PROTOCOLS FOR REAL-TIME MULTIMEDIA DATA TRANSMISSION OVER THE INTERNET

M. Reha Civanlar

AT&T Labs - Research
Newman Springs Lab
100 Schultz Drive, Red Bank, NJ 07701

ABSTRACT

The explosive growth of the Internet and the intranets have attracted a great deal of attention to the implementation and performance of networked multimedia services, which involve the transport of real-time multimedia data streams over non-guaranteed quality of service (QoS) networks based on the Internet Protocol (IP). In this paper, I present an overview of the existing architectural elements supporting real-time data transmission over the Internet. Effective implementations of such systems require a thorough understanding of both the network protocols and the coding systems used for compressing the signals to be transmitted in real-time. The paper includes a section discussing the issues to be considered in designing signal compression applications suitable for network use.

1. INTRODUCTION

Continuing advances in computing technology together with developments in signal coding and network protocols have made transmission of real-time multimedia data over the Internet and intranets a viable and important application. An understanding of the Internet multimedia data transmission architecture is beneficial for developing signal processing applications suitable for this fast growth area. Furthermore, effective design and use of the intermediate protocol layers of this architecture requires in-depth knowledge on both signal processing and networking.

Based on their functionalities, the protocols directly related to real-time multimedia data transmission over the Internet can be classified in four categories:

1. Signaling
2. Session Control
3. Transport
4. Network infrastructure

In this paper, I present a very short overview of the higher layer (signalling and session control) and the lower layer (network infrastructure) protocols, and discuss the transport layer, which is most related to the payload specifics, in more detail.

2. LOWER LAYERS

Currently, all real-time multimedia data transmission applications over the Internet depend on one or both of the fundamental Internet transport protocols, User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) [1], for several functions such as multiplexing, error control, flow control, etc. In turn, TCP and UDP depend on the basic Internet Protocol (IP) for the network services support including network addressing.

Different applications running on a machine with a single network address can be accessed through the multiplexing support. As a part of the error control, the checksum service protects the higher layers from receiving corrupt packets. For example, the UDP layer blocks packets with bit errors in most implementations. So, even a single bit error may result in a lost packet; however, without this service, it is the application's responsibility to deal with bit errors. Flow control provided by TCP targets optimum use of the shared network resources.

The Point-to-Point Protocol (PPP) [2], which defines a standardized method for sending datagrams over communication links such as telephone and ISDN lines, is an integral part of several real-time data transmission applications e.g., the Internet telephony. Several other protocols addressing specific requirements of real-time data delivery are on their path to becoming standards. The Resource Reservation Protocol (RSVP) [3] for defining and implementing QoS requirements, and a family of protocols defining integrated services over specialized links covering a wide range of networking technologies including Ethernet, ATM, etc., are all important for multimedia data delivery over the Internet.

The lower layer protocols have a fundamental impact on the performance and usability of signal coding techniques in a networked application. For example, if the network offers some service guarantees, such as delay bounds or guaranteed packet deliveries (no loss), signal coding techniques with no error resilience can be used. If appropriate data flow control is done at the lower layers, application designers need not worry about network buffer overflows due to short term high output data rates as in the case for I frames in MPEG video. In many cases, such additional services offered by the lower layers are not free, and a price-performance compromise may be obtained by using layered coding techniques. In this case, specialized services are needed only for transmitting a portion of the encoded data streams [4].

3. HIGHER LAYERS

There are several protocols to be used for the higher layer functions of signaling and session control. Signaling includes sending announcements about a multimedia session to prospective participants or inviting selected participants to join a session. In both cases, the details of the session including, e.g., the types of compression techniques used for audio and video signals, the number of audio channels, etc., may be a part of the signaling message. Generating and handling the responses of the receiver to a signaling message, e.g. accept join, reject, busy, forward etc. are handled by signaling protocols also. Additionally, the ability of a receiver to decode the selected

payload types and possible negotiations of the capabilities (capability exchange) may be covered by signaling.

Current protocols supporting signaling for multimedia sessions on the Internet include Session Description Protocol (SDP) [5] for describing multimedia sessions, Session Announcement Protocol (SAP) [6] for announcing the described sessions and, Session Initiation Protocol (SIP) [7] for inviting users (human or machine) to participate in multimedia sessions. The Hyper Text Transfer Protocol (HTTP) and Uniform Resource Locators (URL's) can be used to announce and describe sessions in a "bulletin board" format, which may also be considered as a part of a special type of signaling.

