International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.1021

(07/2012)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Multimedia Quality of Service and performance – Generic and user-related aspects

Buffer models for development of client performance metrics

Recommendation ITU-T G.1021

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	**G.1000–G.1999**
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.1021

Buffer models for development of client performance metrics

Summary

Recommendation ITU-T G.1021 defines buffer models that describe the behaviour of the client buffers used at the receiver side of audio/video applications.

Buffer models receive as input IP packets associated with meta information on their content and provide as output client states associated with relevant timing information.

The buffer models are intended to be used for the development of client performance metrics in order to complement the IP-layer metrics defined in Recommendation ITU-T Y.1540.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.1021	2012-07-14	12

Table of Contents

Recommendation ITU-T G.1021

Buffer models for development of client performance metrics

1 Scope

Audio and video applications currently used in packet networks implement client buffers (e.g., de-jitter buffers and play-out buffers) at the receiver side. The behaviour of these buffers is essential to the end-to-end performance of these applications. This Recommendation defines models for these client buffers for audio/video applications currently used in actual networks, in order to estimate the client state based on the dynamic network behaviour.

In this Recommendation, different buffer models are derived depending on the different applications taken into account. However, these buffer models may be based on common intrinsic characteristics: i.e., buffer size, service rate, initial delay and re-buffering delay. Example buffer performance metrics include peak/average/instantaneous buffer occupation.

In this Recommendation, buffer models may accept as input, either live traffic arriving on measurement points, packet capture (pcap) files, or post-processed files. Buffer models are expected to provide client states (e.g., codec play-out begin, buffer overflow, buffer dry out, play-out resume) associated with relevant timing information. As part of future work, these buffer models could be used to derive new performance parameters and associated statistics in order to complement the IP-layer metrics defined in Recommendation Y.1540.

It is out of scope for this Recommendation to define specific de-jitter buffers for use in actual client media stream delivery.

In this Recommendation the following applications, rate and buffer characteristics, and associated protocol stacks are considered:

a) Audio applications:

   i) Ideally constant fill, constant play-out rate and relatively low bandwidth.

b) Video "low bitrate mode" quarter common intermediate format-quarter video graphics array (QCIF-QVGA) resolutions, mostly for mobile TV and streaming with the sub application areas:

   i) Linear mobile TV over real-time transport protocol (RTP) (including mobile TV over a 3G mobile network with multimedia broadcast/multicast services (MBMS) and with unicast transport over RTP/UDP/IP).

   ii) Multimedia streaming (including 3GPP packet-switched streaming service (PSS) with transport over RTP/UDP/IP).

   iii) [IETF RFC 3984] variable bit rate, up to 2 Mbps, with separate buffers for audio and video, and separate play-out rates for each buffer.

c) Video "high bitrate mode" standard-definition (SD) and high-definition (HD) television, mostly for Internet Protocol television (IPTV) with the following sub-application areas:

   i) Linear broadcast TV (including transmission over MPEG2-TS/RTP/UDP/IP, MPEG2-TS/UDP/IP and RTP/UDP/IP transport).

   ii) Video on-demand (including transmission over MPEG2-TS/RTP/UDP/IP, MPEG2-TS/UDP/IP and RTP/UDP/IP transport).

   iii) [IETF RFC 2250], constant bit rate (CBR) or variable bit rate (VBR) up to 50 Mbps, with separate buffers for audio and video, and separate play-out rates for each buffer.

d) Over-the-top video applications:

  i) Non-adaptive streaming (MPEG4_fileformat/HTTP/TCP, HTTP progressive download, HTTP partial download, chunks):

    – Progressive download: TCP transport, TCP flow control (65Mbps super-peak rates, peak rate depends on network conditions and path), classic single buffer, leaky bucket at constant frame rate.

  ii) Adaptive streaming dynamic adaptive streaming over HTTP (DASH/TCP):

    – Chunk Delivery: (e.g., DASH) bursts of packets, like sequential FTP file transfers.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.1020] Recommendation ITU-T G.1020 (2006), *Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks*.

[ITU-T H.222.0] Recommendation ITU-T H.222.0 (2006) | ISO/IEC 13818-1:2007, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.

[ITU-T Y.1540] Recommendation ITU-T Y.1540 (2010), *Internet protocol data communication service – IP packet transfer and availability performance parameters*.

[IETF RFC 2250] IETF RFC 2250 (1998), *RTP Payload Format for MPEG1/MPEG2 Video*.

[IETF RFC 3984] IETF RFC 3984 (2005), *RTP Payload Format for H.264 Video*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 decoding time-stamp (DTS) [ITU-T H.222.0]: A field that may be present in a PES packet header that indicates the time that an access unit is decoded in the system target decoder.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ARQ	Automatic Retransmission on Request
CBR	Constant Bit Rate
DASH	Dynamic Adaptive Streaming over HTTP
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DTS	Decoding Time Stamp
EWMA	Exponentially Weighted Moving Average

HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IPTV	Internet Protocol Television
FEC	Forward Error Correction
FTP	File Transfer Protocol
MBMS	Multimedia Broadcast/Multicast Services
MPEG	Moving Picture Experts Group
MPEG-TS	MPEG Transport Stream
NTP	Network Time Protocol
OTT	Over The Top
PCAP	Packet Capture
PCR	Program Clock Reference
PDV	Packet Delay Variation
PES	Packetized Elementary Stream
PLC	Packet Loss Concealment
PLL	Phase-Locked Loop
PSS	Packet-switched Streaming Service
QCIF	Quarter Common Intermediate Format
QVGA	Quarter Video Graphics Array
RTP	Real-time Transport Protocol
STB	Set-Top Box
VBR	Variable Bit Rate

5 Conventions

The buffer model behavior is described by the state-machine pseudocode provided in Annex A of this Recommendation.

