WHITE PAPER

Latency in live network video surveillance

June 2024

AXIS® COMMUNICATIONS

Summary

The process of live streaming in IP video surveillance consists of capturing video in the camera, packaging and transporting it through the network, and unpacking it in the receiver to display it. Each of these steps add more or less latency.

- **Latency introduced on the camera side.** Capturing of video is followed by image processing, compression, and packaging. Each step introduces some latency, but overall, the processes within the camera contributes to only a small fraction of the total end-to-end latency.
- **Latency introduced by the network.** This can be very big or very small, and is the most unpredictable factor in the end-to-end latency "equation". You can make it more predictable by investing in good network software and hardware. Network latency depends very much on the data-to-bandwidth ratio. You can configure the camera to reduce the amount of data it generates, hence reducing the amount of packets that need to be sent over the network.
- **Latency introduced on the client side.** On the client side, data is received and buffered to be sorted and queued out to the graphics card and monitor. The latency is most affected, even up to several seconds, by the receiving buffer in the client. Without a big buffer there is a risk that the video stream will not be played evenly.

To reduce latency is always a question of cost. The biggest wins can be achieved by improving the network and the client-side hardware and software.

Table of Contents

1 Introduction

In the video surveillance context, latency is the time between the instant a frame is captured and the instant that the same frame is displayed. This is also called end-to-end latency or sensor-to-screen latency. The process of transporting a frame from the camera sensor to the display monitor involves a long pipeline of steps.

This white paper outlines the various steps that contribute to the total latency. It also provides recommendations for how latency can be reduced.

2 What is latency?

The definition of latency depends on the context. In network technology, latency is commonly perceived as the delay between the time a piece of information is sent from the source and the time the same piece of information is received at its final destination.

In this paper we discuss latency in network video surveillance systems. Here we define latency as the delay between the time an image is captured by a camera and the time it is visible on a video display. During that period of time the image is captured, compressed, transmitted, decompressed, and displayed. Every step adds its own share of delay to the total delay. For simplicity, the end-to-end latency can be seen to consist of three major stages:

- Latency introduced by the camera (image processing latency, compression latency)
- Latency introduced by the network (transmission latency)
- Latency introduced by the receiver side (client buffer, decompression latency, display latency)

To meet the latency goal of a video surveillance system, each of these latencies must be considered during the designing of the video solution.

3 How do we measure latency?

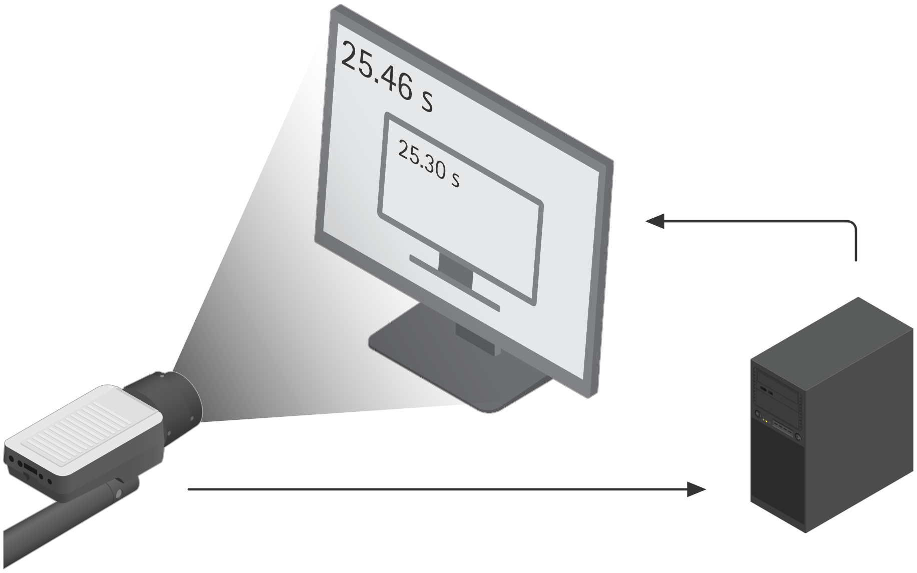
Latency is usually expressed in time units, typically seconds or milliseconds (ms). It is very hard to measure exact latency as this would require the clocks on the camera and the display device to be synchronized exactly. One simple way (with reservation for minimum deviation from the exact values) is by using the timestamp overlay text feature. This method measures the end-to-end latency of a video surveillance system, that is, the time difference between the capture of one image frame in the lens to when that same frame is rendered on a monitoring device.