Session control defines the messages and procedures to control the delivery of the multimedia data during an established session. The Real-Time Streaming Protocol (RTSP) [8] addresses tasks such as providing a means for choosing delivery channels and mechanisms, selecting a multimedia data segment for playback, and controlling playback or recording properties using controls similar to the familiar ones on video cassette recorders.

The H.323 standard defined by ITU-T standardizes both signaling and session control for tightly coupled multimedia communications sessions [9]. A discussion of the relation between H.323 and other Internet protocols can be found in [10].

4. TRANSPORT

Being at the intermediate level of the real-time data transmission architecture, the transport protocol has very tight relationships with the way the multimedia payload types are organized and used. I will discuss the details of the transport layer based on the Real-Time Transport Protocol (RTP) [11, 12] in the following sections.

4.1. RTP

The RTP is designed to deliver various kinds of real-time data over packet networks. It addresses the needs of real-time data transmission only and relies on other well established network protocols for other communications services such as routing, multiplexing and timing. This way, we don't need to re-define these services, which are proven to be satisfactory, in general, for each payload type.

RTP typically runs on top of UDP to make use of its multiplexing and checksum services. This is in addition to the basic networking services provided by the underlying IP layer. However, RTP may also be used with other suitable underlying network or transport protocols, e.g. Asynchronous Transfer Mode (ATM) networks. Also, RTP supports data transfer to multiple destinations using multicast distribution if this functionality is supported by the underlying network as in TCP/IP networks.

RTP is based on the Application Level Framing (ALF) and Integrated Layer Processing (ILP) [13] principles, which dictate using the properties of the payload in designing a data transmission system as much as possible. For example, if we know that the payload is MPEG encoded video, we should design our packetization scheme based on "slices" because they are the smallest independently decodable data units for MPEG video. This approach provides a much more suitable framework for MPEG transmission over networks with high packet loss rates. Also, we can identify and protect the critical information by e.g. repeating it frequently or sending it over a reliable channel. In the MPEG video example, payload format types based on both of these approaches have been defined [14, 15].

The services provided by the RTP are described in the following subsections:

Payload type identification: The type of the payload contained in an RTP packet is indicated by an integer in a special field at the packet header. The receiver interprets the content of the packet based on this number. Certain common payload types have assigned payload type numbers [12]. For other payloads, this association can be defined externally, e.g. through signaling during the starting of a session or with session control protocols. The payload type identification service of the RTP together with the multiplexing services supported by the underlying transport protocol, such as UDP, provides the necessary infrastructure to multiplex a large variety of information effectively. Multicast transmission of several multimedia streams multiplexed together with any other type of information can easily be handled using these services.

RTP allows additional information to be added to its generic headers for each payload type. This information may be used to increase the packet loss resiliency of the transmission. For example, each RTP packet carrying MPEG video contains information about the picture type (intra, predictive, bidirectional), motion vector ranges, etc. copied from the latest picture header, increasing the decodability of individual packets [14].

Packet sequence numbering: Each RTP packet that belongs to a stream contains a 16 bit sequence number field which is incremented by one for each packet sent. The sequence numbers make packet loss detection possible because the lower protocol layers need not provide this information. Also, packets received out of order can be re-ordered using the sequence numbers. The initial sequence number is selected as a random number so that RTP packets do not cause known-plaintext attacks on the encryption that may be used at some later stage of their transmission.

Since the packets may be delivered out-of-order, receipt of a packet with an out-of-order sequence number does not necessarily imply packet loss. In most applications, a certain number of packets are buffered before starting the playback so that late or out-of-order packets can be used when they arrive. The buffer size depends on the network jitter, and, for interactive real-time applications, the buffer size is limited by the allowed delay.

Time Stamping: Each **RTP** packet carries a 32 bit timestamp which reflects the sampling instant of the first byte in the payload portion of the packet. The interpretation and use of the timestamp is payload dependent. For example, for MPEG elementary stream payloads, the timestamp represents the presentation time of the MPEG picture or audio frame, a portion

of which is carried by the packet, based on a 90 KHz clock. It is the same for all packets that make up a picture or audio frame and, in a video stream with B frames, it is not monotonically increasing. On the other hand, for fixed-rate audio (e.g. PCM), the timestamp may reflect the sampling period. If blocks covering n audio samples are read from an input device, the timestamp would be increased by n for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

The time stamp together with the information provided by the associated (RTP Control Protocol) RTCP packets, is to be used for:

1. encoder / decoder clock matching
2. synchronization of several sources
3. measuring packet arrival jitter

as discussed in the next section.

