Chapter 6

The Voice-based Communication Case Study

(.... omissis....)

6.2 Packetized Audio over the Internet

Since the early experiments with packetized voice in the Arpanet network [4], packetized audio applications have become sophisticated tools that many Internet users try to use with regularity. For example, the audio conversations of many international conferences and workshops are now usually conducted over the Mbone (the multicast backbone), an experimental overlay network of the Internet [97]. The audio tools that are used to transmit packet audio over the Internet (e.g. NeVot [95], vat [96], rat [93], the INRIA audio tool [73]) typically operate by periodically sampling audio streams generated at the sending host, packetizing them, and transmitting the obtained packets to the receiving site by using datagram based connections (e.g. UDP). In addition, at the receiving site, packets are buffered and their playout time is delayed in order to compensate for variable network delays that may be frequently experienced.

A number of problems have been identified which negatively impacts the quality of audio conversations, but probably the more critical one with audio is the loss of audio packets. Basically, two are the main causes for audio packet loss over wide-area packet-switched networks: 1) traffic congestion at the interconnecting routers that cause audio packets to be discarded, and 2) too large transmission delays that cause audio packets to arrive at the destination past the time instant at which they are scheduled to be played out (the playout point). With the term playout delay we refer to the total amount of time that is experienced by the audio packets of a given talkspurt from the time instant they are generated at the source and the time instant they are played out at the destination. Summarizing, such a playout delay consists of: i) the ``collection" time needed for the transmitter to collect audio samples and to prepare them for transmission, ii) the ``transmission" time needed for the transmission of audio packets from the source to the destination over the underlying transport network, and finally iii) the ``buffering" time, that is the amount of time that a packet spends queued in the destination buffer before it is played out. A crucial tradeoff exists between audio packet playout delay and audio packet loss: the longer the scheduled playout delay, the more likely it is that an audio packet will arrive at the destination before its scheduled playout deadline has expired. However, if on one side a too large percentage of audio packet loss (over 5-10%) may impair the intelligibility of an audio transmission, on the other side, too large playout delays (e.g. more than 200-250 msec) may disrupt the interactivity of an audio conversation [94].

The described playout mechanisms try to adaptively adjust the playout delay in order to keep this delay as small as possible while minimizing the number of packets that arrive too late (i.e. after their playout point). The next section provides additional information that constitute the background of the algorithm to be presented in this paper. In particular, the main characteristics of the mechanisms that are used to adaptively adjust the playout time for audio packets over the Internet are reviewed.

6.2.1 Background

A typical audio segment may be considered as constituted of talkspurt periods during which the audio activity is carried out, and silence periods during which no audio packet is generated. In order for the receiving site to reconstruct the audio conversation, the audio packets constituting a talkspurt must be played out in the order they were emitted at the sending site. If the delay between the arrival of subsequent packets is constant (i.e. the underlying transport network is jitter-free) a receiving site may simply play out the arriving audio packets as soon as they are received. Unfortunately, this is only rarely the case, since jitter-free, ordered, on-time packet delivery almost never occurs in today's packet-switched networks. Those variations in the arrivals of subsequent packets strongly depend on the traffic conditions of the underlying network. Packet loss percentages (due to the effective loss and damage of packets as well as late arrivals) often vary between 15% and 40% [94]. In addition, extensive experiments with wide-area network testbeds have shown that the delays between consecutive packets may also be as much as 1.5 seconds, thus impairing real-time interactive human conversations. New protocol suites such as the Resource Reservation Protocol (RSVP) [98] might eventually ameliorate the effect of jitter and improve the quality of the audio service over the Internet, but they are not yet widely used. On the other hand, the most used approach is to adapt the applications to the jitter present on the network. Hence, to transport audio over a non-guaranteed packet-switched network, audio samples are encoded (usually with some form of compression), inserted into packets that have creation timestamps and sequence numbers, transported by the network, received in a playout buffer, decoded in sequential order, and finally played out by the audio device, as seen in Fig. 6.1. A symmetric scheme is used in the other direction for interactive conversation.