Pseudocode is an informal, high-level description of the operating principle of an algorithm. It uses the structural conventions of a programming language, but is intended for human reading rather than machine reading. Pseudocode typically omits details that are not essential for human understanding of the algorithm, such as variable declarations, system-specific code and some subroutines.

Note that no standard for pseudocode syntax exists. In this Recommendation, pseudo-code similar to the C programming language has been used.

6 De-jitter buffers

6.1 Network jitter and de-jitter buffers

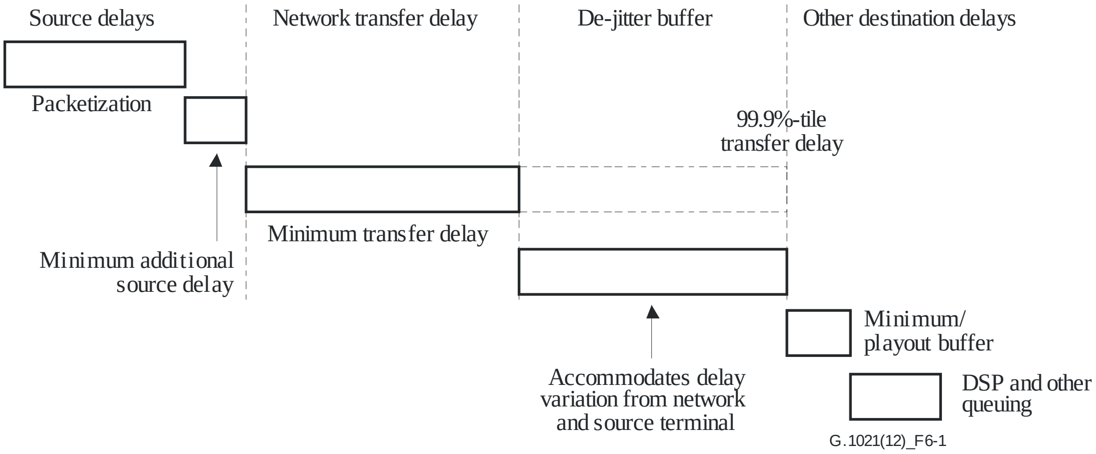
In the context of computer networks, the term "jitter" is often used as a measure of the variability over time of packet latency across a network. A network with constant latency has no variation and therefore no jitter. However, the word is imprecise and, in the context of packet-switched networks, the term "packet delay variation" is often preferred over "jitter" (see [ITU-T Y.1540]).

In order to smooth out the natural packet delay variation or jitter effects of asynchronous networks and to synchronize play-out between sender and receiver, most techniques introduce an additional delay, implemented by having the packets arriving on the reception side equipment being temporarily stored in a buffer. This buffer is designed to counteract the jitter that is introduced by the network, until the moment the audio signal or the image is delivered to the decoding scheme.

This buffer is called the "de-jitter buffer." In order to ensure a continuous play-out of streaming audio or video, the de-jitter buffer parameters have to be tuned, such that, at the moment an audio or a picture is to be played-out, the entire content of the audio or picture resides in the buffer.

There is a trade-off between the end-to-end delay caused by the de-jitter buffer and the packet loss. Packets that arrive before timeout are played out in the same order as they were created at the sender side. Those packets that arrive after timeout are discarded. An ideal algorithm should neither delay the play-out time excessively, nor discard too many packets.

As shown in Figure 6-1, the end-to-end delay experienced by an IP packet (carrying speech or voice band applications) can be decomposed in several parts: packetization delay, network transfer delay, de-jitter buffer delay (accommodating delay variation from source and network) and play-out buffer delay.



NOTE – Figure 6-1 is an exact replica of Figure 5 of [ITU-T G.1020].

Figure 6-1 – Delay of packet networks and network elements

This type of decomposition can be generalized to most audio/video applications.

6.1.1 Fixed size de-jitter buffers

The size of the play-out buffer may be kept fixed; this scheme is referred to as fixed size play-out buffering, or static scheduling. The end-to-end delay is kept constant for all packets in a session. If packets exceed a given transmission time they are discarded.

Although the fixed method, which uses a fixed buffer size, is easy to implement, it can result in unsatisfactory audio or video quality because there is no optimal delay when network conditions vary with time. The fluctuating end-to-end delays experienced in IP networks may cause delay to increase to a level that may be annoying to users (when the buffer is too large), or may cause packet losses due to their late arrival times (when the buffer is too small).