Note that this method will produce a possible error of up to one frame interval. This depends on the fact that the timestamps used to calculate the latency are only collected at frame capture. We will therefore only be able to compute the latency with the factor of the frame rate. Hence, if we have a frame rate of 25 fps, we can calculate the latency as a multiple of 40 ms. If we have a frame rate of 1 fps, we can calculate the latency as a multiple of seconds. This method is therefore not recommended for low frame rates.

To measure latency using the timestamp overlay text feature:

1. Turn on timestamp in the overlay by using (%T:%f)
2. Place the camera at an angle so it captures its own live stream output

3. Take snapshots of the live stream output to compare the time difference between the time displayed in the original text overlay and the time displayed in the screen loop



An example of latency measurement using the timestamp overlay text feature. We can spot a time difference of 25.46–25.30=0.16 seconds or 160 ms. This means that the end-to-end latency is 160 ms.

4 What affects latency?

The total latency is the sum of latencies introduced by the camera, by the network, and by the client side.

4.1 Latency introduced by the camera

Each frame takes a time gap of roughly 1/30 s exposure, followed by a short time to scale and encode the image. The encoded image is chopped up and packaged, and one image is outputted onto the network every 33 ms. The time it takes for this process in the camera can be under 50 ms, but is more commonly a few hundred ms. It varies slightly depending on the camera (PTZ excluded) and on whether the frame is an I-frame or a P-frame.

4.1.1 Capture latency

Let us take a look inside the video camera. The camera's image sensor is made up of millions of photodetectors (photo-sensitive spots), known as pixels. The sensor captures light in its pixels throughout one exposure interval, before converting the registered light energy into electronic signals. Then the pixels are empty and ready for another exposure. The number of exposures that the sensor delivers per time unit, that is, how many frames the camera can capture per second, defines the sensor's capture rate.

The capture latency depends on the capture rate. If you set the capture rate to 30 fps, meaning that the sensor will capture one image every 1/30th of a second, the capture latency will be up to 33.3 ms.

4.1.2 Image processing latency

Each captured frame goes through a pipeline of image processing steps, such as de-interlacing, scaling, and image rotation, which add latency. The more processing, the more latency introduced by the camera. But since the processing in the camera affects how much data is produced, the amount of processing also leads to effects on the network latency when the data is transferred over the network.

Some parameters that affect latency are:

- **Image rotation**. Rotation of the video stream by 90 or 270 degrees adds an additional load to the encoding processor. Pixels will have to be rearranged and buffered and this causes some delay.
- **Resolution**. Higher resolution means more pixels for the processor to encode. The increase in processing time for a higher resolution compared to a lower resolution is usually insignificant because it is balanced by a faster processing unit in high resolution cameras. But higher resolution does result in more data per frame and thus more packets to be transmitted. In a network with limited bandwidth this might lead to delay during transmission. This, in turn, will lead to the need of a larger buffer at the receiver side, causing longer latency.
- **Noise filtering**. Advanced noise filtering requires buffering of multiple frames, which introduces additional latency.
- **Privacy masking**. Advanced privacy masking features, such as AXIS Live Privacy Shield, can introduce additional latency. This is because of the buffering that is required in order to ensure that the correct privacy masks are applied at the right time.

