Protocol overview: RTP and RTCP

Tommi Koistinen
Nokia Telecommunications
Email: Tommi.Koistinen@ntc.nokia.com

Abstract

This paper presents the current status of two internet protocols: Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP). Together these protocols may provide controlled delivery of multimedia traffic over the Internet. The feedback mechanisms for the quality of service (QoS) monitoring, such as delay, jitter and packet loss calculations, are described in detail. The scalability issues with large multicast groups are discussed next. The future development of the RTP protocol with associated extensions for low-speed links and user multiplexing are described lastly.

1 Introduction

Multimedia services, such as video conferencing, Internet telephony and streaming audio, have recently been introduced for the millions of users of the Internet. The popularity of these services and the feedback received has clearly revealed that some modifications and extensions to the current internet protocols are needed to be able to support real-time applications better. Minimization of the end-to-end delay, accurate synchronization of the voice and video streams and a feedback mechanism for the quality of service monitoring are some of the main requirements of these various multimedia applications.

The Transmission Control Protocol (TCP) is the most widely used transport-level protocol in the Internet. However, there are several facts that make TCP quite unsuitable for the real-time traffic. Firstly, TCP includes an in-built retransmission mechanism, which may be useless with strict real-time constraints. Secondly, TCP is a point-to-point protocol without direct support for multicast transmission. Thirdly, there is not any timing information carried, which is needed by most real-time applications. The other widely-used transmission protocol, User Datagram Protocol (UDP), does not either include any timing information. So, a new transport level protocol, called Real Time Transport Protocol (RTP), was specified within the Internet Engineering Task Force (IETF) to cope with the beforementioned problems with the real-time traffic. The IETF's Audio/Video Transport (AVT) working group [1] has since then been the main forum for RTP related discussion and specification work. The International Telecommunications Union (ITU) has also adopted the RTP as the transport protocol for the multimedia. The ITU-T recommendation H.323 [2], and furtherly the recommendation H.225.0 [3] include RTP as the transport protocol of multimedia sessions.

This paper firstly reviews the RTP and RTCP protocols in the chapters 2 and 3 and starts then discussing the various ways to utilize the RTP protocol. These include dynamic QoS control, which is discussed in chapter 4, and the treatment of low-bandwidth links, which is discussed in the chapter 5. The chapter 6 presents the general requirements for a RTP multiplexing scheme. In the concluding chapter also the future plans of the IETF concerning RTP protocol are shortly discussed.

2 Real Time Transport Protocol

RTP [4] is a real-time end-to-end transport protocol. However, considering RTP as a transport protocol may be misleading because it is mostly used upon UDP, which is also considered as a transport protocol. On the other hand, RTP is very closely coupled to the application it carries. So, RTP is best viewed as a framework that applications can use to implement a new single protocol. RTP doesn't guarentee timely delivery of packets, nor does it keep the packets in sequence. RTP gives the responsibility for recovering lost segments and resequencing of the packets for the application layer. There are a couple of benefits in doing so. The application may accept less than perfect delivery and with video or speech there usually is no time for retransmissions. Also the sender may provide, instead of retransmission, new or updated data that tries to fix the consequences of the original loss. What RTP then provides, is:

- Payload type identification
- Source identification
- Sequence numbering
- Timestamping

which are required by most multimedia applications. The accompanying RTP Control Protocol (RTCP) provides feedback of the quality of the data delivery and information about session participants. A RTP session usually is composed of a RTP port number (UDP port), a RTCP port number (consecutive UDP port) and the participant's IP address.

2.1 RTP packet format

The RTP packet format (Table 1) is in detail reviewed in the following.