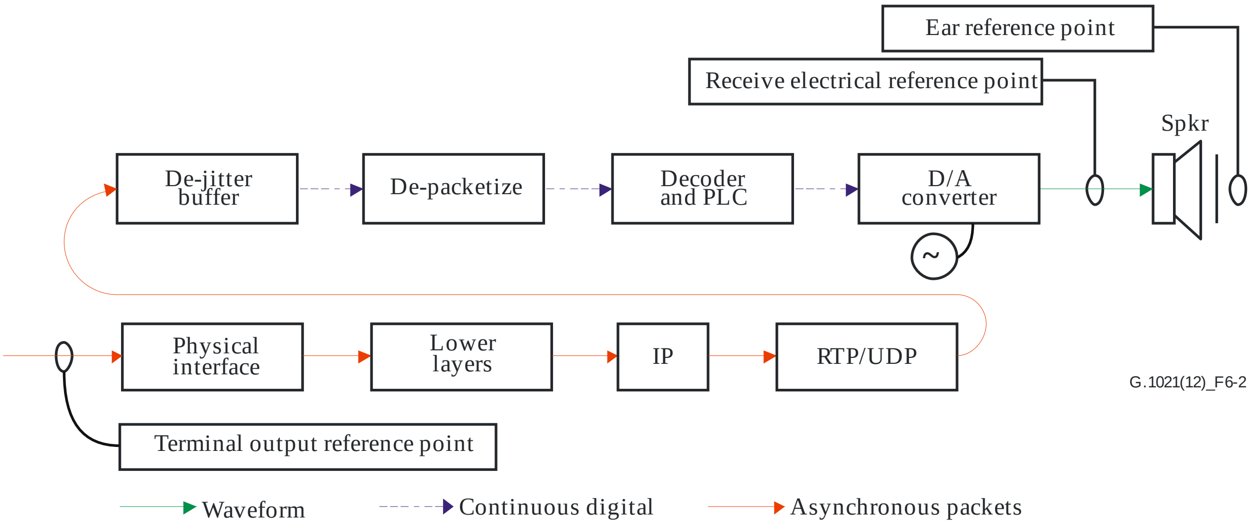
6.1.2 Adaptive size de-jitter buffers

The size of the play-out buffer may also be adaptively adjusted during the transmission; this scheme is referred to as adaptive size play-out buffering, or dynamic scheduling.

Adaptive techniques adapt the delay to changing network conditions, thus preventing the use of long delays during low-congestion conditions and vice versa. In this case, the network behavior is monitored and network delays are continuously estimated and the play-out delay is dynamically adjusted.

6.2 De-jitter buffers for audio applications

Figure 6-2 indicates the positions of measurement points and system components for speech and voiceband applications utilizing IP networks. As shown, the de-jitter buffer is placed after processing the IP and RTP/UDP layers.



NOTE – Figure 6-2 is an exact replica of Figure 3 of [ITU-T G.1020].

Figure 6-2 – Destination terminal components for audio applications

For audio applications, the main objective of de-jitter buffers is to keep the packet loss rate under a given threshold (e.g., 5 per cent) and to keep the end-to-end delay as small as possible.

Audio applications usually take advantage of the signal structure alternating talkspurts and silent periods. With adaptive methods, play-out adjustment is performed during the silent periods between talkspurts. The adjustment is done on the first packet of the talkspurt; all packets in the same talkspurt are scheduled to play out at fixed intervals following the play-out of the first packet. This mechanism uses the same play-out delay throughout a given talkspurt, but permits different play-out delays for different talkspurts. The variation of the play-out delay may introduce artificially elongated or reduced silence periods, but such modification of silence periods is considered acceptable in the perceived speech if that variation is reasonably limited.

If the jitter is so large that it causes packets to be received out of the range of this buffer, the out-of-range packets are discarded and dropouts are heard in the audio. Usually packet loss concealment (PLC) features are employed together with the decoder in order to mask the possible effects of packet losses. One PLC method is artificial regeneration of the packet received prior to the lost packet, followed by insertion of the duplicated packet into the gap. In another method, the digital value of the dropped packet is estimated by interpolation and an artificially generated packet is inserted on that basis. However, when jitter exceeds what the digital signal processor (DSP) can handle to make up for the missing packets, the quality of service (QoS) of audio is degraded.

6.3 De-jitter buffers for "low bitrate mode" and "high bitrate mode" video applications

The decomposition of system components can also be developed for Video "low bitrate mode" and Video "high bitrate mode" applications that are also based on UDP, as shown in Figure 6-3.

