CHANGING THE WAY OPERATORS DELIVER TV

Technical paper

Facilities for de-jittering on X platform

A Functional description to facilitate optimum configuration

Doc. Name : Technical paper on X platform de-jittering
Revision : 2.0
Date : 25.10.2019

Table of Contents

1 Revision History

Ver.	Doc. No.	Date	Comments
1.0		21.10.19	Initial release
2.0		25.10.19	Expansion of PCR_OJ

2 Technical description of X platform de-jittering options with advice for configuration

This paper is structured to allow all types of users to find the information needed to better understand this subject and optimally configure X platform.

Users wishing to gain a basic understanding of why several modes are offered and which is most appropriate for their needs, using default buffer values, need only read section 3.

Advanced users seeking more information on PCR measurement and applicable standards will find this presented in section 4. The paper concludes with sections 5 and 6 which provides a technical description of de-jittering on X platform, with a guide to calculating minimum buffer requirements precisely.

3 What is de-jittering and which de-jittering options should I choose?

De-jittering is required to remove the timing disturbances that inevitably occur when video streams are transmitted over real networks, especially IP networks. These timing disturbances disrupt orderly packet flow and impair the constant bitrate properties of the video stream, making it difficult to process or decode unless the original timing integrity is restored.

All de-jitter processes work by holding packets within a suitably sized buffer to negate the effects of jitter in the network; the buffer enables them to be played out at a constant rate. Many different mechanisms are available to govern how this process works and control how the packets are re-timed. The mechanism must be chosen to optimally match the stream type being de-jittered and there are four primary options available on X platform today. Two of these apply to MPEG transport streams (CBR and PCR de-jitter) and two apply to high-bitrate RTP streams (2022-6 RTP and 2110 mode).

De-jittering MPEG transport streams

PCR De-jittering

When dealing with a simple SPTS (an MPEG_TS carrying one service that has not been MPEG multiplexed with other services) the preferred mechanism to recover the original timing accuracy is to observe PCR. The PCR samples the original MPEG clock that was running within the encoder so using an algorithm to re-time packet delivery based on making PCR time correct will achieve a high degree of de-jittering. A key benefit of this method is that both PCR jitter and network jitter is corrected. Additionally, many services are not encoded at a constant bitrate which means at SPTS level, performing a simple CBR de-jitter to play packets out at a constant rate is not viable: However, PCR samples will always have a linear relationship with time and can be used to re-time CBR or VBR streams. This makes PCR de-jittering an extremely powerful method to use for SPTS's.

It is recommended that PCR de-jitter is used when;

- SERVICE mapping an SPTS from IP input to any output (including decoders)
- SERVICE mapping (cherry picking a single service) from an MPTS input to any output

For the special case where one service is SERVICE mapped from IP input and then output as an SPTS over IP, de-jitter may optionally be set to off. If turned off, the original network jitter will be passed through transparently (as in an IP router).

Note that multiplexing and re-multiplexing processes (creating a new MPTS) are sensitive to jitter because PCR re-stamping will be performed to compensate for MPEG multiplexing delays. If incoming services arrive and are not properly de-jittered (especially those arriving over IP networks with potentially high levels of jitter) the re-stamping process will be inaccurate and will cause the network jitter to translate into PCR accuracy jitter.

One frequent mistake that customers often make in PCR de-jitter mode is simply to set the de-jitter buffer size to correlate with the maximum jitter level observed over the network. This is incorrect because it fails to account for the complexities of using PCR to de-jitter. For example, the buffer must be large enough to accommodate at least 2 x PCR samples so there are important considerations that often require the buffer to be many times larger than the actual incoming jitter levels suggest (please see section 6). Setting too small a buffer size in PCR de-jitter mode is a quick way to invite problems since a de-jitter algorithm without stable PCR references to sample will become un-locked leading to serious traffic bursting problems. When using PCR de-jitter, please be sure not to select values below default unless you are familiar with the manual calculation methods presented in section 6.

CBR De-jittering

When dealing with an MPTS that is being transparently mapped through a system the preferred method to use is CBR de-jittering. This method is preferred whenever PCR is not accessible, as is the case with MPTS streams, or when the lowest throughput latency is required.
MPTS streams always run at a constant bitrate and the CBR de-jitter algorithm simply deduces what this rate is and ensures that incoming packets are clocked out at this constant rate to eliminate network jitter.

