Video Streaming Protocols

Digital Video Encoding

The Helios HD/SD cameras support four encoding modes providing up to two independent streams of digitally encoded video as listed below:

- H.264 Only
- MJPEG Only
- H.264 Stream 1 and MJPEG Stream 2
- H.264 Stream 1 and H.264 Stream 2

(Not all of these modes are available on all of the Helios cameras. The HD camera video source produces progressive scan video and supports all four encoding modes. The SD camera produces interlaced video and only supports the last two dual streaming modes)

Video Transmission

TCP vs. UDP – At the Transport layer of the IP network stack, UDP (User Datagram Protocol) is the preferred method for the delivery of live video streams. UDP offers reduced latency over the reliability that TCP (Transmission Control Protocol) provides. It is a faster protocol than TCP and where time-sensitive applications are involved (i.e. live video or VoIP), it is better to live with a video glitch caused by a dropped packet than to wait for the retransmission which TCP guarantees (which in not very practical where live video is concerned). However, as we will discuss later, TCP is definitely more firewall friendly as some networks will block UDP video.

UDP Video

Multicast Video – One-to-many communication over an IP network/UDP video only

Ideally, in a network where there is more than one client/viewer that wants to see a live video stream, you should use multicast video. Multicast video always uses UDP at the Transport layer. With encoded multicast video, the camera transmits only one instance of the video onto the network and the clients in turn grab a copy of the video from the network. Each client must first connect to the source camera to grab an SDP (Session Description Protocol) file which gives them the information needed to find the video on the network and begin decoding and rendering. With multicast video, CohuHD cameras listen for both RTSP (Real Time Streaming Protocol) and HTTP (Hypertext Transfer Protocol) as a means to make the initial connection to the SDP file.

Example - *rtsp://<ipadd>/stream1multicast or* *http://<ipaddr>/webshared/Stream1.sdp*

Advantages:

- Less network bandwidth utilized than unicast video when multiple clients are involved
- Fewer camera resources are used, which allows virtually an unlimited number of client viewing stations or hardware decoders

Disadvantage:

- Many networks don't support multicast– WARNING: if all of the network hardware that the video transverses does not support multicast/IGMP snooping, you can bring the network to its knees. (i.e. unmanaged switches without IGMP snooping will flood every port with multicast video)

Note: CohuHD Helios cameras in multicast mode listen for and support unicast connections (which we will outline later) with the exception of the following mode:

MPEG2 Transport Stream (M2TS) - CohuHD Helios cameras support another method of multicast streaming and this M2TS mode works for both H.264 and MJPEG encoded video. These UDP multicast packets include M2TS headers making the stream compatible with many broadcast decoders. You can use VLC media player and connect with the following syntax:

udp://@<multicast-destination-ipaddr>: <multicast-destination-port>

Because the M2TS packets include the information necessary for decoding the stream, no connection to the camera (for the SDP file) is needed. You can pull the video packets directly off the network.

Two notes on the M2TS stream:

1) The Helios Viewer and IE (CohuHD ActiveX control) do not support this streaming mode
2) Once enabled all other modes of streaming are disabled. You will need to disable the M2TS stream to re-enable the RTSP media server on the cameras

Unicast Video – One-to-one communication/RTP over UDP

This is the simplest way to transmit video. Each client connects to the camera via RTSP to get the SDP file and after the UDP ports are negotiated, video is streamed to the client. The video is transmitted using RTP (Real-time Transport Protocol) which includes timestamps and end-of-frame marker bits to synchronize the video stream playback. This is very similar to the multicast situation except that with every client, the camera starts a separate video stream. Compared with UDP multicast, this not only increases the network bandwidth used but also requires more load on the camera for each successive client.

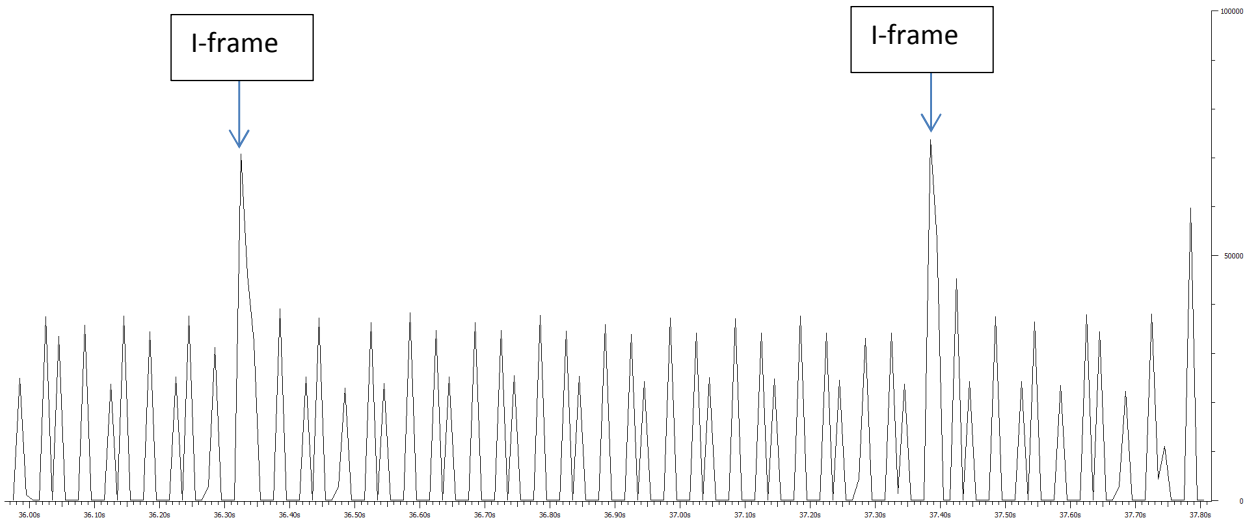
Connection URL example – *rtsp://<ipaddr>/stream1*