4.1.3 Compression latency

Video is encoded in order to compress the data before transferring it. Compression involves one or several mathematical algorithms that remove image data. This takes more or less time depending on the amount of data to process. The compression latency introduced in this step is affected by several aspects of the compression:

- **Complexity of compression algorithms**

  H.264 and H.265 are more advanced compression methods than MJPEG is. However, Axis cameras typically have a higher capacity for H.264 and H.265 compression, compared with MJPEG compression, which means that compression latency with H.264 or H.265 is not necessarily higher. On the other hand, it can be higher on the decoding site. H.264 and H.265 data streams produced by Axis cameras require the decoder to buffer at least one frame, while MJPEG decoding requires no buffer. Furthermore, Zipstream storage profile adds up to two frames of additional latency, that is, 66.7 ms for a 30 fps video.

- **Effectiveness of the compression method**

  The most common encoding schemes used in Axis cameras are MJPEG, H.264, and H.265. They all introduce latency in the camera. H.264 and H.265 minimize the video throughput more than MJPEG does, which means that there will be fewer data packets to send through the network, unpack, and render in the receiver end. This will reduce the total latency.

- **Choice of bitrate**

  Video compression reduces video data size. However, not all frames will be the same size after compression. The compressed data size can vary depending on the scene. In other words, the original compressed data consists of streams of variable bitrate (VBR), which result in variable bitrate being outputted into the network. You need to consider the constraints of the available network, such as bandwidth limitations. The bandwidth limitations of a streaming video system usually require regulation of the transmission bitrate. In some encoders, you can choose between VBR and maximum bitrate (MBR). By choosing MBR you are guaranteed that the network receives a limited amount of data. By avoiding to overload the network, you reduce network delay and the need of a larger buffer in the receiver end of the system.

In Axis cameras, H.264 and H.265 encoders provide the choice to select VBR and MBR. However, we generally recommend using VBR with networked video where the quality is adapted to scene content in real time. It is not recommended to always use MBR as a general storage reduction tool or fix for weak network connections, since cameras delivering MBR video may be forced to erase important forensic details in critical situations.

Using Zipstream in the camera reduces the bitrate. This keeps the amount of data down and thereby reduces latency in the network.

- **Number of streams**. If more than one type of stream is requested from the camera (different frame rates or resolutions), the processing of the additional type of stream will add latency because all streams must be encoded by the same processor.

When choosing a compression method you should take all these aspects into consideration. On the one hand, an advanced encoding algorithm can take a longer time to encode and decode but on the other hand, it will reduce the data volume being sent through the internet, which will in turn shorten transition delays and reduce the size of the receiver buffer.

4.1.4 Buffer latency

Because video is handled one frame at a time, only a limited amount of data can be compressed at once. Short-term buffers are sometimes needed between the processing stages, contributing to the latency in the camera.

4.1.5 Audio latency

In some cases the video stream is accompanied by audio. The audio encoder needs to wait for a certain amount of samples before a block is available to begin the encoding of audio, and this adds delay on the camera side. The sample rates and block sizes are different in different audio encoding algorithms.

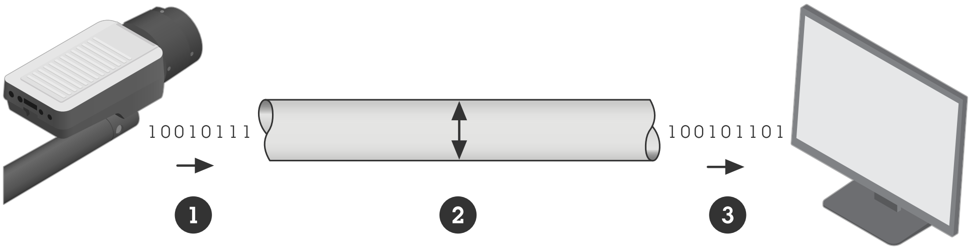
4.2 Latency in the network

After the image is captured, processed, and compressed, the video data will travel through a network before it reaches the client side for rendering. To understand how the network will affect latency we need to first understand some basic concepts in video networking: bitrate, bandwidth, and throughput. Network latency is proportional to bitrate and inversely proportional to bandwidth.