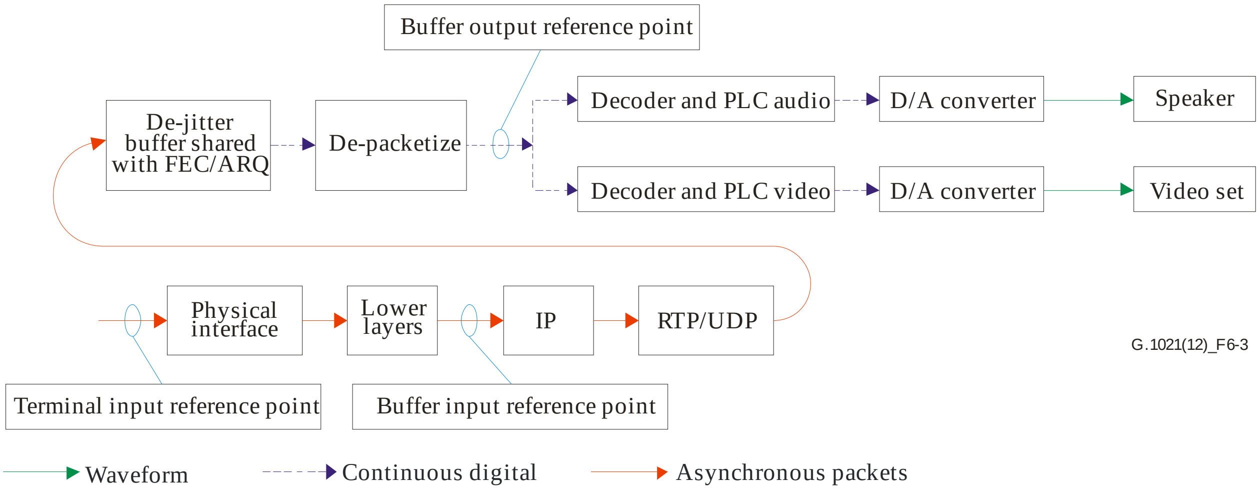


Figure 6-3 – Destination terminal components for "low bitrate mode" and "high bitrate mode" video applications

In the case of video applications, techniques for combating transmission errors have been developed along two directions.

In the first direction, traditional error control and recovery schemes for data communications have been extended for video transmission. These application-layer loss correction techniques aim at lossless recovery. Examples of such schemes include forward error correction (FEC) and automatic retransmission on request (ARQ). Usually FEC or ARQ mechanisms are performed together with de-jitter recovery in the same functional block.

In the second direction, signal-reconstruction and error-concealment techniques have been proposed that strive to obtain a close approximation of the original signal, or attempt to make the output signal at the decoder the least objectionable to human eyes. However, unlike data transmissions where lossless delivery is absolutely required, human eyes can tolerate a certain degree of distortion in image and video signals. For instance, video encoders such as ITU-T H.264 may insert resynchronization markers in the compression loop as well as a structural bitstream arrangement to allow the decoder to continue the decoding process despite erasures encountered in the compressed image, which essentially prevents frame skipping.

If the FEC source block structure is chosen with proper alignment, for example with the random access points of the video stream, then the FEC latency can be absorbed into the video decoding buffer latency. This would mean that the protection period is in general not additive to the end-to-end latency, but generally lower.

6.4 De-jitter buffers for over-the-top video applications

Video delivered over the Internet without involvement of the service provider's own video service (but on the public Internet) is defined as over-the-top (OTT).

The delivery mode may be based on non-adaptive streaming (e.g., HTTP progressive download), where transport is accomplished using TCP flow control (the peak rate depending on network conditions), and where play-out is performed at constant frame rate.

It may also be based on adaptive streaming (DASH/TCP), which consists of delivering chunks (i.e., bursts of packets) like sequential file transfers. Well-known implementations include: Microsoft smooth streaming, HTTP live streaming by Apple, and Adobe dynamic streaming.

The decomposition in system components for OTT video applications may be adapted as shown in Figure 6-4.

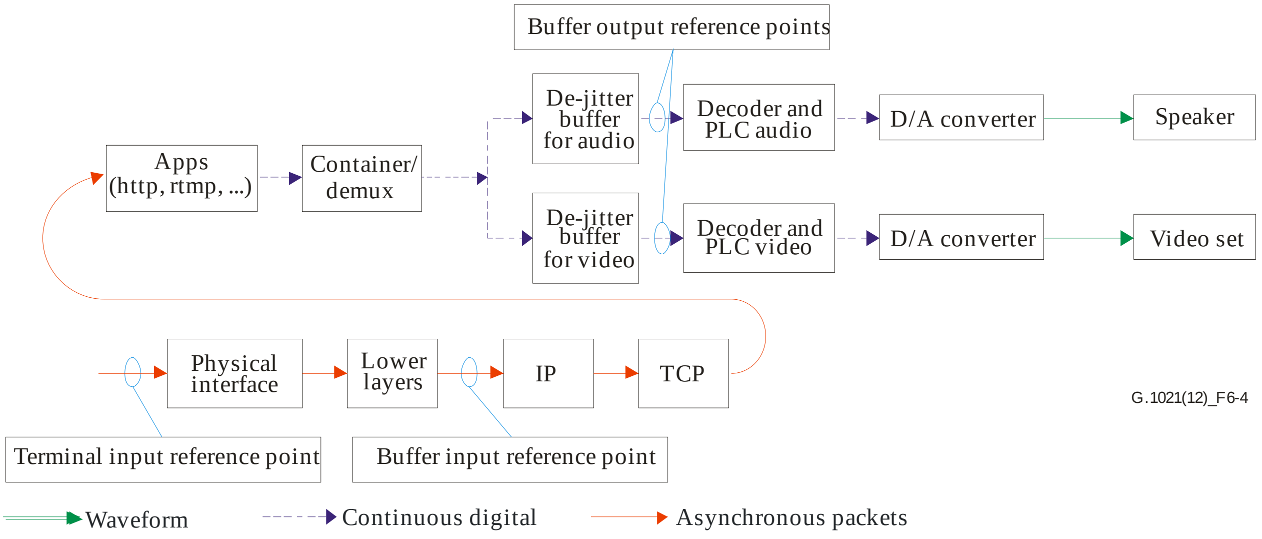


Figure 6-4 – Destination terminal components for OTT video applications

6.5 De-jitter buffer performance outcomes

In the case of UDP-based applications, the performance of a de-jitter buffer may be characterized by two types of outcomes corresponding to the following undesirable situations:

- The de-jitter buffer overflows, which corresponds to a new packet arriving in a full de-jitter buffer which is unable to store it; this new packet is lost,
- The de-jitter buffer dries out, which corresponds to an empty de-jitter buffer upon play-out request by the decoder.