Similar to the initial value of the sequence numbers, the initial value of the timestamp is random to make known-plaintext attacks on encryption difficult.

Source identification: The source of each RTP packet is identified by an integer called "Synchronization SouRCe identifier (SSRC)" included in the packet header. Each sender initially picks a random number for its SSRC. It is the senders' responsibility to detect and resolve collisions where more than one source picks the same number in the same session. The relation between several sources participating in a session as well as their characterizing names are established through RTCP as described in the next section.

4.2. Delivery Monitoring - RTCP

The delivery monitoring function of the RTP is carried out using the associated protocol, RTCP. RTCP is based on periodic transmission of control packets from all participants of a session to all other participants using the same distribution mechanism as the RTP data packets. RTCP's main functions are discussed in the following sections:

Feedback on the quality of distribution and timing: In an RTP session, each sender and each receiver send periodic reports to each session participant. Part of this report contains information on the quality of reception characterized as the:

1. fraction of the lost RTP packets since the last report
2. cumulative number of packets lost since the beginning of reception
3. packet interarrival jitter
4. delay since receiving the last sender's report

Sender and receiver reports contain enough information to determine these quantities at each participant's location. This feedback in reception quality is an integral part of the RTP protocol and it is intended to be used for congestion and flow control purposes as well as network performance input for the adaptive coding applications. Since RTP does not define an explicit flow control mechanism, an RTP application is capable of generating high traffic rates causing network congestion. It is important to prevent this by analyzing the RTCP packets coming from the receivers so that other network applications are not disturbed.

Sending the feedback reports to all participants makes it possible to determine the extent of network problems. Additionally, a network management entity may monitor the network performance by observing these reports without actively participating in each session.

As for the timing, each sender's periodic RTCP packets contain 64 bit Network Time Protocol (NTP) [16] timestamps, indicating the wallclock (absolute) time when the RTCP packet was sent. This information can be used in combination with the timing information returned in reception reports from other receivers to measure round-trip propagation to those receivers. Additionally, the sender's RTCP packet contains an RTP timestamp that corresponds to the same time as the NTP timestamp (above), but in the same units and with the same random offset as the RTP timestamps of the RTP data packets. This correspondence is to be used for intra- and inter-media synchronization for sources with synchronized NTP timestamps. A detailed discussion of the clock synchronization procedures can be found in [16].

Participant identification: Special RTCP messages are used to establish a connection between the real identification of an RTP source, called its canonical name (CNAME), and the current SSRC numbers used by it. CNAME's are very similar to e-mail addresses following the "user name"@host syntax. Also, identification messages carry additional information about the participants such as their names, e-mail addresses, phone numbers, etc.

Scale the control packet transmission with the number of participants: As the number of the session participants increases, unregulated RTCP message traffic may consume significant bandwidth. In order to prevent this, RTCP scales itself by changing its message transmission interval based on the number of session participants. The suggested RTCP bandwidth is less than 5% of the bandwidth allocated for a session. Algorithms to achieve this are discussed in [11].

Minimal session control information: This optional functionality can be used for conveying simple session information, e.g. names of the participants, to everyone.

5. MULTIMEDIA DATA STREAM PROPERTIES FOR NETWORK USE

Although it is possible to deliver real-time data encoded in any form over the Internet, real-time multimedia streams with the following properties are more convenient for networked applications:

Natural breakpoints for packetization: Packetizing a stream that has natural breakpoints can be easier and more efficient. As an example, if a picture is JPEG coded and is presented to a packetizer, the resulting packets contain arbitrary sections of the encoded data. If one of these packets is lost, it will be practically impossible to decode the remaining packets even if they are received. However, if the same JPEG coded picture contains

special "restart markers" indicating starting of independently decodable blocks, a lost packet won't cause such a problem.

Adjustable packet sizes: Different technologies used as parts of the Internet have different frame (largest data unit) sizes. In order to carry a packet whose size is larger than the smallest frame size allowed on its path, called Maximum Transmission Unit (MTU), the packet needs to be fragmented and re-assembled. If the size of a packet can be changed based on the MTU, fragmentation can be avoided.

