Internet Engineering Task Force   Audio Video Transport WG
Internet Draft   K. El-Khatib, University of Ottawa
Oct 22, 1999   G. Luo, Nortel Networks
Expires: April 22, 2000   G. Bochmann, University of Ottawa
                                       Pinjiang Feng, University of Ottawa

Multiplexing Scheme for RTP Flows between Access Routers

< draft-ietf-avt-multiplexing-rtp-01.txt >

ABSTRACT

This draft proposes a light-weight data driven approach for multiplexing low bit rate RTP streams at the edge router of the Internet in order to reduce the RTP/UDP/IP header overhead associated with each RTP stream. Audio packets from different sources in a local access network destined to different users in the same remote access network are multiplexed into one packet, with the original RTP/UDP/IP header of each packet replaced with a mini-header (2 bytes), resulting in a reduction of the overhead. Access routers can use the IP telephony Border Gateway Protocol (TBGP) to exchange the reachability of IP destinations in their domains.

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 64kbps 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] [Kore99][Hand98b]. 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 (figure 1).

```
                      ________ Gateways________
                     |                         |
                     |                         |
                    \|/     _____________     \|/
                     |     |             |     |
       ___A____            |             |            ___B____
EU_A1-|        |  _______  |             |  _______  |        |--EU_B1
      | PSTN   |\|Gateway|/|  Internet   |\|Gateway|/| PSTN   |
      |        |/|___A___|\|             |/|___B___|\|        |
EU_A2-|________|           |             |           |________|--EU_B2
                           |             |
                           |_____________|
```

Figure 1. Gateways A and B connect two section of the PSTN over the Internet

Most of 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. Section 7 gives a comparison between several drafts for multiplexing RTP flows that were submitted to the IETF AVT working group.

2. 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 2 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 end users EU_A1 and EU_A2 use voice streams to communicate with remote end users EU_B1 and EU_B2. Packets from EU_A1 and EU_A2 could be multiplexed at the router RA and de-multiplexed at router RB and vice versa for packets generated from end users EU_B1 and EU_B2.

```
                      _____Edge Routers _______
                     |                         |
                     |                         |
                    \|/      ______________   \|/
                     |      |              |   |
        ____A____           |              |          ____B____
EU_A1--|         |  ______  |              |  ______  |         |--EU_B1
       | Access  |\|  RA  |/|   Internet   |\|  RB  |/| Access  |
       | Network |/|______|\|              |/|______|\| Network |
EU_A2--|_________|          |              |          |_________|--EU_B2
                            |              |
                            |______________|
```

Figure 2. The Internet with two access networks and two edge routers.

3. RTP Streams Multiplexing Scheme

3.1 Overview of the Scheme

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 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, the timestamp and the sequence number are modified, 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.

3.2 Mini-header Format

Figure 3 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 3. Format of the mini-header with the extension header

3.3 Mapping Tables

Figure 4 and 5 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 lifetime 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.

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.

Abbreviation used	Description
Source IP	Source User IP Address
Source Port#	Source User Port Number
Destination IP	Destination User IP Address
Destination Port#	Destination User Port Number
PT	Payload Type
IRouter IP	Ingress Router IP Address
ERouter IP	Egress Router IP Address
ERouter Port#	Egress Router Port Number
CID	Channel Identifier
LTR	Last_Time_Refreshed
LPR Time.	Last Packet Reproduced Timestamp
LPR Seq#	Last Packet Reproduced Sequence Number

Table 1. Abbreviations used in the mapping tables

```
<---- Search Key ---->
+------------+-------------+----+---------+------------+-----+-----+
| Source     | Destination | PT | IRouter |   ERouter  | CID | LTR |
|------------|-------------|    | Port#   |------------|     |     |
| IP | Port# | IP | Port#  |    |         | IP | Port# |     |     |
|----|-------|----|--------|----|---------|----|-------|-----|-----|
```

```
|----|-------|----|--------|----|---------|----|-------|-----|-----|
|----|-------|----|--------|----|---------|----|-------|-----|-----|
|----|-------|----|--------|----|---------|----|-------|-----|-----|
|----|-------|----|--------|----|---------|----|-------|-----|-----|
+----+-------+----+--------+----+---------+----+-------+-----+-----+

            Figure 4. Mapping table of the ingress router.


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