In the case of OTT applications, the performance of a de-jitter buffer is different because the TCP protocol adapts the stream bitrate according to network conditions. Therefore, the performance will be characterized by the following outcomes: the buffer is playing, the buffer is stopped, or the buffer is buffering or re-buffering.

When the buffer is buffering, play-out devices usually display a black screen with an icon showing that a buffering or re-buffering process is taking place. For re-buffering the last displayed frame is typically displayed.

7 De-jitter buffer models

Even if all manufacturers use de-jitter buffers in the reception side equipment, many different technical implementations may exist, depending on manufacturers, on applications (video or audio) and on the protocols used (RTP/UDP, HTTP, etc.).

A general description of the de-jitter buffer model is shown in Figure 7-1.

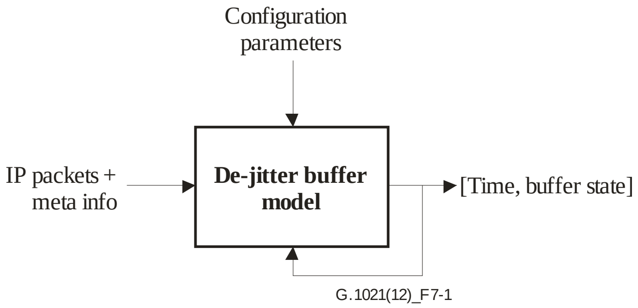


Figure 7-1 – De-jitter buffer model

The de-jitter buffer has four main configuration parameters: maximum buffer size, service strategy, initial buffering size and re-buffering size.

The **maximum buffer size** is the amount of memory dedicated to the de-jitter buffer.

The **service strategy** describes how packets or frames are played out by the application.

The **initial buffering size**, also called build-up time, is the amount of audio or video data of a stream which are stored in the buffer before playing out the first packet of this stream in order to minimize the risk of buffer dry-out.

The **re-buffering size** is the amount of audio or video data of a stream which are stored in the buffer before resuming play-out of the first packet of a flow, after the buffer has dried out, in order to minimize the risk that the buffer will dry out again immediately. Note that a similar configuration description is also used by the Broadband Forum Technical Report [b-BBF-TR-135], which defines in the case of video applications, set-top box (STB) characteristics such as:

- MaxDe-jitteringBufferSize: describes the maximum de-jittering buffer size, in bytes, supported by the STB.
- BufferSize: de-jittering buffer size, measured in milliseconds, on the receiver side of the customer premises equipment (CPE). The STB is responsible for converting this into the actual buffer size in bytes.
- BufferInitialLevel: describes the number of milliseconds worth of data that should be in the receiver side de-jittering buffer before play-out can start. Typically, this would be half of the buffer size.

The input to the de-jitter buffer model is composed of IP packets relative to the stream of interest, their arrival times, and meta information on their content relative to this stream as follows:

- offset of the different audio/video components contained in the packet;
- information relative to the frames of the audio component (frame counter, frame size, DTS, is the frame complete or not);
- information relative to the frames of the video component (frame counter, frame size, DTS, is the frame complete or not).

The output of the de-jitter buffer model is the state of the de-jitter buffer (i.e., playing, stopped, buffering, or re-buffering), together with the time of entry in each state.

This behaviour may be represented by the state machine described in Figure 7-2.

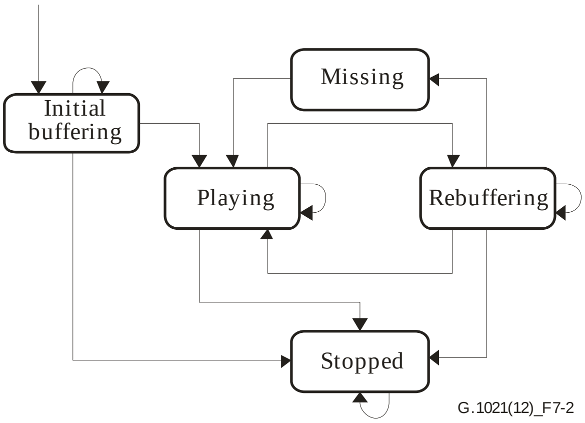


Figure 7-2 – State-machine description of the de-jitter buffer model

In the model, the state of the de-jitter buffer is estimated at time epochs corresponding to each packet arrival and to each frame extraction to the decoder. A pseudo-code description of this model is given in Annex A of this Recommendation.

7.1 Maximum buffer size

The maximum buffer size depends on the type of application.

Note that the maximum buffer size is usually expressed in milliseconds (ms). However, it can easily be expressed in bytes when the average bit rate of the application is known.

For audio applications (e.g., VoIP) generating IP packets every 20 ms, the buffer generally may contain only a small number of packets (up to two packets), thus corresponding to 40 ms.

For "high bitrate mode" video applications (based on MPEG2) generating IP packets every 4 ms on average, the buffer generally may contain a larger number of packets; a maximum buffer size of 80 ms is commonly used, corresponding to 20 packets in this case.

For OTT video applications (e.g., progressive download, DASH), the buffer generally may contain a very large number of packets. Classical implementations are currently using maximum buffer sizes of several tens of seconds.