Table 1: Format of the RTP packet

V	P	X	CC	M	PT	Sequence number
Timestamp						
Synchronization source (SSRC) identifier						
Contributing source (SSRC_1) identifier						
…						
Contributing source (SSRC_n) identifier						
P A Y L O A D						

The first 32 bits of the header consists of several control bits. The version number (V) is currently 2. The padding bit (P) indicates if there is padding octets inserted at the end of this packet. Padding may be required by some applications with fixed length packet sizes. The extension (X) bit indicates if there is an experimental extension after the fixed header. The count field (CC) tells the number of contributing source identifiers (CSRC) following the fixed header. The marker bit (M) may be used as general marker, f.g. indicating the beginning of a speech burst. The payload type (PT) field identifies the payload format, which are discussed in the chapter 2.2. The sequence number is an incrementing counter which is started by a source from a random number. The timestamp corresponds to the generation instant of the first octet in the payload. The synchronization source identifier (SSRC) is a randomly generated value that uniquelly identifies the source within a session. Even if it is very unlikely that two sources generate the same SSRC number, every RTP implementaton should have a mechanism to cope with this chance. Following the fixed header there are one or more contributing source identifiers which are supplied by the mixer (mixers are described in chapter 2.3) and the payload.

2.2 Payload Types

Before RTP may be used for a particular application the payload codes and the actual payload formats should be defined in a profile specification, which may also describe some application specific extensions or modifications to RTP. The RFC 1890 [5] defines a set of standard encodings and their names when used with RTP. These payload types include for example G.721, GSM Full Rate, G.722 and G.728 speech codecs and JPEG and H.261 video codecs. A new revision of this RFC is about to come, which adds some new types including G.723, G.729 and H.263 codecs. Additionally, there are several separate RFCs or drafts for different codecs (f.g. for MPEG1/2/4, JPEG, H.261 and H.263) which define the payload formats and transport policies in more detail. There are also new drafts for payloads of telephone signal events and DTMF tones.

2.3 Mixers and Translators

As RTP is designed to support multicast transmission the RTP packet includes a source identifier (SSRC) which identifies the particular sender from the group. There are, however two special kinds of sources: a mixer and a translator. A mixer combines packets from multiple senders and forwards them to one or more destinations. The mixer assignes itself as the sender of the packet and it also resynchronises the sending (SSRC). The identifiers of all contibuting sources (CSRC) are attached to the combined RTP packet. A translator may change the format of the data in the packet, for example if there is a difference in the allowable transfer rate of the end-points.

3 RTP Control Protocol

The RTP data transport is augmented by a control protocol (RTCP), which provides the RTP session participants feedback on the quality of the data distribution. The underlying protocol must provide multiplexing of the data and control packets, with UDP this is usually implemented using separate port numbers. The format of the RTCP packets is fairly similar to RTP packets, f.g. the type indication is at the same location. The main function of the RTCP are:

- QoS monitoring and congestion control
- Identification
- Session size estimation and scaling

The RTCP packets contain direct information for quality-of-service monitoring. The sender reports (SR) and receiver reports (RR) exchange information on packet losses, delay and delay jitter. This information may be used to implement a TCP like flow control mechanism upon UDP at the application level using adaptive encodings. A network management tool may monitor the network load based on the RTCP packets without receiving the actual data or detect the faulty parts of the network.
The RTCP packets carry also a transport-level identifier (called a canonical name) for a RTP source, which is used to keep track of each participant. Source description packets may also contain other textual information (user's name, email address) about the source. Albeit the source of the RTP packets is already identified by the SSRC identifier, an application may use multiple RTP streams, which can be easily associated with this textual information.
The RTCP packets are sent periodically by each session member in multicast fashion to the other participants. The more there are participants the more RTCP messages should be exchanged. That's why the fraction of the control traffic must be limited. There is in fact a trade-off between up-to-date information and the amount

of the control traffic. The control traffic load is scaled with the data traffic load so that it makes up about 5% of the total data traffic.
There are, however, some weaknesses related to the scalability of the current RTCP algorithms. These problems are listed in below.

- Congestion due to floods of RTCP packets in highly dynamic groups
- Large delays between receipt of RTCP packets from a single user
- Large size of the group membership tables

The first and third problem are studied in detail in chapters 3.5 and 3.6, which describe a timer reconsideration algorithm and sampling of the group membership.