CBR de-jittering offers a good solution when;

- Mapping an MPTS from IP input to any output (ASI, DVB-S2, IP)
- Mapping an SPTS containing NULL packets (which ensures it has a constant bitrate) as an alternative to using PCR de-jittering.

CBR de-jittering will also work with SPTS's although since PCR is not observed and is not corrected, it is not preferred compared with PCR de-jittering and can only be considered for SPTS's with constant total bitrate. The only potential advantage of considering CBR de-jitter for SPTS's is related to the simplicity of this method; Since PCR is not involved, it is not necessary to accrue several PCR samples within the buffer meaning that smaller buffer sizes can often be set, achieving the lowest possible latency.

Note that when TS mapping an MPTS from IP input to IP output, it is possible to set CBR de-jittering to OFF. This mode adds no additional buffer and therefore no extra latency but any input jitter will be passed transparently through the platform un-corrected (as in an IP router).

De-jittering RTP HBR streams

SMPTE 2022-6 RTP de-jittering

This de-jittering method is applicable to high bitrate video that has been natively encapsulated into RTP packets such as un-compressed video and TICO compressed video using SMPTE

2022-6 encapsulation. Note this method is not applicable for JPEG2K because this is MPEG TS based.
SMPTE 2022-6 packets are created as CBR and are encapsulated within RTP packets which are timestamped. This de-jitter mode uses the RTP time stamps to play packets out from the buffer at a consistent rate to achieve de-jittering.

SMPTE 2110 de-jittering

For the newer SMPTE 2110 standard, packets are optionally referenced to PTP. As part of Appear TV's growing support for this standard, a SMPTE 2110 de-jitter option will be introduced that enables clock recovery and hence de-jittering to be performed using PTP.

Recommended de-jitter buffer settings for typical applications

Calculating the minimum de-jitter buffer size for an individual application can be relatively complex, especially for PCR de-jitter applications, and is only necessary for applications requiring the lowest possible latency. For the majority of applications it is sufficient to simply set sensible default values that will work in most circumstances. The table below has been provided as a guide based on scenarios with relatively high network jitter.

Service type	De-jitter method	Recommended buffer	Comments
Low bitrate Radio service	PCR	**>=250ms**	0 to 1500ms possible
MPEG DTH Video service (2 -8Mb/s)	PCR	**>=150ms**	0 to 1500ms possible
MPEG high bitrate distribution service (JPEG2K, 100Mb/s)	PCR	**>=85ms**	0 to 1500ms possible
MPEG MTPS	CBR	**>=150ms**	0 to 1500ms possible
Uncompressed video or TICO with SMPTE 2022-6 encapsulation	2022-6 (RTP)	**>=150ms**	0 to 1500ms possible Do not use this option with MPEG_TS streams
Uncompressed video or TICO with SMPTE 2110 encapsulation	2110	**>=4ms**	0 to 50ms possible Do not use this option with MPEG_TS streams

Note: If ensuring lowest latency is important for your application, please see section 6 which covers the manual calculation of smallest buffer size.

4 Understanding Jitter in detail

Network jitter

Network packet jitter is defined as the variation in transport delay between the source of the network flow and the device receiving the flow. Both native RTP encapsulated and MPEG-TS traffic can be adversely affected.

The primary causes of network jitter include;

- Network congestion caused by competing traffic in the network
- Buffers with variable delay within the network, typically present in switches/routers
- Re-routing or redundancy switching within the network

Additionally, MPEG_TS traffic can be negatively affected by the framing process where up to 7 MPEG_TS packets are encapsulated into an IP frame before transmission.

Network jitter should conform to the maximum specified by DVB and SMPTE. The following standards specify maximum permitted jitter levels;

- **ETSI TS 102 034:** DVB Transport of MPEG-2 TS Based DVB Services over IP Based Networks. Section 7.2.1.1 Packet Jitter specifies the maximum allowed packet jitter is no more than 40ms (peak-to-peak)
- **SMPTE 2022-2:** Unidirectional Transport of Constant Bit Rate MPEG-2 Transport Streams over IP Networks. Section B.2 (Jitter Tolerance) specifies network jitter can be absorbed by buffering at the receiver. The aim is to provide a “jitter budget” buffer of 120ms. This buffer should run at a mean occupancy of 50% providing a 60ms latency.