```
+------------+---------+-----+----------+----+-----+-------------+
| IRouter    | ERouter | CID |RTP/UDP/IP| PT | LTR |     LPR     |
|------------|  Port#  |     | Header   |    |     |-------------|
| IP | Port# |         |     |          |    |     | Time | Seq# |
|----|-------|---------|-----|----------|----|-----|------|------|
|----|-------|---------|-----|----------|----|-----|------|------|
|----|-------|---------|-----|----------|----|-----|------|------|
|----|-------|---------|-----|----------|----|-----|------|------|
|----|-------|---------|-----|----------|----|-----|------|------|
|----|-------|---------|-----|----------|----|-----|------|------|
+----+-------+---------+-----+----------+----+-----+------+------+
```

Figure 5. Mapping table of the egress router

3.4 Payloads Arrangement in One IP Packet

Figure 6 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.

```
 _______________
|  IP Header     |
|  Addr. B       |
|_______________|
|     UDP        |
|     Header     |
|_______________|
|      RTP       |
|     Header     |
|_______________|
|  ___________   |
|  ___________   |
| |    M 1    | |
| |-----------| |
| | Payload 1 | |
| |___________| |
|  ___________   |
| |    M 2    | |
```

```
                           | |-----------| |
                           | | Payload 2 | |
                           | |___________| |
                           |  ___________  |
                           | | *****| 0  | |
                           | |----------| |
                |------->  | | RTP/UDP/IP| |
 Packets        |          | |___________| |
   For    -------+         |  ___________  |
```

```
 Stream 3         |          | |    M 3    | |
                  |------->  | |-----------| |
                             | | Payload 3 | |
                             | |___________| |
                             |       :       |
                             |       :       |
                             |_______________|
```

Figure 6. IP packets with multiplexed streams.

3.5 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 situation 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 application 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 flashed 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.

4. Locating the Address of the Peer Egress Router

When an access router receives a new stream destined to a certain IP address, it has to know the IP address of the egress access router with multiplexing capability, if there is one, that serves the access network where the destination IP resides. Information in the IP routing table can only give the IP address of the next hop toward the destination node, and not the IP address of the egress router. The problem is similar to the problem of finding the IP address of a gateway to complete a call originating from the Internet to the PSTN network. This has been referred to as "Telephony Routing over IP"[Rose] or "gateway location problem" [Squi99].

A framework for Telephony Routing over IP (TRIP) is described in details in [Rose99]. In the framework, Location Servers (LSs) are entities that keep information about gateways (egress routers in our case). LSs from different domains use the Gateway Location Protocol to exchange reachability information of PSTN and IP destinations. The protocol does not have an auto-discovery functionality, and the peer LSs are manually configured.

Two implementations of the TRIP framework, IP Telephony Border Gateway Protocol (TBGP) [Hamp99] and Gateway Location Protocol (GLP) [Squi99], have been presented as drafts to the IETF Audio/Video Transport (avt) working group. The two drafts differ in the base protocol used. While the TBGP is based on the Border Gateway Protocol 4 (BGP-4)[Rekh95], GLP uses a variant of the Server Cache Synchronization Protocol (SCSP)[Luci97] to accomplish database synchronization on different LSs.

To build the database about reachable egress routers that support our multiplexing scheme, we are planning to use the TBGP between peer routers. Our decision is based on the fact that TBGP supports information about reachability of IP addresses. Additional attributes would also include the version and the variant of the multiplexing scheme, and the port number used to receive the multiplexed data.

5. 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.

5.1 A simple Scheme

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 within that single RTP packet.

5.2 A Scheme with Consistency

The simple 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 expend 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.