3.1 RTCP packet formats

Each RTCP packet starts with an header similar to that of the RTP data packets. The payload type field identifies the type of the packet. In [5] there are five RTCP payload types (200-204) defined:

- Sender Report (SR)
- Receiver Report (RR)
- Source Description (SDES)
- Goodbye (BYE)
- Application-defined packet (APP)

The contents of these packets are in detail described in the following.

Table 2: Format of the Sender Report

V	P	RC	PT=200	Length
SSRC of the sender				
NTP timestamp (MSB)				
NTP timestamp (LSB)				
RTP timestamp				
Sender's packet count				
Sender's octet count				
First reception report block (SSRC_1)				
…				
Last reception report block (SSRC_n)				

The first 32 bits of the header of the sender report (Table 2) consists of several control bits. The version number (V) and padding field (P) are the same as in RTP packet. The reception report count (RC) indicates the number of receiver reports attached to this packet. The maximum number of receiver reports is 32. The payload type (PT) for sender report is 200. The length field indicates the length of the packet in 32-bit words minus one.
The second 32-bit word includes the SSRC of the sender and the next two words include the high and low parts of the 64-bit NTP (Network Time Protocol) timestamp. The RTP timestamp indicates the relative sending time of this packet. Last sender related words include the sender's packet and octet counts. Following the sender's information block (greyed area in the table 2) there are zero or more reception report blocks, which follow the same format as in the receiver reports.

Table 3: Format of the Receiver Report

V	P	RC	PT=201	Length
SSRC of the sender				
SSRC of the first source				
Fract. lost	Cum. no of packets lost			
Ext. highest sequence number received				
Interarrival jitter estimate				
Last sender report timestamp (LSR)				
Delay since last sender report (DLSR)				
…				
Last reception report block				

The greyed area in the table 3 is considered as one reception report block. The first 32-bit word in that block is the SSRC of the source, for which this reception report is aimed. The fraction lost field indicates the number of packets lost divided by the number of packets expected (according to the highest sequence number received) since last receiver report. The lower part of the next 32-bit word includes the highest sequence number received since last report, whereas the higher part is used as an extension to the sequence number revealing possible resets of the sequence numbering. The contents and use of the interarrival jitter field, Last Sender Report timestamp (LSR) field and Delay since Last Sender Report (DLSR) fields are explained in detail in the subchapters 3.2 and 3.3.

Table 4: Format of the Source Description

V	P	SC	PT=202	Length
SSRC/CSRC of the sender				
Type	length	text		
text continued				
…				
Last chunk				

The Source Description (SDES) packet is a three-level structure composed of a header and zero or more chunks (greyed area in the table 4), which describe the source identified in that particular chunk. An end system sends only one chunk with its SSRC but a mixer incorporates as many chunks as there are contributing sources to be identified. Each SDES item starts with an 8-bit type field followed by an 8-bit octet count, which identifies the length of the following text field. The defined SDES items are: canonical end-point identifier (CNAME), which should follow the format *user@host*, user name (NAME), being the real user name, electronic mail address (EMAIL) in format John.Doe@megacorp.com, phone number (PHONE), geographical user location (LOC), application or tool name (TOOL), notice (NOTE) and private extensions (PRIV). Only the item CNAME is mandatory.

Table 5: Format of the BYE packet

V	P	SC	PT=203	Length
SSRC/CSRC of the sender				
length			reason for leaving	
…				
Last chunk				

The BYE packet indicates the receivers that a source is leaving the session and the prolonged silence will be caused by that reason instead of a network failure. The BYE packet may optionally include a textual description of the reason for leaving.

Table 6: Format of the application defined packet

V	P	Sub	PT=204	Length
SSRC/CSRC of the sender				
name (ASCII)				
application-dependent data				

The application defined packet is intended for experimental use without requiring packet type value registration. The SUB field may be used to implement two-level type hierarchy if needed. The ASCII-based NAME field should uniquelly define the application among other applications which may be received. The last field is for application-dependent data.