7.2 Service strategy

The service strategy describes the way packets are played out by the decoder. Packets or frames are usually played out using a constant service rate, at least on relatively long periods of time. However, the service strategy may be more complex.

7.2.1 UDP-based applications

The service rate generally depends on the service clock used by the application server sending the real-time application packets (e.g., IPTV server). In an ideal case, the service rate matches the service clock (i.e., the decoder's clock matches the encoder's system clock).

Two buffer handling methods, push and pull, are commonly used. In the push method, the buffer controls the flow of data and periodically transfers appropriate data to the decoder. In the pull method, the decoder requests data from the buffer and the buffer transfers the requested data to the decoder.

Several options may be used for the service strategy:

- *The service rate is independent of the service clock used by the application server*; it can be based, for example, on a local free running oscillator (e.g., VoIP applications). In this case, depending on the offset between the clocks used at both sides, and on the duration of the communication, some effects can be noticed on the de-jitter buffer. It should be noted that, even when the service rate follows the service clock used by the application server, the use of a free running clock is generally required at the beginning of a clock recovery process during the stabilization period.

- *The service rate follows the service clock used by the application server, which is synchronized by a synchronization protocol* (e.g., network time protocol (NTP) or IEEE 1588). Such a mechanism may be accurate enough to ensure that no significant drift between service rate and service clock is observed during the service lifetime.
- *The service rate follows the service clock used by the application server, which is recovered at the de-jitter buffer side*. This clock recovery can be obtained either from the timestamps contained in the application packets, or from the knowledge of the nominal rate of the application (in case of constant bit rate flows). In general, the clock recovery is not perfect, as it is impacted by the packet delay variation of the network (e.g., low frequency components of the network packet delay variation (PDV) may not be filtered by the clock recovery mechanism).

Adaptive de-jitter strategies, such as adaptation of the play-out rate when the buffer is almost empty in an effort to avoid de-jitter buffer starvation, are often included as well. The adaptation logic is based on the jitter estimates computed from the arrival characteristics of the media packets. Adaptive de-jittering involves introducing discontinuities in the media play-out, which may be offensive to the listener or viewer.

UDP-based flows usually carry their own clock structure. For example, RTP flows and MPEG video flows contain service timestamps. In RTP flows, IP packets contain a 4-byte timestamp that is used to enable the receiver to play back the received samples at appropriate intervals. In addition, this feature generalizes the use of this protocol to variable bitrate flows.

In MPEG2 video flows, MPEG2 frame information is encapsulated within MPEG2 transport stream (MPEG2-TS) packets, which are in turn transported in IP packets, according to the MPEG/RTP/UDP/IPv4 protocol stack. MPEG2-TS flows are organized in streams related to the different components of the audio-visual application (video, audio, etc.). Each stream generates 188-byte packet elementary stream (PES) packets. These PES packets are then assembled in groups of seven, resulting in 1316-byte IP packets (7 × 188 = 1316). PES packets may contain a program clock reference (PCR) enabling the decoder to present synchronized content, such as audio tracks matching the associated video. Usually the PCR is imbedded in the video PES packets.

The bit rates outputted from MPEG-2 encoders may be constant bit rate (CBR) or variable bit rate (VBR). CBR MPEG-2 streams operate at a constant bandwidth data throughput regardless of the complexity of the video content while VBR MPEG-2 streams are encoded using different bit rates. The CBR technique is the predominant mode of operation for carrying video services over digital subscriber line (DSL) based networks.

If a service clock is available in IP packets, the clock reference timestamps, which are samples of the system time clock, are applicable both to a decoder and to an encoder. Therefore, they can be utilized to implement clock reconstruction control loops in decoders with sufficient accuracy. Annex D in [ITU-T H.222.0] gives some suggestions for implementing them.

Another method of slaving the decoder's clock to the received data stream is via a phase-locked loop (PLL). Classical PLL implementations are based on exponentially weighted moving average (EWMA) methods, where the weight is chosen in order to smooth the PLL reaction.

7.2.2 OTT applications

OTT applications typically use constant play-out rates for the audio and the video components. Typical play-out rates for the audio component are 44100 samples/s grouped in multiple samples per frame.

Typical play-out rates for the video component are either 24 frames/s or 30 frames/s, depending on the standard used for TV sets.

7.3 Initial buffering strategy

7.3.1 UDP-based applications

The initial buffering size added to the very first packet of a given flow must be chosen carefully since it impacts the performance of the application.

When small buffers are used (e.g., conversational audio services), the initial buffering size is constrained by the maximum buffer size itself and three main buffering strategies can be used.

- The first strategy assumes that the de-jitter buffer is full when media assessment begins. The monitored transmissions are allowed to consume the complete buffer resource before any buffer dry-out event is registered. In this strategy, the probability of a packet experiencing a buffer overflow becomes very high.
- The second strategy assumes that the de-jitter buffer is empty when media assessment begins, the opposite of the first approach. The monitored transmissions are allowed to completely fill-out the buffer before any buffer overflow event is registered. In this strategy, the probability of a packet experiencing a buffer dry-out increases. This strategy is often used in VoIP applications since the overall delay is the key parameter.
- The third strategy assumes that the de-jitter buffer is half-full when media assessment begins. The monitored transmissions are allowed to consume half the buffer resource before any buffer dry-out event is registered, or to completely fill-out the buffer before any buffer overflow event is registered. This strategy minimizes the probability of a packet experiencing a buffer overflow or a buffer dry-out, and it is commonly used in video applications.