Well defined high priority information: If certain parts of a data stream are vital for decoding the rest of it, it is preferable to have them in easily identifiable and separable sections so that they can be transmitted more reliably.

Flexible rate control: An encoding scheme whose rate can easily be changed is useful in adapting its output to network conditions.

Ease of transcoding: The heterogeneity of bandwidths used for the Internet access requires using different rates for the same multimedia material. Data streams that can easily be transcoded to change their bandwidth are definitely preferable.

Layered coding: Layered coding is beneficial for two purposes. The first one is to remove the need for transcoding by providing representations of the same multimedia source at different bitrates without noticeably increasing the overall bandwidth. The second benefit is to obtain a price/performance compromise by sending only a portion of a stream through channels with special provisions [17, 18] as discussed in section 2.

Resilience to error propagation: Assuming that the packet losses will be unavoidable in the foreseeable future, techniques which prevent or reduce the propagation of data loss effects are preferable [19].

6. DISCUSSIONS

A complete set of standard protocols supporting real-time multimedia stream delivery over the Internet will be available in the very near future. An understanding of these protocols is beneficial for the signal processing community in designing new techniques for network use. Also, in depth knowledge of the underlying signal processing methods are often needed for effective use of the existing protocols and extending them for future applications.

7. REFERENCES

[1] D. E. Comer, "Internetworking with TCP/IP," ISBN 0-13-468505-9, Prentice-Hall, NJ, 1991.

[2] W. Simpson, Editor, "The Point-to-Point Protocol (PPP), " IETF RFC 1661*, July.1994.

[3] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," IETF RFC 2205, September 1997.

[4] S. McCanne and V. Jacobson, "Receiver-driven Latered Multicast," ACM SIGCOMM'96, Stanford CA.

[5] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF MMUSIC Group, Internet Draft**, draft-ietf-mmusic-sdp-04.txt.

[6] M.Handley "SAP: Session Announcement Protocol." IETF MMUSIC Group, Internet Draft, draft-ietf-mmusic-sap-00.txt.

[7] E. Schooler, H. Schulzrinne, M. Handley, "SIP: Session Initiation Protocol," IETF MMUSIC Group, Internet Draft, draft-ietf-mmusic-sip-04.txt.

[8] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP), IETF MMUSIC Group, Internet Draft, draft-ietf-mmusic-rtsp-05.txt.

[9] ITU-T, Recommendation H.323 -- Multi-Media Conferences for Packet-based Network Environments.

[10] M. Handley, J. Crowcroft, C. Bormann and J. Ott, "The Internet Multimedia Conferencing Architecture," IETF MMUSIC Group, Internet Draft, draft-ietf-mmusic-confarch-00.txt.

[11] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[12] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 1890, January 1996.

[13] Clark, D., and Tennenhouse, D. "Architecture Considerations for a New Generation of Protocols," Proc. of ACM SIGCOM '90, Sept. 90,pp.201-208.

[14] D. Hoffman, G. Fernando, V. Goyal and M. R. Civanlar, "RTP Payload Format for MPEG1 / MPEG2 Video," IETF AVT Group, Internet Draft, draft-ietf-avt-mpeg-new-02.txt, November 1997.

[15] M. Reha Civanlar, Glenn L. Cash, Barry G. Haskell, "RTP Payload Format for Bundled MPEG," draft-civanlar-bmpeg-01, Internet Draft, November 1997.

[16] Mills, D., "Network Time Protocol Version 3," IETF RFC 1305, March 1992.

[17] R. Aravind, M. R. Civanlar, Amy R. Reibman, "Packet Loss Resilience of MPEG-2 Scalable Video Coding Algorithms," IEEE Transactions on Circuits and Systems for Video Technology, October 1996.

[18] M. F. Speer, S. McCanne, "RTP Usage with Layered Multimedia Streams," Internet Draft, draft-speer-avt-layered-video-02, December 1996.

[19] S. Wenger, "video Redundancy Coding in H.263+," Proceedings of AVSPN'97, Aberdeen, 1997.

* Internet Engineering Task Force (IETF) Request For Comments (RFC) are available on-line at several web sites, including:

http://ds.internic.net/ds/dspg1intdoc.html and

http://info.internet.isi.edu/1/in-notes/rfc

** IETF Internet Drafts are available on-line at several web sites, including:

http://www.ietf.org/1id-abstracts.html