- **Bitrate** is the amount of data, measured in bits, that is processed per unit of time. In video surveillance, bitrate is defined by the amount of camera-generated data to send through the network per unit of time. The bitrate depends on many factors; it depends very much on the filmed scene, the processing done in the camera, and the video stream settings. When the camera is producing more data to be transmitted, you can expect higher network latency if the bandwidth is limited.

- **Bandwidth** is how much data the network between the camera and the monitor can potentially handle. It is the maximum capability of your link. The bandwidth depends on the length and the infrastructure of the link, that is, switches, routers, cables, proxies, and more. If we increase the capacity of the network, more data will be able to pass through, leading to lower latency.

- **Throughput** is the actual achieved speed (in bits/s) of your data transfer. It depends on if you are sharing the link with others and on the electromagnetic interference of the cables in the link. The throughput may also be capped by the QoS (quality of service) settings configured on the ports.



If we compared the link (the network) between the camera and the monitor to a pipe, the bitrate (1) would be how much data is being carried out to the pipe per time unit, the bandwidth (2) would be the thickness of the pipe, and the throughput (3) would be a measure of much data actually comes through the pipe per time unit.

1 *Bitrate*
2 *Bandwidth*
3 *Throughput*

The total latency in the network depends on three major factors: the infrastructure of the link between the camera and the video viewing device (which determines the bandwidth), the amount of data produced by the camera (which determines the bitrate), and the choice of transmission protocol.

4.2.1 The infrastructure

The network is the most unpredictable source of the end-to-end latency. Switches, routers, cables, proxies: everything in the network between sender and receiver affects the total latency. In a LAN, the latency in the network could be only a few ms, which is insignificantly low and can be ignored. However, if the video stream is to be transmitted over the internet with unspecified routes, the network latency will be hard to predict and could in many cases be the main contributor to the end-to-end latency.

With careful network management and bandwidth allocation, the unpredictable factors of the network latency can become more predictable. The link between the camera and viewing device needs to have a guaranteed throughput.

In a LAN, this can be done by making sure there are as few hops as possible in the link. The link should not by shared with other traffic such as voice over IP (VoIP) or other protocols that will be prioritized over video by default, or other demanding services that will overload the link.

If the link is over a wide area network (WAN), the QoS needs to be guaranteed in each hop, that is, in the routers and switches. This can also be accomplished by leasing a point-to-point route through your local internet provider.

Configurable factors that affect the throughput:

- Overhead of the package (protocol-dependent example: VLAN header)
- Proxies and firewalls
- QoS of each link in the whole route
- Burst mode or not (enabled —> higher speed)
- MTU - the size of the video payload

Cost-related factors that affect the throughput:

- Processor speed and port buffer of the switches and routers
- Type of cable (or wireless)

4.2.2 The amount of video stream data

The choice of image processing and compression method in the camera affects the network latency, since these choices affect the amount of video data being produced. Sending a smaller amount of data will clearly take less time.

4.2.3 The transmission protocols

The video frames from the camera are passed on to a transport protocol application, usually RTP or HTTP. Transmission to the rendering client is done over an IP network. The transmission is either through reliable TCP, which is a connection-oriented protocol with re-transmission for lost data packets, or through UDP, which is a simpler protocol that does not guarantee delivery and provides no facility for retransmission of lost packets.

With Axis cameras, you have the following options when you encapsulate the encoded data stream for transmission:

Table 4.1 Options for encapsulating the encoded data stream.