When performing repeated/continuous stream monitoring operations, it should be permissible to pass the buffer occupation information from one assessment interval to the next. This would permit longer-term impairment (re-buffering) occurrence to be assessed while still satisfying the need for short-term results. More importantly, re-buffering will likely have a drastically negative effect on user experience and therefore, long-term assessment is seen as valuable.

7.3.2 OTT applications

OTT applications commonly use large buffers.

The choice of the initial buffering size is generally a compromise between the time a user has to wait before the video is displayed on the screen, and the probability of re-buffering events.

Current implementations use initial buffering sizes ranging from 2 seconds to 10 seconds.

7.4 Re-buffering strategy

7.4.1 UDP-based applications

Before defining a re-buffering strategy, it is necessary to detect whether re-buffering is needed. Several criteria may be used for deciding when re-buffering is needed.

- When a sufficiently long sequence of packets arrives too late for their play-out time, these packets will all be designated as discarded. In practice, any combination of consecutive lost and discarded packets in sufficient number should cause re-buffering.
- The variable number of consecutive lost or discarded packets can be matched to the specific receivers in use. A default value can be set, following discussion.

- Alternatively, a threshold could be set on average buffer occupation (delay) over a sliding window of packets, and trigger re-buffering when the average falls below a variable threshold (again, with a default value).

When a de-jitter buffer dries out during media assessment, different strategies are possible, depending on the origin of the event.

- If the event occurs as the de-jitter buffer is slowly drying out, this likely corresponds to a slow drift between the sender clock and the receiver clock. This drift should normally be taken into account by the de-jitter buffer service rate. However, this mechanism may fail to prevent the de-jitter buffer to dry out. Also, the bitrate available in the network may be less than the average encoder rate; this produces the same effect. In this situation, there is no way for the de-jitter buffer to refill naturally. Therefore, the solution should be to consider that the first packet observed after de-jitter buffer dry-out is the first packet of a new period, that the buffer has to be considered in an initial state, and that the media assessment is interrupted (i.e., a video freeze or an audio silence).
- If the de-jitter buffer is depleted rapidly, this likely corresponds to a severe degradation of network conditions, producing a combination of consecutive lost packets (packets lost by the network) and discarded packets (packets arriving too late for their play-out time) in sufficient numbers. In this situation, the de-jitter buffer may refill naturally when the period of consecutive lost and discarded packets ends. But the natural refill of the buffer may take some time, during which a dry-out event is more likely to happen. Therefore, in this situation also, the solution could be to consider that the first packet observed after de-jitter buffer dry-out is the first packet of a new period, that the buffer is in an initial state, and that the media assessment is interrupted (corresponding to an unavailable period).

Therefore, in all cases the re-buffering delay chosen may be equal to the initial buffering delay.

7.4.2 OTT applications

In the case of OTT applications, no information is lost at the application layer since TCP compensates for poor network conditions by managing packet retransmissions in order to guarantee loss-free behavior to the application. The side effect of the possible packet retransmissions is that the de-jitter buffer may dry out and generate a re-buffering event.

The re-buffering size may be different than the initial buffering size. Furthermore, if several re-buffering outcomes occur, the sizes of these different re-buffering outcomes may not be identical. Most play-out devices adapt to varying network conditions. If the performance of the network degrades further, it is likely that the player will increase the re-buffering size in order to avoid additional buffer dry-out and re-buffering outcomes.

Annex A

Pseudocode of the state-machine description of the de-jitter buffer model

(This annex forms an integral part of this Recommendation.)

Annex A provides the pseudocode associated to the state-machine describing the behavior of the de-jitter buffer.

A.1 Algorithm overview

The de-jitter buffer receives input packets that have been processed by RTP, HTTP, MPEG-TS, Flash Video (FLV), MP4 and other application layer protocols which identify the locations of media frames in the packets. This information is conveyed to the de-jitter buffer with each packet as a media frame information list. The de-jitter buffer AddPacket is called each time a new packet arrives, and RemoveMediaFrame is called each time a media frame is scheduled to be decoded. In the absence of a model that can call the de-jitter buffer to remove media frames, it can be simply be modeled by a play-out process with the use of a timer, which calls the RemoveMediaFrame function at a configurable, fixed time interval directly associated with the frame rate used for playback in the application area.

The output of the model includes a list of events recorded by the calls to the record event function.

The StopNotification function can be called externally to stop the operation of de-jitter buffer.

A.2 Configuration parameters

Maximum buffer duration: maximum duration of media frames in the de-jitter buffer

Initial buffering duration: minimum duration of media frames in the de-jitter buffer needed before the model enters play state at the start of the session

Re-buffering duration: minimum duration of media frames in the de-jitter buffer needed before the model enters play state after having entered the re-buffering state

Drop buffer duration: minimum duration of media frames in the de-jitter buffer needed before the model enters play state after having entered the missing state

Missing packet wait duration: maximum duration that this buffer model is allowed to spend in the missing state

Blocking mode: boolean needed for applications such as OTT where the buffer model is not allowed to drop frames. When set to 1 and the buffer fills up, the buffer model will not drop frames and will return a failure condition, indicating to the rest of the system that the packet was not added and needs to be reinserted. When set to 0 and the buffer fills up, the buffer model will drop frames and will return a success condition, indicating to the rest of the system that the packet was processed.