Network jitter is assessed by analyzing the packet to packet jitter, measured as the packet inter-arrival time IAT (e.g. tcpdump -i ni0 src 10.10.82.144 and udp port 3021 -ttt -c 1000 | sort)

Appear TV’s X platform now includes the ability to include test packets to obtain various real-time statistics concerning distribution network performance. The integrated test generator creates ‘smart’ TS null packets containing accurate timestamps which can be used by another Appear TV chassis receiving the IP traffic to track the difference between the maximum and minimum transport delay. The maximum difference is an accurate measurement of the network jitter.

PCR basics and PCR Jitter

When MPEG transport streams are being used, the integrity of the PCR mechanism which synchronize the encoder / decoder clocks must be preserved.
The MPEG processing ecosystem (with encoder / multiplexer / decoder) is designed to process video and audio presented to it from a source at a given rate. The encoder clock is typically locked to the incoming SDI rate using a phase locked loop. The SDI playout source will deliver video frames at a frame rate precisely governed by its internal (or externally referenced) system clock, but as with any clock it will not be totally accurate and will contain a percentage of frequency offset and drift.
The next item in the chain that ultimately needs to be locked to this clock is the decoder at the end of the MPEG chain. If the decoder is correctly locked, it will avoid the need to drop / repeat frames and will present the video at precisely the same rate it was delivered by the video source. This is only achievable if a mechanism exists to convey the actual rate of the 27MHz MPEG clock running in the encoder to the decoder, and this is achieved using PCR timestamps. Each PCR provides a ‘snapshot’ of a counter, driven by the program source clock and inserted into packets within the TS at regular intervals (typically ~35ms). There are two options for transmitting the PCR; it can either be transmitted as a separate PID or, more usually in new systems, carried within the video PID (or audio PID for radio systems). At the receiver, these

PCRs are recovered from the TS and the counter values are compared with a similar counter driven from a locally adjustable 27 MHz clock in the decoder. Any mismatch between these counters is used to slowly adjust and lock the local 27MHz clock in the decoder to the original source clock.

PCR tolerance is specified by MPEG (ISO/IEC 13818-1 subclause 2.4.2.2): *"The PCR tolerance is defined as the maximum inaccuracy allowed in received PCRs. This inaccuracy may be due to imprecision in the PCR values or to PCR modification during re-multiplexing. It does not include errors in packet arrival time due to network jitter or other causes. The PCR tolerance is ±500ns".*

Whenever network jitter is present (MPEG transport delay is not constant and significant jitter can accumulate with transmission over IP especially), it is necessary to place a jitter-removing element (de-jitter stage) between the network and an MPEG-2 decoder. De-jittering is achieved by controlling packet flow through a buffer (the de-jitter buffer) in a controlled way.

If errors in packet arrival time are not removed prior to MPEG re-multiplexing, the network jitter will be transformed into PCR inaccuracy in the multiplexer causing the PCR instability to become permanently propagated through the MPEG chain. This will prevent the decoder from being able to recover a stable clock reference leading to potential decoding issues. Generally, the MPEG delivery chain must be designed to ensure the PCR jitter tolerance is less than +/-500ns for packets entering the MPEG decoder.

The measurement of PCR jitter within a transport stream is specified in ETSI TR 101 290 V1.3.1 (2014.07).The following measurements are defined;

- PCR frequency offset (**PCR_FO**) – represents the frequency offset of the 27 MHz clock contained within the PCRs, measured against a stable external reference. From the MPEG specification (ISO 13818-1), the frequency offset is required to be within 27MHz ± 810Hz (27MHz ± 30ppm).

- PCR drift rate (**PCR_DR**) – represents the rate at which low frequency changes of the 27 MHz clock contained within the PCRs occur, measured against a stable external reference – therefore including any transmission timing changes. From the MPEG specification (ISO 13818-1), the rate of change of the clock frequency shall be less than 75 mHz/s (2.77 ppb/s).

- PCR accuracy (**PCR_AC**) – represents inaccuracies in the 27 MHz clock contained within the received PCRs, but does not include any transmission timing impairments. Measurement is made using estimated PCR arrival times calculated from their byte position within the constant bitrate transport streams. The PCR_AC measurement is valid for both real time and off-line measurements, **but can only be measured with constant bit rate streams containing MPEG NULL packets.** From the MPEG specification (ISO 13818-1), the PCR accuracy is required to be within +/-500ns.

