Understanding Digital Signal Processing and Streaming

Table of Contents

The following information provides a basic introduction to Digital Signal Processing, and Streaming technologies. These two technologies are used extensively in the Media Server, therefore understanding these concepts will assist developers in creating customized media services for the Media Server.

Introduction to Digital Signal Processing

Digital Signal Processing, as the name suggests, is the processing of signals by digital means. A signal in this context can mean a number of different things. Historically the origins of signal processing are in electrical engineering, and a signal here means an electrical signal carried by a wire or telephone line, or perhaps by a radio wave. More generally, however, a signal is a stream of information representing anything from stock prices to data from a remote-sensing satellite. The term "digital" originates from the word "digit", meaning a number, therefore "digital" literally means numerical. This introduction to DSP will focus primary on two types digital signals: audio and voice.

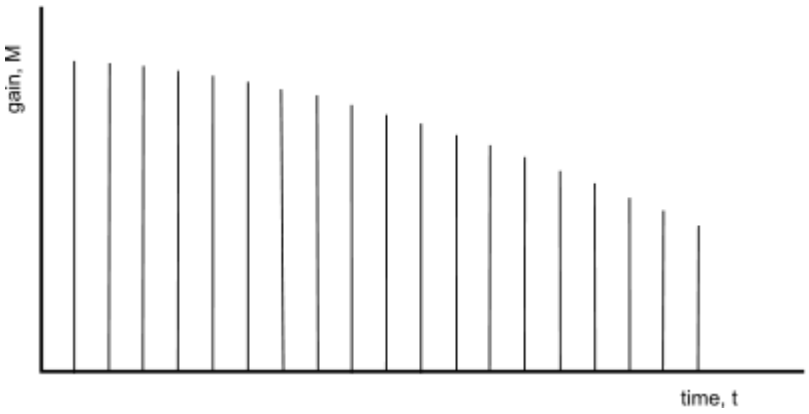
Analog and Digital Signals

Data can already be in a digital format (for example, the data stream from a Compact Disk player), and will not require any coversion. In many cases however, a signal is received in the form of an analog electrical voltage or current, produced by a microphone or other type of transducer. Before DSP techniques can be applied to an analog signal, it must be converted into digital form. Analog electrical voltage signals can be digitized using an analog-to-digital converter (ADC), which generates a digital output as a stream of binary numbers. These numbers represent the electrical voltage input to the device at each sampling instant.

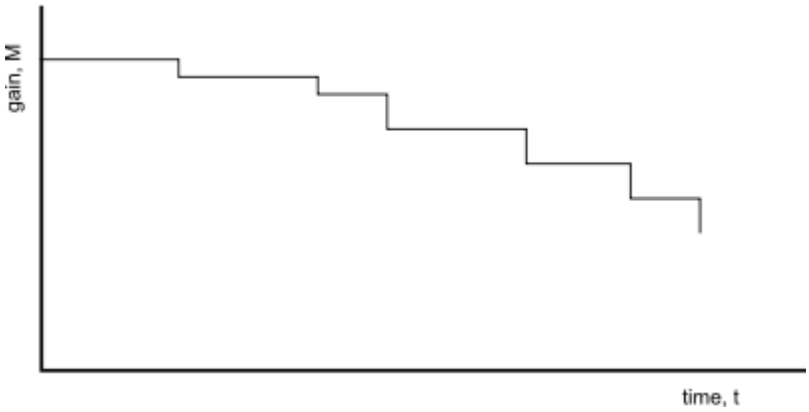
Discrete Signals

When converting a continuous analog signal to a digital signal, the analog signal must be converted to a signal format that computers can analyze and perform complex calculations on. Discrete Signals are easily stored and transmitted over digital networks and have the ability to be discrete in magnitude, time, or both.

Discrete-in-time values only exist at certain points in time. For example, if a sample of discrete-in-time data is taken at a point in time where there is no data, the result is zero.



Discrete-In-Magnitude values exist across a time range, however, the value of the datum in each time range consists of one constant result, rather than a variable set of results.



By converting continuous analog signals to discrete signals, finer computer data analysis is possible, and the signal can be stored and tranmitted efficiently over digital networks.