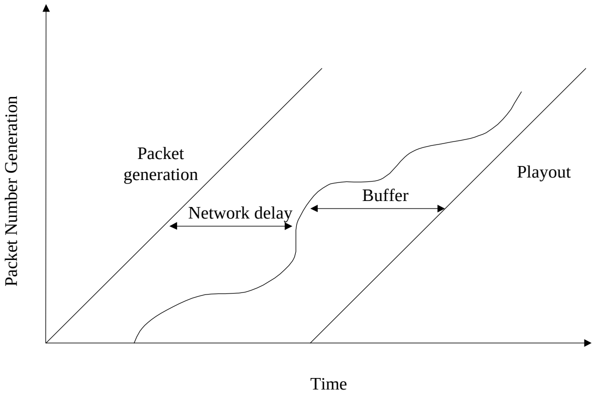

Figure 6.0: Smoothing out jitter delay at the receiver..

The smoothing playout buffer is used at the receiver in order to compensate for variable network delays. Received audio packets are queued into the buffer, and the playout of each packet of a given talkspurt is delayed for some quantity of time beyond the reception of the first packet of that talkspurt. In this way, dynamic playout buffers can hide, at the receiver, packet delay variance at the cost of additional delay. A crucial tradeoff exists between the length of the imposed additional quantity of delay and the amount of lost packets due to their late arrival: the longer the additional delay, the more likely it is that a packet will arrive before its scheduled playout deadline. However, too long playout delays may in turn seriously compromise the quality of the conversation over the network.

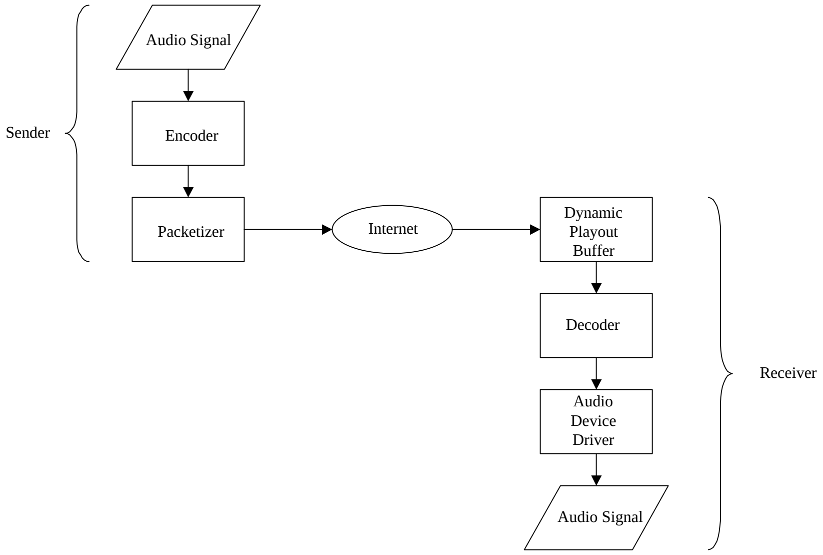

Figure 6.1: Audio data flow over the Internet.

Typical acceptable values for the end-to-end delay between packet audio generation at the sending site and its playout time at the receiver are below the threshold of 200-250 msec, furthermore a percentage of no more than 5 - 10% of packet loss is considered quite tolerable in human conversations [73]. Besides adjusting the audio playout delay in order to compensate for the effect of the jitter, modern audio tools typically make also use of error and rate control mechanisms based on a technique known as forward error correction (FEC) to reconstruct many lost audio packets [73]. For example, the INRIA audio tool adjusts the audio packet send rate to the current network conditions, adds redundant information to packets (under the form of highly compressed versions of a number of previous packets) when the loss rate surpasses a certain threshold, and establishes a feedback channel to control the send rate and the redundant information. Simply put, the complete process is controlled by an open feedback loop that selects among different available compression schemes and the amount of redundancy needed, as described in the following. If the network load and the packet loss are high, the amount of compressed redundant information carried in each packet is increased by adding to each packet compressed version of the previous two to four audio packets. In 5-seconds intervals the receiver returns (using the Real Time Protocol suite RTP-RTCP [99]) quality of service reports to the sender in order to regulate and adapt the quantity of redundant information being sent. As discussed above, efficient playout adjustment mechanisms have been developed to minimize the effect of delay jitter. Typically, a receiving site in an audio application buffers packets and delays their playout time. Such a playout delay may be kept constant for the duration of the audio conversation, or dynamically adjusted from one talkspurt to the next. Due to the fluctuating end-to-end (application-to-application) delays experienced over the Internet, constant, non-adaptive playout delays may result in unsatisfactory quality for audio applications. Hence, two are the approaches widely exploited for adaptively adjusting playout time: the former approach keeps the same playout delay constant throughout a given talkspurt, but permits different playout delays in different talkspurts. In the latter approach, instead, the playout delay is adjusted on a per-packet basis. However, an adaptive adjustment on a per-packet basis may introduce gaps inside talkspurt and thus is considered as of being damaging to the perceived audio quality. On the contrary, the variation of the playout delay from a talkspurt to the next may introduce artificially elongated or reduced silence periods, but this is considered acceptable in the perceived speech if those variations are reasonably limited. Hence, the totality of the above mentioned tools adopt a mechanism for adaptively adjusting the playout delays on a per-talkspurt basis. However, in order to implement such a playout control mechanism, almost all the above cited audio applications make use of the following two strong assumptions.