3.2 Round-trip delay

Receiver reports may be used to estimate the round-trip delay between sender and receiver. The receiver report includes the LSR (timestamp from the last sender report received) and DLSR (delay since last sender report received) fields, from which the sender can directly calculate the round-trip delay according the formula 1, where A is the time instant when the receiver report was received by the sender.

$$D = A\text{-}LSR\text{-}DLSR \quad (1)$$

The figure 1 shows the round-trip calculation against the time axis. The middle 32-bits of NTP timestamp are copied by the receiver to LSR field and the delay since last particular sender's report is stored until a corresponding receiver report is sent. It should be noted that as the minimum interval between consecutive reception reports is defined to be 5s, the delay estimate can not be used as a real-time measure.

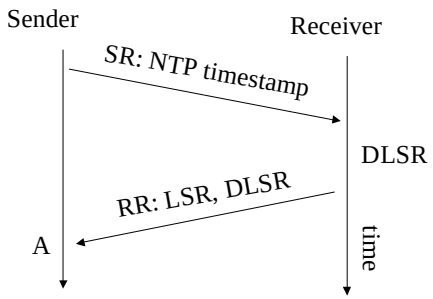

Figure 1. Calculation of round-trip delay.

3.3 Inter-arrival jitter

The receivers observe continously the variance of the inter-arrival time of incoming RTP packets. An estimate for inter-arrival jitter is calculated as follows. Firstly, the difference D in packet spacing at the receiver compaired to the packet spacing at the sender is calculated according to the formula 2,

$$D = (R_j\text{-}R_i)\text{-}(S_j\text{-}S_i) \quad (2)$$

where R is the time of arrival and S is the RTP timestamp for a certain packet. This delay variation value is calculated after every RTP packet. To avoid temporary fluctuations the final value for inter-arrival jitter estimate is smoothed according to equation 3,

$$J_i = (15/16)J_{i-1}+(1/16)D \quad (3)$$

which gives only a small weight to the most recent observation. It is proposed that the change in this jitter estimate could indicate congestion before it leads to packet loss.

3.4 Packet loss

The receiver reports also contain information about the lost packets. The fraction of lost packets is defined to be the number of packets lost divided by the number of packets expected, which are calculated based on actually received packets and the highest sequence number received in RTP packets. A cumulative number of packets lost is also maintained. These packet loss measures may be used as congestion indication for the sender to reduce the application's sending rate. This kind of feedback system is discussed in chapter 4.

3.5 Timer Reconsideration

As mentioned previously, the current RTCP algorithm of scaling the transmission interval of the RTCP reports is linearly proportional to the group size estimate (L). As the group size grows, sender and receiver reports are send less frequently. This algorithm works fine for group sizes up to several hundreds but when scaled to a very

large and very dynamic multicast group certain problems may arise. It can be observed that in large multicast groups, in cabel TV networks for example, a great number of users change channels at almost the same time when shows begin and end. This "step-join" phenomenon is not handled very efficiently with the current RTCP algorithm. The unrestricted flood of RTCP packets in case of large step-join is very likely to cause congestion, which even makes the situation worse because disappeared packets keep the group size estimates inaccurate. In these situations the 5% target for control traffic is most likely exceeded. In the reference [6] a timer reconsideration method is proposed, which should restrict the number of packets sent especially in rapid step-join environments.
The current RTCP algorithm for transmission interval is based on the following formula (4),

