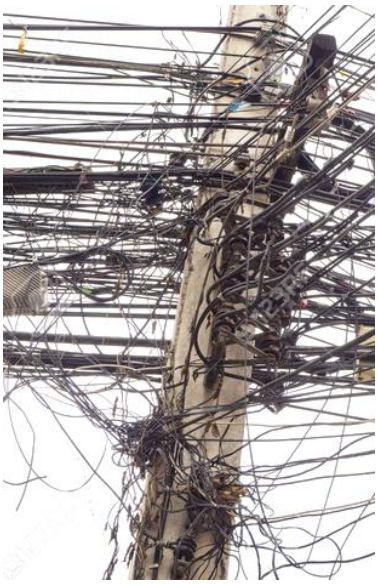
Micro Local Cameras – Best Practices

By Dan Desjardins – Director/Owner Videstra LLC

With an investment in one or more tower and rooftop cameras (we call them Micro-Local cameras there are going to be some high expectations from viewers as well as station management. Adding reliability and security to these cameras doesn't have to be complicated. There are a few things you can do to make sure your teams get everything they expect from your investment by following a few suggestions we will present in a series of articles called Best Practices. Today's White Paper:

Best Practices - Diagnosing Remote Issues



The Internet…

The Internet is a magical realm. The services and information available are mind numbing – but what is more amazing is the technology. The volume of technology whizzing and spinning between you and anything you access resembles a living road map of New York City with complex intersections, bridges, gates and … numerous dead-end streets. Then there's the occasional multi-car pileup… Despite the complexity it works well – and if most people pay attention to the rules it will continue to do so. Unfortunately, there are challenges that can make what should be, or could be a simple solution turn complex and ugly. This is when having some basic troubleshooting skills can be handy – if not essential!

Cameras – The Protocols

When you put a Micro Local camera on a tower or building and connect it to the Internet you are creating what is called an endpoint. This endpoint will employ the Internet protocol called TCP/IP so you can connect to it from your facility, configure the camera and receive live video and images. Layered on top of TCP/IP are a couple of other protocols specifically designed for camera control and video. Camera control is all done via a familiar protocol called HTTP and video is returned from the camera via RTSP (Real Time Streaming Protocol). Camera control over HTTP always uses TCP/IP and this is the most common protocol on the Internet. Camera *video* can use RTSP over TCP/IP or via UDP. TCP/IP is a much more reliable protocol, but it includes elements that require additional bandwidth that makes sure every packet sent from the camera to the receiver (decoder) gets there. UDP on the other hand, is a

lighter protocol requiring less bandwidth, however video sent via UDP is done so without regard for successful reception. UDP is often described as a *fire-and-forget* protocol. Video frames sent via UDP can get lost along the way as there is no mechanism in UDP to guarantee reception. You might think this to be a problem, but under normal circumstance it really isn't. While losing a video frame can (and does) result in stuttering/freezing video – the mechanism used by TCP/IP to make sure all packets are delivered is a slow and redundant process that allows the camera to resend packets of video the receiver hasn't acknowledge in time. With most data on the Internet this is just fine. But with video the frames encoded in a lost pack will often no longer be needed after a lengthy back-and-forth between the camera and decoder. Simply put, the video frame(s) lost may be older than the frames currently playing on the decoder so the decoder will throw them away. That old saying, "*better late than never*" does not apply to frames of video.

Video is a highly *temporal* data stream. Unlike documents and still images, all video frames must arrive on time to be played in their original order and frames that arrive too late, due to a poor connection and protocol negotiations are of no value and get thrown away anyway. With UDP, lost frames are simply lost and never re-requested because the assumption is that they would arrive after the party and the bar is closed…

But – with all that said TCP/IP does offer some additional surety to an RTSP stream that can make your connection to a camera more reliable in the long run. Large buffers that delay the playout on the decoder will usually allow sufficient time for re-transmission of missing frames due to lost packets. Of course, large buffers mean the video may be delayed by many seconds – and in most cases this can be acceptable.

Videstra allows connections via UDP or TCP to most cameras. Below is a chart that outlines the advantages and disadvantages.

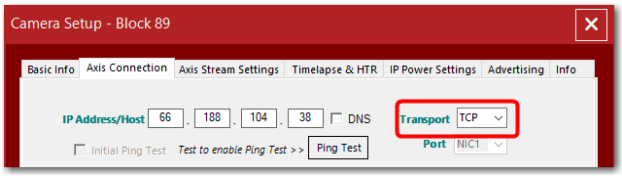
UDP

- Requires extreme network reliability
- Offers lower latency
- Uses less bandwidth

TCP

- Offers more reliability
- Uses/requires more bandwidth
- Has greater latency (longer video delay)

While low latency and less bandwidth may seem like great advantages, the requirement for an extremely reliable network often negates the use of UDP. For this reason, Videstra recommends using TCP for all your camera connections. Within the camera panel settings in Videstra you may select UDP or TCP.



Setting Transport Protocol to TCP

Lost Packets

A packet is how data is transmitted from one device to another on a network. For this paper these packets will contain full, or partial frames of video produced by your camera(s).