1. An external mechanism exists that keeps synchronized the two system clocks at both the sending and the receiving site. Usually, the IP-based Network Time Protocol (NTP) is used for this purpose.
2. The delays experienced by audio packets on the network follow a Gaussian distribution.

Extensive experiments have been carried out that shown that the playout delay control mechanisms based on that two assuntions above may be adequate to obtain acceptable values for the tradeoff between the average playout delay and the loss due to late packet arrivals. However, in some circumstances, the cited mechanisms may suffer from a number of problems, especially when they are deployed over wide-area networks. In particular, the following problems may be pointed out [94,100]:

- The ``external" software-based mechanisms (e.g. the NTP protocol) used to maintain the system clocks synchronized at both the sending and the receiving sites are not typically widespread all over the Internet. In addition, those mechanisms may turn out to be too much inaccurate to cope with the real-time nature of the audio generation/playout process. For example, even if the NTP protocol may achieve computer clock synchronization within a few tens of milliseconds over most paths in the Internet of today, however, there may be frequent exceptions with synchronization values up to a few hundreds of milliseconds, especially if a client host is not directly connected to a primary server of the NTP hierarchy but achieves synchronization through a stratum-2 (or higher) server via a congested link [101]. The problem with clock synchronization is that if the two different clocks (respectively, at the source and at the destination) do not run at the same rate and the synchronization mechanism is not sufficiently accurate, they will tend to drift further and further apart. Extensive experiments have shown that the above mentioned behavior may have a very negative impact on the provided formulas for the calculation of the playout time, thus resulting in an increased number of lost packets [100].
- The widely adopted assumption that the packet transmission delays over the Internet follow a Gaussian distribution seems to be a plausible conjecture only for those limited time intervals in which the overall load of the underlying network is quite light. Indeed, recent experimental studies carried out over the Internet have indicated the presence of frequent and conspicuously large end-to-end delay spikes for periodically generated packets (as is the case with audio packets) [73, 102].

(.... omissis....)

References

[73] J. Bolot, H. Crepin, A. Vega Garcia, ``Analysis of Audio Packet Loss on the Internet", in Proc. of Network and Operating System Support for Digital Audio and Video, 163-174, Durham (NC), 1995

[93] V. Hardman, M.A. Sasse, I. Kouvelas, ``Successful Multi-Party Audio Communication over the Internet", in Communications of the ACM 41:74-80, 1998

[94] S.B. Moon, J. Kurose, D. Towsley, ``Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms, in ACM Multimedia Systems 6:17-28, 1998

[95] H. Schulzrinne, ``Voice Communication across the Internet: a Network Voice Terminal", Tech. Rep., Dept. of ECE and CS, Univ. of Massachusetts, Amherst (MA), 1992

[96] V. Jacobson, S. McCanne, vat, ftp://ftp.ee.lbl.gov/conferencing/vat/

[97] M. Macedonia, D. Brutzmann, ``mbone Provides Audio and Video across the Internet", in IEEE Computer Magazine 21:30-35, 1994

[98] L. Zhang, ``RSVP: A New Resource Reservation Protocol", in IEEE Network Magazine 7:8-18, 1993

[99] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, ``RTP: A Transport Protocol for Real-Time Applications", Request for Comments 1889, IETF, Audio-Video WG, 1995

[100] A. Vega Garcia, ``Mecanismes de Controle pour la Transmission de l'Audio sur l'Internet", Doctoral Thesis in Computer Science, University of Nice-Sophia Antipolis, Ecole Doctoral SPI, 1996

[101] D. L. Mills, ``Improved Algorithms for Synchronizing Computer Network Clocks", in Proc. of ACM SIGCOMM'94, 317-327, London (UK), 1994

[102] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson ``On the Self-Similar Nature of Ethernet Traffic", in IEEE/ACM Trans. on Networking 2:1-15, 1994