$$t_n = t_{n-1} + R(\alpha) \max(T_{min} , CL(t_{n-1})) \qquad (4)$$

where t_n is the current sending time, t_{n-1} is the previous sending time, $R(\alpha)$ is a randomizing factor between 0.5 and 1.5, T_{min} is initially 2.5s and 5s after that, C is a priori calculated interval according to 5% target for the control bandwidth and $L(t_{n-1})$ is the previous group size estimate. In practise, at time t_{n-1} a timer is set to be run out at time t_n for sending the next packet. The reconsideration algorithm changes this sceme so that when timer has run out the sending time is recalculated using the most recent information about the current group size. The group size estimate $L(t_n)$ may have already changed rapidly from t_{n-1} to t_n in case of a large step-join. If the recalculated sending time is beyond the initial t_n, the packet is rescheduled to be sent later. Otherwise it is sent according to the initial plan.
Two operation modes for reconsideration algorithm are proposed: conditional and unconditional. With conditional mode the reconsideration is done only if group size estimate has changed. With unconditional mode the reconsideration is always done, which makes the reconsideration to act more rapidly when the group size changes beacause incoming reports are not waited. Also the randomisation smoothes the beginning of the group size increase. The figure 2 presents a simulation results of reconsideration algorithm seen by a single user when 10000 new participants join the session. The step-join causes a burst of 10000 packets which are sent in current algorithm to be reduced to 197 packets with conditional and to 75 packets with unconditional reconsideration. These values are far more close to 5 % target of RTCP traffic than that of all sending initally at full speed.

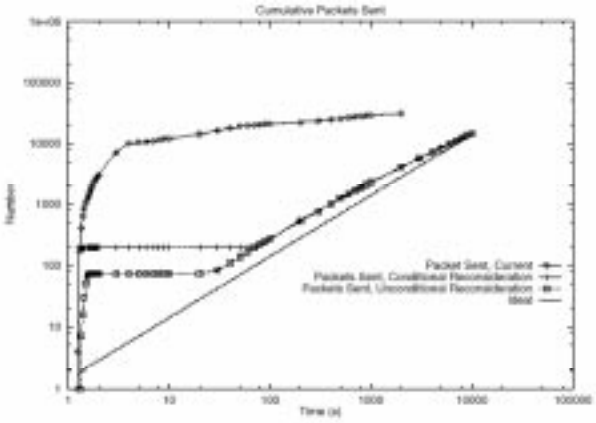

Figure 2. Effect of reconsideration algorithm.

It should be noted that the reverse "join-out" situation may as well cause problems. This problem is, however, not studied extensively so far.

3.6 Sampling of the group membership

The requirement to keep track of all SSRCs of the active members in a session may become a problem with very large multicast groups, where the number of participants may easily grow beyond thousands. Storage of an SSRC table with one million members, for example, requires at least four megabytes, which may be too much for embedded devices with limited memory capacity. In reference [7] is presented a sampling method of group membership, which reduces the need for storage space significantly.
Each participant should maintain a key (K) and a mask (M), both 32-bit wide. The mask has (m) bits as ones and the rest of bits as zeroes. When a RTCP packet arrives with a new SSRC (S) label, this new SSRC is ANDed with the mask and compaired to the ANDed value of the key and the mask. This sampling decision is presented in the equation 5 in mathematical form.

$$D = (K*M == S*M) \qquad (5)$$

The effect of this sampling method is to reject one new SSRC out of 2^m and thus reducing the required storage capacity. The current group size L is estimated at any moment by multiplying the number of storage elements in SSRC table by 2^m.

4 Dynamic QoS Control

The feedback provided by the RTCP reports may be used to implement a flow control mechanism at the application level. In the reference [8] is described an experiment with an video application, the sending rate of which is adjusted according to the packet loss indication from the receiver reception reports. The software video codecs support this kind of adjustment well because the

sending rate can be reduced easily with trade-offs in spatial resolution and quantization. Most voice codecs use fixed length frame sizes so the sending rate is easily changed only by changing the codec on the fly. The smooth switching , preferably on a frame basis, on the quality-rate scale is the main challenge when designing a variable rate speech codec. The experiment in [8] is arranged according to the figure 3.