Sampling, Quantization, and Packetization

Sampling is the process of recording the values of a signal at given points in time. For ADCs, these points in time are equidistant, with the number of samples taken during one second dictating the called sample rate. It's important to understand that these samples are still analogue values. The mathematic description of the ideal sampling is the multiplication of the signal with a sequence of direct pulses.

Quantization is the process of representing the value of an analog signal by a fixed number of bits. The value of the analog signal is compared to a set of pre-defined levels. Each level is represented by a unique binary number, and the binary number that corresponds to the level closest to the analog signal value is chosen to represent that sample.

Sampling and quantization prepare digitized media for future processing or streaming. However, streaming and processing over individual samples is not effective for high volumes of data transferred via a network. The risk of data-loss is much higher when a large portion of data is transferred in a block. Networked media should be transmitted using media packets that carry several samples, thereby reducing the risk of data loss through the transmission process. This process is referred to as packetization.

Transfer Protocols

The Real-time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP) and the Real-time Transport Control Protocol (RTCP) were specifically designed to stream media over networks. The latter two are built on top of UDP.

Real-time Transport Protocol

RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP

does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by the Real-time Control Protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers.

A RTP packet consists of a RTP header, followed by the data to send. In the RTP specification, this data is referred to as the payload. The header is transmitted in network byte order, just like the IP header. The Figure 5 shows the RTP header format.

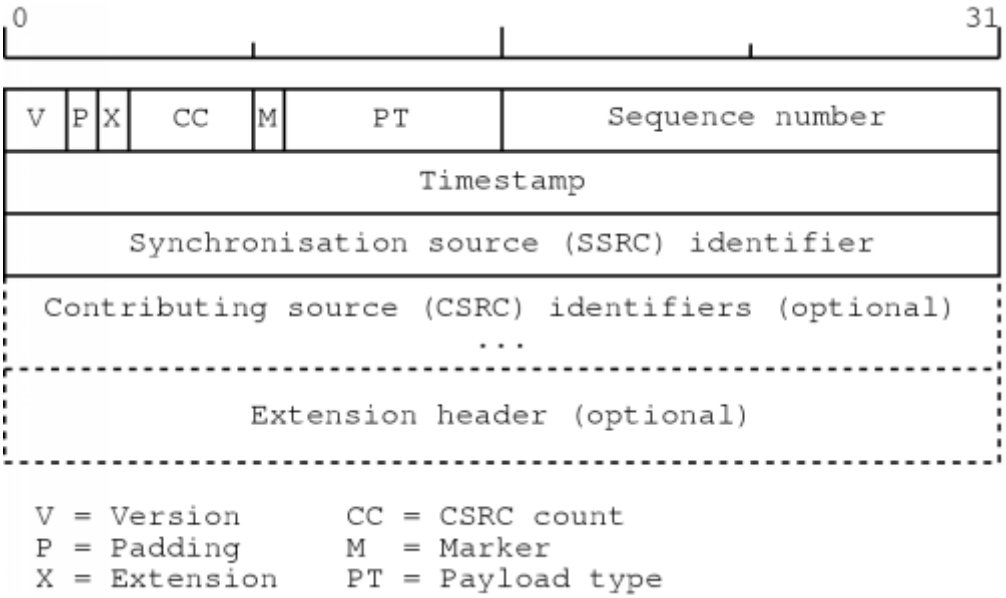


Table 1. Supported RTP Formats

Header Component	Description
V (Version)	Contains the version number of the RTP protocol. For example, the current version number is 2. This part of the header consumes 2 bits of the RTP packet.
P (Padding)	Contains padding bytes, which are excluded from the payload data count. The last padding byte contains the number of padding bytes present in the packet. Padding may be required for certain encryption algorithms that need the payload to be aligned on a multi-byte boundary.
X (Extension)	Specifies whether the header contains an Extension Header.
CC (CSRC Count)	Specifies how many contributing sources are specified in the header.
M (Marker)	Contains arbitrary data that can be interpreted by an application. The RTP specification does not limit the information type contained in this component of the header. For example, the Marker component might specify that media data is contained within the packet.