A.3 De-jitter buffer variables

Current state: the state of the de-jitter buffer one of: (stopped, playing, initial buffering, re-buffering, missing), initialized to initial buffering

Next DTS: the expected DTS of the next media frame to be played, initialized to the DTS of the first valid media frame received

Time buffered: the cumulative duration of all the frames stored in the de-jitter buffer, initialized to 0

Dropped packet count: the number of packets that have been dropped, initialized to 0

Discarded packets: a list of packets (packet objects) that have been discarded because the buffer was full, initially this list is empty

Buffered packets: a list of packets (packet objects) that have been received but not modeled as played, initially this list is empty

A.4 De-jitter buffer input

Packet: Describes the input of the de-jitter buffer model. This can be called a packet object and it contains the IP packet including a media frame information list. The media frame information list is described as follows:

– If an IP packet does not contain media frame data, the media frame information list is empty. Example media frame information list: {}

– If an IP packet contains part of one media frame, the media frame information list contains one entry composed of (IP packet arrival time, byte offset to the start of media frame data in the IP packet, the size in bytes of this media frame data, the type of media: audio or video, boolean indicating this is a part of a media frame, DTS in milliseconds of the frame). Example media frame information list: {12:01:32.001, 34, 101, video, 0 for partial, 00:00:00.000}

– If an IP packet contains one media frame, the media frame information list contains one entry composed of (IP packet arrival time, byte offset to the start of media frame data in the IP packet, the size in bytes of this media frame data, the type of media: audio or video, boolean indicating this is a complete a media frame, DTS in milliseconds of the media frame). Example media frame information: {12:01:32.001, 34, 101, video, 1 for complete, 00:00:00.000}

– If an IP packet contains multiple media frames then the media frame information list contains one entry for each complete or incomplete media frame, in the order that they appear in the packet. Example media frame information list: { {12:01:32.001, 34, 101, video, 0, 00:00:00.000}, {12:01:32.001, 135, 50, video, 1, 00:00:00.041}, {12:01:32.001, 185,100, video, 0, 00:00:00.083} }

A.5 De-jitter buffer high level function pseudocode

```
Boolean AddPacket(Packet)
        IF (current state == Stopped) THEN
            record event()
            return success
        END IF

        IF (Packet DTS < Next DTS) THEN
            dropped packet count ++
            record event()
            return success
        END IF

        IF (current state == Playing) THEN
            IF (Time Buffered > Maximum Buffer Duration) THEN
                IF (Blocking Mode) THEN
                    return failure
                ELSE
                    dropped packet count ++
                    insert (in DTS order) Packet to Discarded Packets
                    record event()
                    return success
                END IF
            END IF
        END IF
```

```
    insert (in DTS order) Packet to Buffered Packets
    FOR (each Media Frame completed by, or contained in Packet)
      Time Buffered += Media Frame Duration
    END FOR

    IF (current state == Initial Buffering) THEN
      IF (Time Buffered > Initial Buffering Duration) THEN
        current state  = Playing
      END IF
    ELSE IF (current state == Re Buffering) THEN
      head DTS = Get earliest DTS of a complete Media frame in Buffered Packets
      IF (Time Buffered > Re Buffering Duration) THEN
        IF (head DTD <= next DTS) THEN
          current state = Playing
        ELSE
          missing state start time = current time
          current state = Missing
        END IF
      END IF
    ELSE IF (current state == Missing) THEN
      head DTS = Get earliest DTS of a complete Media frame in Buffered Packets
      IF (head DTS <= next DTS) THEN
        current state = Playing
      END IF
      IF ((Time Buffered > Drop Buffer Duration) or
         ((current time - missing state start time) > missing packet wait duration)) THEN
        next DTS = head DTS
        current state = Playing
      END IF
    END IF

    record event()
    return success

StopNotification()
    current state = Stopped

RemoveMediaFrame()
    IF (current state == Stopped) THEN
      record event()
      return
    END IF

    IF (current state == Missing) THEN
      IF ((current time - missing state start time) > missing packet wait duration) THEN
        head DTS = Get earliest DTS of a complete media frame in Buffered Packets
        next DTS = head DTS
        current state = Playing
      END IF
    END IF

    IF (current state == Playing) THEN
      buffered head DTS = Get earliest DTS of a complete media frame in Buffered Packets
      dropped head DTS = Get earliest DTS of media frame in Dropped Packets
      IF (buffered head DTS <= next DTS) THEN
        removed Frame = remove next (in DTS order) complete Media frame from Buffered Packets
```

```
            remove any partial frames with earlier DTS than next DTS from
Buffered Packets
            next DTS   = removed Frame DTS + removed Frame Duration
            Time Buffered -= removed Frame Duration
         ELSE IF (dropped head DTS <= next DTS) THEN
            removed Frame = remove next (in DTS order) Media frame from Dropped
Packets
            next DTS   = removed Frame DTS + removed Frame Duration
         ELSE
            current state = Rebuffering
         END IF
      END IF
      record event()


record event()
      generate a message indicating the current state including:
      -   Current time
      -   Current state
      -   Next DTS
      -   Time Buffered
      -   Dropped packet count
      -   Number of packets in Buffered Packets
      -   Number of packets in Dropped Packets
```

Bibliography

[b-BBF-TR-135] Broadband Forum Technical Report (2007), *Data Model for a TR-069 Enabled STB.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2013