Video Streaming Protocols

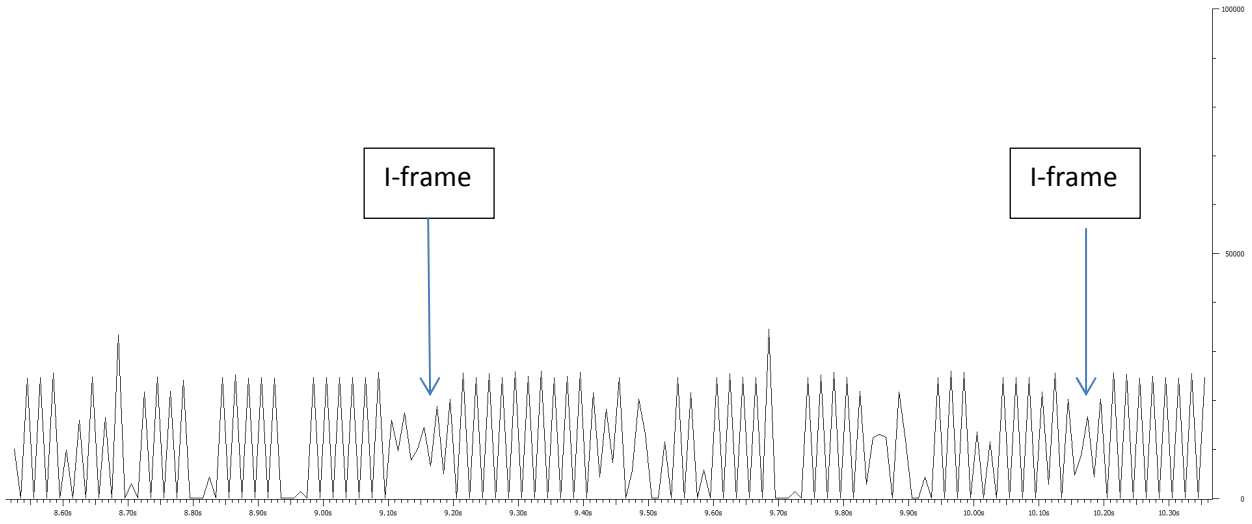
Constrained Mode – UDP video for cellular/wireless networks

With H.264 (not MJPEG) UDP video, CohuHD supports a constrained mode of operation. In testing and actual deployment, we have found this can greatly improve the reliability of a video stream over cellular and wireless/radio networks that are not only more prone to network interruptions but may have smaller buffers for handling UDP video packets. The large H.264 I-frame(s) that are sent by the camera can overrun what can be reliably handled by the cellular/wireless devices and UDP packets can be frequently dropped causing video anomalies and in the worst cases, no video at all.

With the constrained mode enabled, the camera momentarily buffers the transmission of the large I-frame and then sends the data out in a more evenly distributed fashion. The tradeoff is a small amount of increased latency due to the buffering on the camera side. Below is the graph of our normal UDP video stream.



You can see the peaks that show the I-frame transmission. Here is a graph of the video packets with the constrained mode enabled:



As you can see, the video transmission is distributed without the large burst of data that normally occurs with these H.264 I-frames.

TCP Video

As mentioned earlier, UDP is usually the preferred method of video transport. However, some networks will block UDP video. In particular, firewalls and NATs may prevent UDP video from getting to the clients. In those cases, we offer a couple of firewall-friendly TCP modes of video.

RTSP Interleaved – RTP/RTSP over TCP

This starts in the same manner as UDP video with a client sending an RTSP request to the camera over the well-known RTSP port TCP 554. Then, instead of the client and camera negotiating the UDP ports that will be used for the video, the client requests the video to be interleaved over the existing RTSP socket that is already established. This should get thru most networks that may block UDP video.

Connection URL example – *rtsp://<ipaddr>/stream1*

Video Streaming Protocols

As you can see, this is the same syntax as in the RTP/UDP unicast video example. The client in the RTSP SETUP command, as part of the initial connection process, makes the request for interleaved video.

HTTP Tunneling – RTP/RTSP/HTTP over TCP

In some cases, even the well-known RTSP port -TCP 554 can be blocked. Firewalls and proxies can restrict access to only allow traffic on TCP port 80 - the HTTP port used for Internet/web access. With HTTP tunneling, the RTSP Interleaved connection can be mapped for use over port 80. Using standard HTTP GET and POST commands, video is transmitted over the same HTTP port that a client would normally use when connecting to an Internet web page.

Connection URL example – *rtsp:///<ipaddr>/stream/http/stream1*

About CohuHD Costar™

CohuHD Costar™ is a leading manufacturer of high-definition video surveillance camera systems designed for the performance requirements associated with critical infrastructure applications and is now part of Costar Technologies, Inc.

CohuHD Costar solutions integrate the latest high-definition video imaging and compression technologies into our ruggedized camera products. CohuHD Costar is a high-value, solutions provider for monitoring in the most critical, sensitive environments. We focus on providing the most reliable, rugged, outdoor PTZ cameras in the market.

For more than 50 years, we have manufactured in the U.S. the most reliable, rugged video cameras available.