Header Component	Description
PT (Payload Type)	Specifies the type of data the packet contains, which determines how an application receiving the packet interprets the payload.
Sequence Number	Contains a unique numerical value, that can be used by applications to place received packets in the correct order. Video streams rely on the sequence number to order the packets for individual video frames received by an application. The starting number for a packet stream is randomized for security reasons.
Time Stamp	Contains the synchronization information for a stream of packets. The value specifies when the first byte of the payload was sampled. The starting number for the Time Stamp is also randomized for security reasons. For audio, the timestamp is typically incremented with the amount of samples in the packet so the receiving application can play the audio data at exactly the right time. For video, the timestamp is typically incremented per image. One image of a video will generally be sent in several packets, therefore the pieces of data will have the same Time Stamp, but use a different Sequence Number.
SSRC ID	Contains the packet Synchronization Source (SSRC) identifier of the sender. The information contained in this component of the header is used to correctly order multiple RTP streams contained in a packet. This scenario often occurs when an application sends both video and audio RTP streams in one packet. So the receiving application can correctly order and synchronize the data, the identifier is chosen randomly. This reduces the chance of a packet in both streams having the same identifier.
CSRC ID	Contains one (or more) Contributing Source (CSRC) identifiers for each RTP stream present in the packet. To assist audio streams re-assembly, the SSRC IDs can be appended to this packet component. The SSRC ID of the packet then becomes the source identifier for the forwarded packet.
Extension Header	Contains arbitrary information, specified by the application. The RTP defines the extension mechanism only. The extensions contained within the Extension Header are controlled by the application.

RTP headers do not contain a payload length field. The protocol relies on the underlying protocol to determine the end of the payload. For example, in the TCP/IP architecture, RTP is used on top of UDP, which does contain length information. Using this, an application can determine the size of the whole RTP packet and after its header has been processed, the application automatically knows the amount of data in its payload section.

Real-time Transport Control Protocol

The RTP is accompanied by a control protocol, the Real-time Transport Control Protocol (RTCP). Each participant of a RTP session periodically sends RTCP packets to all other participants in the session for the following reasons:

- To provide feedback on the quality of data distribution. The information can be used by the application to perform flow and congestion control functions, and be used for diagnostic purposes.
- To distribute identifiers that are used to group different streams together (for example, audio and video). Such a mechanism is necessary since RTP itself does not provide this information.
- To observe the number of participants. The RTP data cannot be used to determine the number of participants because participants may not be sending packets, only receiving them. For example, students listening to an on-line lecture.
- To distribute information about a participant. For example, information used to identify students in the lecturer's conferencing user-interface.

There are several types of RTCP packets that provide this functionality.

- Sender
- Receiver
- Source Description
- Application-specific Data

Sender reports (SR) are used by active senders to distribute transmission and reception statistics. If a participant is not an active sender, reception statistics are still transmitted by sending receiver reports (RR).

Descriptive participant information is transmitted in the form of Source Description (SDES) items. SDES items give general information about a participant, such as their name and e-mail. However, it also includes a canonical name (CNAME) string, which identifies the sender of the RTP packets. Unlike the SSRC identifier, the SDES item stays constant for a given participant, is independent of the current session, and is normally unique for each participant. Thanks to this identifier it is possible to group different streams coming from the same source.

There is a packet type that allows application-specific data (APP) to be transmitted with RTP data. When a participant is about to leave the session, a goodbye (BYE) packet is transmitted.

The transmission statistics which an active sender distributes, include both the number of bytes

sent and the number of packets sent. The statistics also include two timestamps: a Network Time Protocol (NTP) timestamp, which gives the time when this report was created, and a RTP timestamp, which describes the same time, but in the same units and with the same random offset of the timestamps in the RTP packets.

This is particularly useful when several RTP packet streams have to be associated with each other. For example, if both video and audio signals are distributed, there has to be synchronization between these two media types on playback, called inter-media synchronization. Since their RTP timestamps have no relation whatsoever, there has to be some other way to do this. By giving the relation between each timestamp format and the NTP time, the receiving application can do the necessary calculations to synchronize the streams.

