ca
Tel. (613) 562 5800 ext. 6385
Fax. (613) 562-5175
8. Bibliography


[Hand98] Mark Handley (ISI), AVT group meeting minutes for Aug 1998 meeting.