- PCR overall jitter (**PCR_OJ**) – represents inaccuracies in the 27 MHz clock contained within the received PCRs, including cumulative PCR accuracy errors and any transmission timing impairments (jitter). Measurement is made using PCR arrival times compared to an external reference. The PCR_OJ measurement is only considered to be valid for real time measurements (or recordings including actual arrival times, such as PCAP recordings). Since the MPEG (ISO 13818-1) specification does not include jitter, PCR_OJ can only be compared against the +/-500ns maximum specification if the transmission jitter is assumed to be zero (so not valid for IP network measurement).

ETR TR 101 290 states that PCR Accuracy shall be measured for every received PCR by comparing the time between two PCR values, PCR(i') and PCR(i"), with the time calculated from the number of bytes between the two PCRs and the constant bitrate of the TS. This means that the mux clock (that was used to generate the constant bitrate) is used as a timing reference when measuring the accuracy of the PCRs. This allows PCR accuracy to be measured independently of any transport jitter effects, since it is essential to distinguish between these, and makes this measurement totally immune to transport delay.

When making PCR measurements involving Appear TV X platform, it is important to know how the various modules and stages lock and treat PCR.
The video source clock in an Appear TV encoder is phase locked to the incoming video SDI clock. PCR values are inserted into the TS by the encoder according to MPEG rules. Since the encoders lock to the SDI source clock, any frequency offset and drift present within the SDI source domain will be transparently forwarded into the system.

When making PCR measurements involving Appear TV Encoders (SDI input, IP output):

- The PCR frequency offset (PCR_FO) will be transparent from the SDI source
- The PCR drift rate (PCR_DR) will be transparent from the SDI source
- The PCR accuracy (PCR_AC) will typically be less than 100ns

When ingesting IP input sources, the Appear TV IP de-jitter algorithm will remove any IP transport jitter and at the same time track the offset and drift of the original SDI clock source using the program PCR. Any incoming PCR inaccuracy will efficiently be removed by this IP de-jitter function.

When taking PCR measurements from Appear IP Gateways (IP input, IP output) where a defined CBR output rate has been set:

- The PCR frequency offset (PCR_FO) will be transparent from the SDI source
- The PCR drift rate (PCR_DR) will typically be less than 30 ppb/s
- The PCR accuracy (PCR_AC) will typically be less than 100ns

5 How X platform de-jitters: An algorithmic description

X platform is designed to handle both RTP and MPEG_TS streams and is provided with a comprehensive toolkit to manage all scenarios optimally. This toolkit comprises the following modes;

CBR de-Jitter mode

The CBR de-jitter mode provides the best option for de-jittering multiplexed transport streams (MPTS) although it can be used for constant bitrate or null-padded SPTS's as well if achieving the lowest possible delay is a high priority.

This method;

- Works with constant bitrate (CBR) inputs flows. It is important to discriminate between the CBR video PID case, where an encoder is running at a constant bitrate, and what s being referred to here. In this case, the total transport stream rate is the important factor. As long as the final transport stream rate is constant, and is being padded with null packets, the CBR/ VBR status of the encoder is not important. CBR de-jitter will only function with CBR transport streams padded with NULL packets.

- Will remove IP network jitter

- Is applicable to DVB, MPEG and ATSC transport streams

- Provides low latency (in the order of the network jitter)

The CBR de-jitter option works by measuring the constant input bitrate of the flow; This measured bitrate is used as a reference for governing the orderly playout bitrate from the de-jitter buffer. This causes packets to be played out of the de-jitter buffer at the same intervals as the packets were originally were played out into the IP network, thereby removing any variable delays in the IP network.

A control loop within the Appear TV IP packet receive stage determines what the constant bitrate value should be and slowly adjusts the playout bitrate to match. This ensures the flow rate into and out of the de-jitter buffer is identical over time with the jitter effectively nullified.

Note: The first stage of the CBR de-jitter process is to measure the average input bit rate. Getting a good estimation of the CBR rate will take longer if the input has high jitter. Packets will be playout out without jitter compensation during this estimation stage.

Note: Any PCR jitter (PCR inaccuracy) present in the TS before the IP network will also be present after the CBR de-jittering.

CBR de-jitter mode alarms are being added to X platform at the time of writing.