Between you and your cameras there can be *dozens* of hardware switches and multiple public and private networks. It will depend upon your ISP, but it is not unusual to have 30 or more "hops." Each hop can be considered a potential failure point. Each hop is a router or switch being managed by people that include your ISP, public/private agencies, and your own IT department.

The number of acceptable lost packets along this circuitous pathway is zero. That may sound unrealistic (and it is) – but for the sake of high-quality video (that is the goal) than you cannot afford to lose *any* TCP/IP or UDP packets. Remember TCP/IP is *supposed* to guarantee all packets get to their destination. Unfortunately, we live in the real world and somewhere along a 39-hop path packets will sometimes encounter overwhelmed, underperforming, or defective hardware that cannot keep up. Packets will get lost from time to time as swamped equipment takes too long to respond. Each packet in a stream is marked with a *time to live* value – and once expired any router along the way is obligated to discard it. Any lost packet may contain an entire, or portion of a video frame. The result is now stuttering or freezing video.

Losing video frames

Video frames from Micro Local cameras are in a format called h.264 (AVC) or h.265 (HEVC). This is a *temporally* compressed format. Not all frames contain the same amount of information. There are **I frames, B frames and P frames**. I frames are the largest and contain an entire picture. B and P frames are the smallest and only contain *differences* since the last I frame. You may have heard the term GOP (Group of Picture) when referring to AVC or HEVC. Simply stated, the GOP is a number and it is the number of frames between I frames in an AVC or HEVC video stream. A typical GOP can range from 15 to 60 frames (sometimes much more). Videstra recommends a GOP size of 32 frames. It's important to recognize that the *all-important* frame in a video stream is indeed the **I frame**. It is the foundational frame in any stream and losing a I frame will *always result in poor video* that is seen as freezing, smearing, or large portions of the frame showing as all gray/green/black. How poor video is shown is a matter of how the video decoder handles errors. With Videstra *any lost frames* result in freezing of the last full frame received or properly assembled from I, B and P frames. While not ideal, we feel a frozen whole frame is superior to smearing or obviously broken video.

With Videstra *any lost frames* result in freezing of the last full frame received

Here's the rub. If you are losing packets that contain I frames, the decoder will freeze *until is can receive another intact I frame.* If your GOP is 32 frames than you will see the video freeze for at least 1 full second (30 frame video rate). In practice, it may take more than one I frame to let the decoder recover because the errors causing the initial packet loss will likely persist long enough to eliminate more than one I frame. Now you know.

What to do When Losing Packets/Video Frames

If you are seeing freezing/stuttering video here are the most common causes

- Insufficient Upload Bandwidth at the camera endpoint
- Packet loss along the path from the endpoint to the decoder
- A defective Camera
- A defective decoder

Experiencing packet loss can be a frustrating experience – but having some strategies to diagnose the issue will go a long way to fixing it. American inventor and engineer Charles Kettering famously said: "A problem well defined is a problem half solved."

> *A problem well defined is a problem half solved*
>
> -Charles Kettering

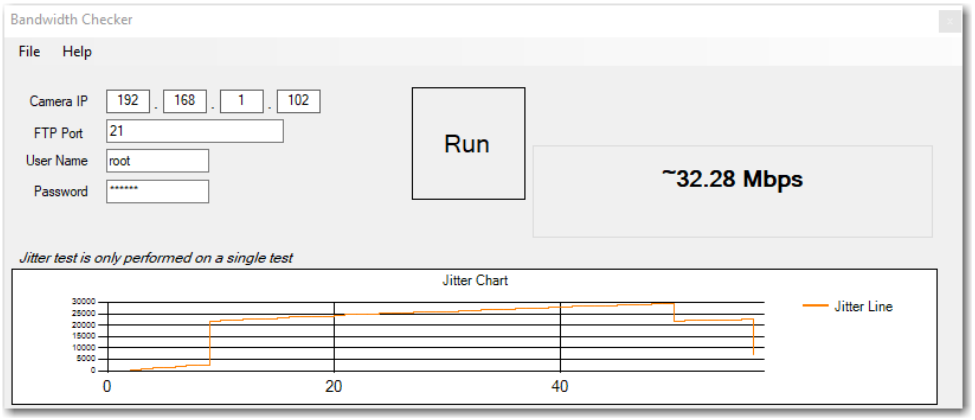
In our bullet point list, we indicate that one cause can be either a defective camera or decoder. We will not address these issues in this paper. We will, instead, focus on making sure you have sufficient upload bandwidth or are not suffering from significant packet loss.

Diagnosing Insufficient Upload Bandwidth

If you read our first white paper Remote Internet Service Checklist then you learned that your endpoint *upload* bandwidth is a critical factor. Since the camera is sending video **to** the internet it is the available upload bandwidth with which you must be concerned. If you are suspicious that your ISP has not provided you with sufficient upload bandwidth then you must run some tests to determine what you have. Normally this must be done on-site. ROAD TRIP! Take a laptop, go on site, connect to the Internet and then go to https://www.speedtest.net.