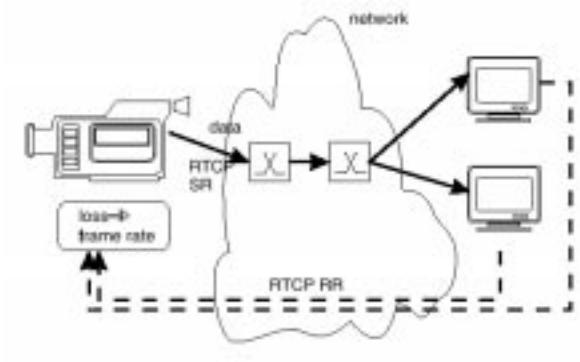

Figure 3. The end-to-end application control

On receiving a RTCP receiver report, the sender analyses the packet loss and delay measures and classifies each receiver either as unloaded, loaded or congested. After that the sender's bandwidth is adjusted. The adjustment is done either based on the receiver with the highest average loss rate or based on certain proportions of unloaded, loaded and congested receivers. The bad thing in the former approach is that a receiver with a low speed link may provide low quality also to all the other receivers. On the other hand, the latter approach will let some amount of congested receivers to suffer continously. The bandwidth is adjusted using a multiplicative decrease but only an additive increase to be able to react rapidly to congestion, still being beware of too rapid increase after the congestion. In the figure 4 is depicted an experiment with the Internet where the threshold of decreasement was set to 10 % of smoothed loss. It can be seen that the sender's bandwidth is reduced when this 10 % threshold is exceeded.

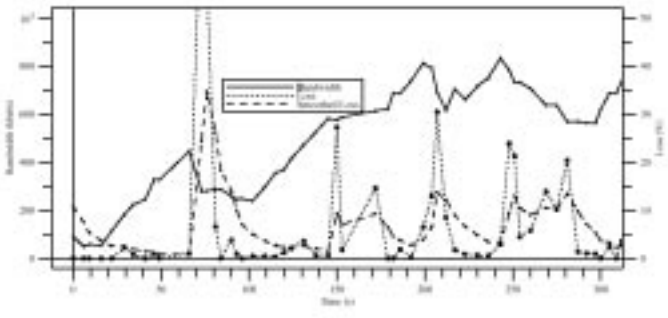

Figure 4. Results of an flow control experiment.

Also the behaviour of the jitter was studied but there were no significant changes in jitter as expected before the losses occurred.

5 Extensions to RTP to support low-bandwidth links

Low-bandwidth links have some special concerns for RTP or RTCP utilization. When a data rate is computed for a multicast session the maximum bandwidth of the low-speed links should be included in that calculation or those low-speed links should be treated specially in some other way.

5.1 Header Compression

The 12-byte RTP header together with 20-byte TCP and 20-byte IP-header produces a quite high overhead to the payloads. This overhead becomes a major problem in low-bandwidth links such as dial-up modems at 14.4 or 28.8 kbps. The brand-new RFC 2508 [9] presents a compression method which reduces the RTP/TCP/IP-header to only two bytes for the most packets. The main idea is that half of the bytes of the TCP and IP headers remain constant over the life of the connection. After sending the uncompressed header once, these fields may be dropped off from the compressed headers that follow. From the RTP headers it can be seen that although several fields change in every packet, i.e. sequence number and timestamp, the difference from packet to packet is often constant and of good use in compressor and decompressor. It is stated that there is no use of compressing RTCP packets, which constitute only 5% of the bandwidth. Also additional memory for saving the context of SDES items should be needed.

5.2 Mobile Networks

In order to provide multimedia services to mobile users, the RTP/RTCP protocol suite should be assessed keeping the limitations of the mobile environment in mind. The most severe limitation is the low bandwidth. It is suggested that a translator should locate at the border of the fixed and the mobile network and change the data format to appropriate format for the mobile link. Another problem is the control traffic, which may have been scaled according to the high-bandwidth side taking still a too big share of the capacity of the mobile link. In reference [10] is proposed an architecture, which consists of a supervisor host (SH), at the border of the fixed and mobile network, and of mobile hosts (MH). The high error rates and the frequent disconnections of the radio interface should be isolated inside the mobile subnetwork. The supervisor host performs two operations for the data traffic: recodes the data to a lower rate or in worst case discards intelligently some of the

data away. For the control data supervisor host buffers and combines the information of the RTCP reports from the fixed network's side and adjusts the control traffic load of the mobile subnetwork to an appropriate level.