A participant to a RTP session distributes reception statistics about each sender in the session. For a specific sender, a reception report includes the following information:

- Fraction of lost packets since the last report. An increase of this value can be used as an indication of congestion.
- Total amount of lost packets since the start of the session.
- Amount of inter-arrival jitter, measured in timestamp units. When the jitter increases, this is also a possible indication of congestion.
- Information used by the sender to measure the round-trip propagation time to this receiver. The round-trip propagation time is the time it takes for a packet to travel to this receiver and back.

Because the RTCP packets are sent periodically by each participant to all destinations, the packet broadcast interval should be reduced as much as possible. The RTCP packet interval is calculated from the number of participants and the amount of bandwidth the RTCP packets may occupy. To stagger the broadcast interval of RTCP packets to participants, the packet interval value is multiplied by a random number.

Jitter

The term Jitter refers to processing delays that occur at each endpoint, and are generally caused by packet processing by operating systems, codecs, and networks. Jitter affects the quality of the audio and video stream when it is decoded by the receiving application.

End-to-end delay is caused by the processing delay at each endpoint, and may be caused in part by IP packets travelling through different network paths from the source to the destination. The time it takes a router to process a packet depends on its congestion situation, and this may also vary during the session.

Although a large overall delay can cause loss of interactivity, jitter may also cause loss of intelligibility. Though Jitter cannot be totally removed, the effects can be reduced by using a Jitter Buffer at the receiving end. The diagram below shows effect with media buffer and without media buffer

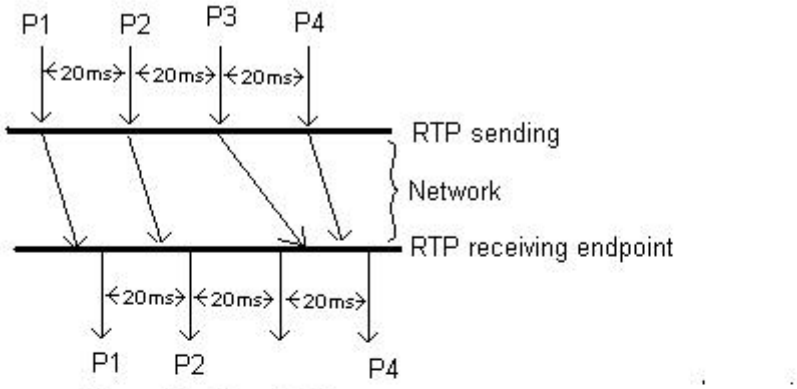


Fig a. No Jitter Buffer

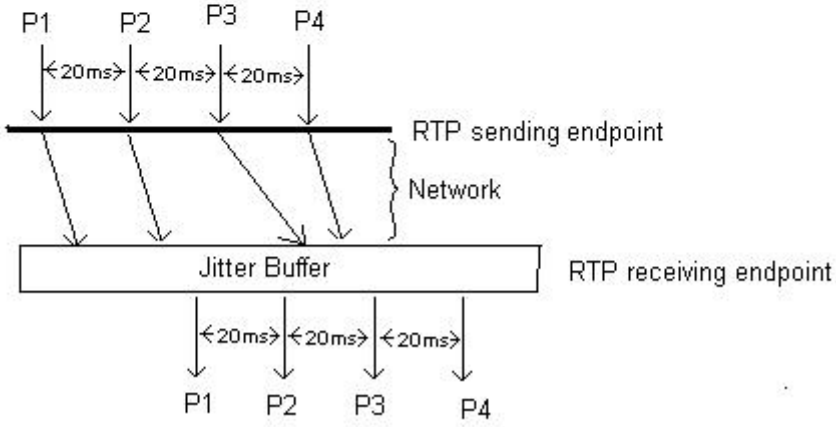


Fig b. With Jitter Buffer

Fig a. Shows that packet 3 is lost as it arrived late. Fig b uses Jitter buffer and hence arrived packets are stored in jitter and media components reads from Jitter once its half full. This way even if Packet 3 arrives little late, its read by the components.