```
                  _0_1_2_3_4_5_6_7_8_9_0_1_2_3_4_5_
                 |      CID       |X |     PL      |
                  ---------------------------------
                 |         Extension Header        |
                  ---------------------------------
                 |         Sequence number         |
                 |                                 |
                 |---------------------------------|
                 |           Timestamp             |
                 |                                 |
                 |_________________________________|
```

Figure 8. A variant of the simple mini-header

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.

6. 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].

7. Comparison of Different Proposals for RTP Flow Multiplexing

Several drafts were submitted to the IETF AVT working group concerning RTP flow multiplexing. Table 2 summarizes a comparison among these drafts in terms of performance and support issues.

	Our Proposal	Nokia	Bell Labs	TCRTP	Hitachi	GeRM

Header Per Payload	2/4	2	2/4	4~7	12	1~13
max payload size	128/ 65536	64	65536	no limit	no limit	no limit
non-RTP multiplex	yes	yes	no	yes	no	no

mux & demux	simple	simple	simple	simple	simple	hard
max # of user streams	256	256	128	no limit	no limit	no limit
timestamp preserved	optional	no	no	optional	yes	yes
sequence # preserved	optional	no	no	optional	yes	yes
lost packet affect others	possible	possible	possible	possible	no	no
padding header required	no	no	yes	no	no	no
between edge routers	yes	yes	yes	yes	no	no

Table 2. Comparison of Different Proposals

Nokia's proposal suffers from the drawback that the payload size must be smaller than 64 bytes, and it does not mention any additional support to the transmission of time-stamp and sequence number.

Bell Labs' proposal requires that "all multiplexed streams in one packet have the same clock rate". It also requires padding.

For the TCRTP proposal, the minimum size of the header can only be 4 bytes while others' proposal have a minimum header size one(1) [Hand98b] or two (2)[Subb98][Rose99] (our scheme also).

Hitachi's proposal requires a fixed header size (full RTP header (12 bytes), but not the UDP and IP headers), which does not save much on low bandwidth streams.

To achieve high performance using the GeRM multiplexing (header size = **1 byte), all multiplexed streams must have the same RTP header** (timestamp, payload type,...) and the SSRC's differ by one (1). This requires that all sources be synchronized (start, stop, packetization interval) and have the same payload type. This renders GeRM in-applicable when the RTP sources are dispersed.

Both GeRM and Hitachi's proposals are packet loss resilient, where a lost packet can not affect the de-multiplexing of subsequent packets. All other proposals do not have this advantage.

Our proposal provides high performance by using a minimum size Mini-header (2-4 byte) that can support large size payloads (up to 65536 bytes). It can also support the transfer of timestamp and sequence number of the RTP header through different variants of the scheme.

8. 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.

9. 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. 6244
Fax. (613) 562-5175

Pinjiang Feng
University of Ottawa
Colonel By Hall

161 Louis Pasteur, Rm. A519
Ottawa, On K1N-6N5, Canada
E-mail: pfeng@site.uottawa.ca
Tel. (613) 562 5800 ext. 6244
Fax. (613) 562-5175

10. Bibliography

[Tani98] K. Tanigawa, T.Hoshi and K. Tsukada: "Simple RTP multiplexing transfer methods for VoIP.", IETF draft, , work in progress, draft-tanigawa-rtp-multiplex-01.txt, Expired.

[Rose98] J. Rosenberg and H. Schulzrinne: "An RTP payload format for user multiplexing", IETF draft, work in progress, draft-ietf-avt-aggregation-00.txt, Expired.

[Subb98] B.Subbiah, S. Sengodan: "User Multiplexing in RTP payload between IP Telephony Gateways", IETF draft, work in progress, draft-ietf-avt-mux-rtp-00.txt, Expired.

[Hand98a] Mark Handley (ISI), AVT group meeting minutes for Aug 1998 meeting.

[Newm96] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, G. Minshall, " Ipsilon Flow Management Protocol Specification for IPv4", IETF RFC 1953, May 1996.

[Lin97] Steve Lin, Nick McKeown, "A simulation Study of IP Switching", Tech Report CSL-TR-97-720 (Standford Uni.), April, 1997. Available from pubs@shasta.standford.edu

[Rose99] J. Rosenberg, H. Schulzrinne, "A Framework for Telephony Routing over IP", IETF Draft, work in progress, 1999.

[Squi99] M. Squire: "A Gateway Location Protocol", IETF Draft, work in progress, 1999.

[Hamp99] D. Hampton, D. Oran, H. Salama, D. Shah, "The IP Telephony Border Gateway Protocol (TBGP)" ", IETF Draft, work in progress, 1999.

[Luci97] J. Luciani, G. Armitage, J. Halpern, N. Doraswamy, "Server Cache Synchronization Protocol (SCSP)", RFC 2334, April 1997.

[Rekh95] Y. Rekhter and T. Li, A border gateway protocol 4 (BGP-4), Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, Mar. 1995. (Obsoletes RFC1654).

[Kore99] T. Koren, P. Ruddy, B. Thompson, A. Tweedly, D. Wing, "Tunneled multiplexed Compressed RTP ("TCRTP") ", IETF Draft, work in progress, June 1999.

[Hand98b] M. Handley, "GeRM: Generic RTP Multiplexing", IETF Draft, work in progress, draft-ietf-avt-germ-00.txt, Expired.