Topology	Recommended Axis video packets encapsulation modes
LAN / fewer hops and directly managed nodes	MJPEG / HTTP / TCP
LAN / fewer hops and directly managed nodes	H.264, H.265, or MJPEG / RTP / RTSP / HTTP / TCP
LAN / fewer hops and directly managed nodes	H.264, H.265, or MJPEG / RTP / RTSP / TCP
WAN / several hops where you do not have full control over the nodes	H.264, H.265, or MJPEG / RTP / Unicast / UDP
WAN / several hops where you do not have full control over the nodes	H.264, H.265, or MJPEG / RTP / Multicast / UDP
Remote connection / cloud / WAN / several hops where you do not have full control over the nodes	H.264, H.265, or MJPEG / RTP / WebRTC / UDP or TCP

Normally it takes longer to transport a packet through TCP than through UDP because of the extra connection setup, the acknowledgement messages, and the re-transition of packages when a loss is detected. On the other hand, with UDP, the user will experience artefacts or interruption in the video stream when packets are lost on the way. TCP will yield jitter on packet loss, while UDP will yield artefacts and/or interruptions on packet loss. If data loss and temporary quality degradation is acceptable, UDP could be a choice for networks with low bandwidth.

If you use TCP, there will be more packets to be sent, and to support this you need a larger bandwidth. If you know there is a lot of congestion in the network, you should select UDP as your transmission protocol. Since packet loss is accepted, at the same time it will also lead to packet loss resulting in lower image quality.

With WebRTC and adaptive bitrate, the video will be adapted to the network to avoid uncontrollable latency spikes such that can occur with TCP.

4.3 Latency on the client side

After the video is received on the client side of the video system, it is unpacked, reordered and decoded, and a media player is used to render the video. Each step contributes to the total latency generated on the client side, and the computer itself plays an important role. The CPU capacity, the operating system, the network card, and the graphic card affect the latency. Usually MJPEG is the method with lowest decoding and display latency because data can be drawn on screen as it arrives without time codes. H.264 and other video compression standards assign time codes to each picture and require them to be rendered accordingly.

4.3.1 Play-out buffer

Real networks are often very large and complicated, with bursting traffic behavior and packets arriving in different order. To compensate for variations introduced by network transport, a buffer is used on the client side. Known as play-out buffer or jitter buffer, this buffer makes sure that the packets get into the right order and buffers enough data so the decoder does not "starve"; uniform frame rate is displayed in the viewer.

This buffer contributes to relatively high latency. Different viewer applications have different play-out buffer size, and in most viewers you can change it. But it is important to keep in mind that reducing the buffer will increase jitter, so you need to find a balance between jitter and acceptable latency.

4.3.2 Audio buffer

In playback, audio streaming is more sensitive to hiccups or delays than video streaming is. A single delayed audio packet generates an annoying crack in the soundtrack, and the audio has to be lip-synchronized with the video. For these reasons, you need to set up a large play-out buffer when video is accompanied with audio, and this, of course, increases the latency.

4.3.3 Decompression

The time required for the decompression process depends on which encoding method is used. The decoding latency depends very much on the hardware decoder support available in the graphics card. Decoding in hardware is usually faster than in software. Generally, H.264 is more time-consuming to decode than MJPEG. For decoding in H.264, the latency also depends on the profile you chose in the encoding phase. Base is the easiest to decode, while main and high will take longer. The H.264 data stream produced by Axis video products requires the decoder to buffer at least one frame.

4.3.4 Display device

The display device also affects latency, through the transfer time, the refresh rate, and the response time.

The transfer time is the time it takes for the decoded video data to be sent from the decoder through the cable (for example HDMI) to the monitor. The transfer time depends on the speed of the cable and the resolution of the monitor. For an FHD monitor connected with a standard HDMI cable, this adds about 10 ms latency.

The display device's refresh frequency also affects latency. For computer monitors, the refresh rate is around 17-20 ms, whereas special gaming monitors have a refresh rate of 4-5 ms.

The response time is the time it takes for the pixels in the monitor to change value. This depends on how large the change is, but for larger changes it can add 5-20 ms latency.