6 RTP and User Multiplexing

User multiplexing has become a hot topic in IETF mainly because Voice over IP (VoIP) industry has seen that certain benefits could be gained by multiplexing RTP sessions in gateway-to-gateway links. Without multiplexing each user could have a separate RTP session, which is not very efficient because of the header overhead and companied RTCP traffic. The header overhead is emphasized because the payloads carried in each packet are generally very small, f.g. 10 octets with G.729 speech codec. By multiplexing the header overhead can be reduced. This may also reduce the packetization delay because the header overhead is no more a concern. Yet another benefit may be the reduction of interrupts in gateways which is a consequence of reduced amount of packets received. Less packets is also better for the intermediate routers so multiplexing lowers the chance of congestion.
Some general requirements for the multiplexing protocol are listed in reference [11]:

- Data from different users should be clearly delineated
- The protocol should support variable length blocks from each user
- The channel to which the data belongs must be identified
- The protocol must produce low overhead
- The payload type of each user should be identified

There are multiple IETF drafts [12,13] proposing quite similar multiplexing schemes. Every proposal includes some kind of user-based miniheader, which is attached to user payloads. The IETF plans to analyse and simulate the different multiplexing proposals during the year 1999.

7 Conclusions

The RTP protocol seems to suite the delivery of the real-time traffic pretty well. The RTP protocol provides timing information and the identification of the source and the payload type for the multimedia applications. The accompanying control protocol, RTCP, provides information about the perceived quality of service. However, there are some limitations in the scalability of the RTP sceme. Header overhead may become a problem on low-speed links or on large trunk lines. Thus, a header compression scheme and an user multiplexing scheme are presented. Also the amount of the control traffic may need to be limited. Large multicast groups may utilize timer reconsideration and enhanced group membership sampling to avoid congestion and memory problems.
The current goals of the IETF's Audio/Video working group are to revise the main RTP specifications, to complete the RTP MIB (Management Information Base) and to produce a guidelines document for future developers of payload formats. The different user multiplexing options will be studied in the near future. The discussion on payload formats such as MPEG-4, DTMF and PureVoice and on forward error correction (FEC) techniques shall be continued during the year 1999.

References

[1] Audio/Video Transport (AVT) Working Group, IETF. <http://www.ietf.org/html.charters/avt-charter.html >

[2] ITU-T Recommendation H.323 : Packet-Based Multimedia Communications Systems. February 1998.

[3] ITU-T Recommendation H.225.0 : Media Stream Packetization and Synchronization on Non-Guaranteed Quality of Service LANs. November 1996.

[4] RTP : A Transport Protocol for Real-Time Applications. IETF RFC 1889. January 1996.

[5] RTP Profile for Audio and Video Conferences with Minimal Control. IETF RFC 1890. January 1996.

[6] Rosenberg, J. Schultzrinne, H.Timer Reconsideration for Enhanced RTP Scalability. IETF Draft. July 1997.

[7] Schultzrinne, H. Rosenberg, J. Sampling of the Group Membership in RTP. IETF Draft. November 1998.

[8] Busse, I. Deffner, B. Schultzrinne, H. Dynamic QoS Control of Multimedia Applications based on RTP. Computer Communications. January 1996.

[9] Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. IETF RFC 2508. February 1999.

[10] Brown, K. Singh, S. Extensions to RTP to Support Mobile Networking. Third International Workshop on Mobile Multimedia Communications. September 1996.

[11] Rosenberg, J. Schultzrinne, H. Issues and Options for RTP Multiplexing. IETF Draft. October 1998.

[12] Rosenberg, J. Schultzrinne, H. An RTP Payload Format for User Multiplexing. IETF Draft. May 1998.

[13] Subbiah, B. Sengodan, S. User Multiplexing in RTP payload between IP Telephony Gateways. IETF Draft. August 1998.