PCR de-Jitter mode

PCR de-jitter mode is suitable for SPTS MPEG transport streams only since it acts at PCR level. This makes it applicable for;

- SPTS DVB, MPEG and ATSC input flows

- Validity also encompasses JPEG2K which is transmitted as an MPEG_TS

- Works with both variable bitrate (VBR) and constant bitrate (CBR) inputs streams.

- Will remove IP network jitter

- Will remove PCR inaccuracy

- Requires that a valid PCR is present and signaled in PMT

- Provides medium latency (see detailed buffer calculations below)

When individual services are selected (Service mapping mode), X platform will route every service as a separate variable bitrate SPTS towards the output module. The PCR de-jitter algorithm is built to work with such single program variable bitrate transport streams. Packets are played out of the de-jitter buffer as indicated by the PCR timestamps. This process ensures every packet is adjusted in time to match the timing indicated by the PCR and therefore removes both network jitter and PCR inaccuracy from the incoming stream.
Central to the packet timing adjustment is a control loop within the Appear TV algorithm that tracks the PCR source clock and slowly adjusts the recovered clock to match the frequency of the PCR source. This ensures that the flow rate into and out of the de-jitter buffer is identical over time.

The PCR de-jitter algorithm is quite complex and is performed in several stages:

1. The PMT of the service will be inspected to find the PCR pid for the service. The PCR PID will be used by the de-jitter algorithm to recover the original packet timing.

2. By definition, the interval between two PCR packets will mark a constant bitrate interval. The de-jitter algorithm will calculate the constant bitrate interval between any two PCR packets by finding the length of the interval (from the PCR values) and by counting the number of bits (TS-packets) transmitted during this interval.

3. A control loop will recover the original PCR clock frequency. From the MPEG specification (ISO 13818-1), the rate of change of the clock frequency shall be less than 75 mHz/s (2.77 ppb/s) or 10 ppm/hour. Because of this, the allowed rate of change of the control loop is low, so that the recovered PCR clock falls within the drift limits of the MPEG specification.

4. A consequence of the slow drift of the recovered PCR clock is that the buffer filling may vary depending on the current difference between original PCR clock and the recovered PCR clock. For example, If the source has abrupt changes in PCR clock frequency, the PCR de-jitter buffer will slowly increase or decrease while the average PCR clock is recovered.

Note: Since every TS packet is moved according to the timestamps indicated in the PCR, any PCR inaccuracy from the original source will turn into small inaccuracies in the variable bitrate when played out from the de-jitter buffer. In this process, PCR inaccuracies will be removed.

Note: The PCR de-jitter function is built to intentionally introduce PCR drift during the first 5-15 minutes after stream initialization, or when a sudden change is observed in the incoming PCR. This PCR drift is introduced in order to quickly recover the video source clock. Since the frequency offset of the video source is allowed to be 30ppm, and the max drift is specified as 10 ppm/hour, it would take up to 3 hours to recover the video source clock without violating the specifications. Since this may cause initial buffer buildups (with added latency), the Appear equipment will intentionally introduce PCR drift in the order of 30 ppb/s during initial video clock recovery, reducing the worst case recovery time from 3 hours to ~15 minutes.

The PCR de-jitter function is monitored by alarms but these only become active once the algorithm has locked. A failing system will generate alarms but a de-jitter process that has not successfully initialized will not.

RTP de-Jitter mode

The RTP de-jitter option is only applicable for RTP flows and can be implemented with 2022-6 HBR streams only. It will remove IP network jitter and offers low latency (in the order of the network jitter)

RTP de-jitter works by extracting the transmit timestamps in the RTP headers and reuses these timestamps as playout times from the de-jitter buffer. This will cause packets to be played out of the de-jitter buffer at the same intervals as the packets originally were played out from the IP transmitter into the IP network, thereby removing any variable delays in the IP network.

A control loop on the IP receiver will track the RTP source clock and slowly adjusts the recovered RTP clock to match the frequency of the source. This ensures the flow rate into and out of the de-jitter buffer is identical over time.

The resolution of the RTP timestamps is application specific, and at the time of writing, RTP de-jitter is currently only implemented for 2022-6 streams.