5 Reducing latency

Designing a system to meet low-latency goals will require other tradeoffs. You need to decide what the acceptable latency is and find a balance between video quality and cost of the surveillance system. This chapter provides a few simple recommendations regarding the camera side, the network, and the client side, to reduce the end-to-end latency.

5.1 Reducing latency in the camera

- **Resolution.** Choose a lower resolution if possible. Higher resolution implies more data to be encoded, which could lead to higher latency.
- **Enhancements.** Reduce image enhancements, such as rotating, de-interlacing, and scaling. The use of image enhancements can add latency.
- **Low latency mode.** You can optimize the image processing time of your live stream by turning on low latency mode in the plain config settings. The latency in your live stream is reduced to a minimum, but the image quality is lower than usual.
- **Privacy masking.** Consider not using privacy masking, because it adds latency.
- **Encoding.** Make sure that the encoder provides the level of control over latency that your system requires. There needs to be a balance between the amount of data and the capacity of network infrastructure. If the video is sent through a network with limited bandwidth, choose H.264 or H.265 as the encoding method. This will lead to lower bitrate due to harder compression. Choose baseline profile if the network can manage the bitrate, because baseline will be easier to encode and decode.
- **Storage profile in Zipstream.** Consider not using the storage profile, because it adds latency.
- **Number of streams.** Limit the number of streams from the camera with different settings. Each unique combination of settings, such as resolution, frame rate, and compression, requires its own individual encoding process, which adds load to the processor and cause delay.
- **Bitrate.** Try to use a lower bitrate. To reduce the latency in the camera we need to reduce the amount of data being generated. The bitrate is affected by many factors, such as light conditions, scene type, as well as compression level, resolution, frame rate, and more.
- **Frame rate.** Use as high frame rate as possible. As frames are encoded and decoded one frame at a time, the buffers will delay at least one frame. With higher frame rates, the delays caused in buffers will be reduced. For a stream with 30 fps, each frame takes 1/30 of a second to capture. We can then expect a latency of up to 33 ms in buffers. For 25 fps, the delay will be up to 40 ms.
- **Capture mode.** Consider using a capture mode with as low resolution and as high frame rate as possible. The low resolution means fewer pixels to process and the high frame rate means reduced buffer delay.
- **Audio.** Consider not using audio. Audio that must be synced with the video requires a larger playback buffer, which leads to higher latency.

5.2 Reducing latency in the network

Many of the recommendations concerning the camera side are aimed at limiting the total data volume being sent though the network. In most cases, a limited network is the largest contributor to the end-to-end latency. If the network has a high capacity then many of the above recommendations are not needed. Make sure that your network has a good quality of service, and that all the hops within the network are configured to suit your video demand. Make sure that your bitrate over the network is guaranteed to be able to deliver the data output from the camera.

5.3 Reducing latency on the client side

Improvements on the client side have a big impact on the end-to-end latency, and there is usually a lot you can do.

Processor and graphics card. The CPU plays a central role in the client side latency. Use a good processor with enough capacity to process the video stream and handle other requests simultaneously. Use a good graphics card that is updated with the latest software and support for decoding.

Viewer/VMS. Make sure that your viewer does not have an unnecessarily long play-out buffer, but be aware of the tradeoff with video jitter.

Display. Use a display with as fast refresh rate as possible. For a pleasant live view (not affecting the latency, though), adjust the screen frequency to a multiple of the camera's capture frame rate. An example would be 60 Hz for 30 fps mode or 50 Hz for 25 fps mode.

T10180461/EN/M1.11/2406

About Axis Communications

Axis enables a smarter and safer world by creating solutions for improving security and business performance. As a network technology company and industry leader, Axis offers solutions in video surveillance, access control, intercom, and audio systems. They are enhanced by intelligent analytics applications and supported by high-quality training.

Axis has around 4,000 dedicated employees in over 50 countries and collaborates with technology and system integration partners worldwide to deliver customer solutions. Axis was founded in 1984, and the headquarters are in Lund, Sweden

AXIS® COMMUNICATIONS