Note: For MPEG transport streams the RTP timestamp accuracy is 11us due to the 90kHz resolution of RTP timestamps for MPEG transport. Since the jitter tolerance for transport streams is 500ns, ***the RTP de-jitter option is not suitable for MPEG TS.***

Note: For 2110 transport, the 2110 receivers (decoders) have a configurable de-jitter buffer built into the 2110 receiver/decoder module. This buffer must be set large enough to account for all network jitter. Clock recovery for 2110 is performed using PTP.

RTP de-jitter mode is not currently monitored by alarms although it is planned to introduce this in future.

6 Manual calculation of de-jitter buffer size

Most customers simply set the de-jitter buffer size to a large, safe value and therefore avoid the need to perform actual occupancy calculations. However, X platform allows custom buffer values to be set so the buffer size can be precisely matched to actual requirements, to minimize latency, when this is desired.
This section provides guidance for customers needing to closely reconcile buffer values for this reason.

In general, when de-jittering is initiated, the de-jitter buffer will fill up with packets to the configured value regardless of the de-jittering mode; this causes the latency to match the configured value of the buffer. Any short term delays in packet transmission will drain the buffer, while short term bursts will increase it. Over time, the mean buffer occupancy is maintained at the configured size.

Calculating HBR 2022-6 buffer requirements is simple, but the analysis required for CBR and PCR de-jitter modes is a little more complex;

Calculating the minimum buffer size in HBR 2022-6 de-Jitter mode

For HBR streams with 2022-6 (RTP) de-jitter, the recommendation is simple: The buffers should be set to a size that matches the worst case packet jitter in the network.

Calculating the minimum buffer size in CBR de-Jitter mode

The minimum buffer size that can be successfully used to perform CBR de/jitter depends on the following factors:

1. **Network jitter**. The buffer needs to compensate for the variance in transport delay in the network.

2. **TS-per-UDP jitter**. This delay component will only be present at low bitrates with a high number of TS-per-UDP. For example, a low bitrate (radio service) MPTS running at 1Mbps with 7 TS per IP will have a packetization induced jitter of 10ms. At10Mbps, the packetization effect becomes less significant and is reduced to only 1ms. For higher bitrates and / or a lower TS per IP ratio, the packetization effect can be neglected.

3. **CBR clock drift**. The CBR de-jitter algorithm will lock to the original CBR bitrate from the MPTS multiplexer. If CBR bitrate drift is present, some short time margin is needed while the de-jitter buffer slowly adapts to the changes in CBR bitrate.

4. **Packet loss**. Packet loss directly affects the filling of the de-jitter buffer, and if severe can force the algorithm to restart.

Calculating the minimum buffer size in PCR de-Jitter mode

Calculating the minimum buffer size for PCR de-jittering can be complex. The safest choice may be to choose a relatively high buffer size to be sure that the buffer can handle any kind of jitter, but if tuning is required to minimize latency the following factors must be considered;

1. **Network jitter**. The buffer needs to compensate for the variance in transport delay in the network.
2. **TS-per-UDP jitter**. The importance of this component depends on the bitrate and the number of TS-per-UDP. For a system with 1 TS-per-UDP this factor is 0 and can be ignored. However at a more commonly used 7 TS-per-UDP and with a low bitrate radio service at 100kbps, the resulting jitter will be greater than 100ms.

3. **Maximum PCR interval**. The PCR de-jitter algorithm requires at least two PCR timestamps to be available within the PCR de-jitter buffer so that the correct constant bitrate can be calculated for that PCR interval. In order to be resilient to packet loss (lost PCR timestamps), it is strongly recommended to keep at least three PCR timestamps in the buffer at all times, so this factor should be set to two PCR intervals.

4. **PCR clock drift**. The PCR de-jitter algorithm will lock to the original PCR clock. While the control loop locks to the PCR frequency there will be a period when the recovered clock frequency is higher or lower than the original PCR clock. In this period the buffer filling will follow the slow oscillations of the recovered clock. The offset will depend on incoming jitter, PCR clock frequency accuracy and drift. For the maximum legal change of input PCR clock frequency from +30ppm to -30ppm frequency offset the regulator requires an extra 30ms buffer but for typical MPEG encoders with frequency offset of <5ppm, the regulator will require <10ms.

5. **Transport delay changes**. Packet loss or other sudden changes in transport delay will directly affect the filling in the PCR de-jitter buffer and must be accomodated.

The required PCR dejitter buffer size is the sum of all of the factors listed above.

End