But that is a real pain. You can do a *reverse* bandwidth check if you an Axis™ camera and have opened port 21 for FTP to the camera.

Videstra has a small software tool called the Reverse Bandwidth Tester which can be used to connect to an Axis camera from your facility and measure the Upload Bandwidth available to the camera. You can leave this tool running to get an upload bandwidth profile over time. Do not run the tool while using the camera though as it is likely to steal most of the available bandwidth in bursts.



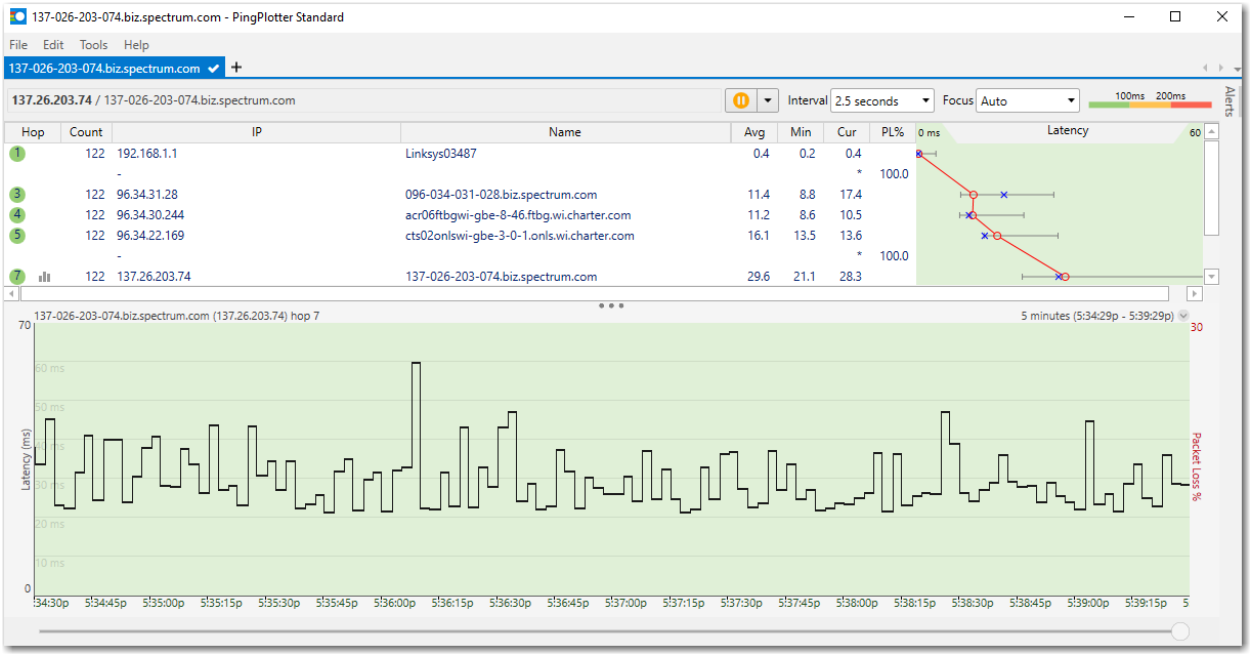
A few rules about the reverse bandwidth tester – it is basic and only works with Axis cameras. You must have the root password for the camera as it uses an FTP connection. On Axis cameras FTP is only available to the root user.

If you do not have an Axis camera, or do not have the password for the "root" user, or do not have access to FTP at all – then this tool will not work. You will need to do an on-site visit to determine upload bandwidth.

Examining the Trace Route

Before you make an on-site visit it will be worth running some trace route tests to see if packets are getting lost. For this we highly recommend a tool called **PingPlotter**. This is a useful tool that comes in

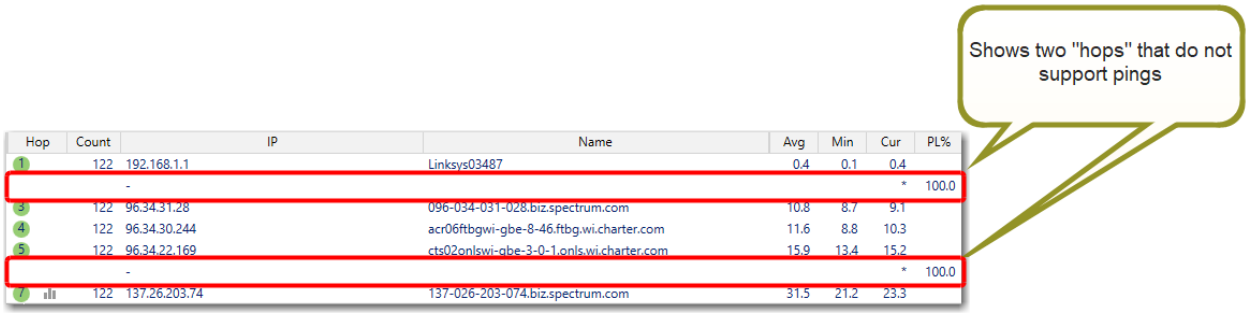
free or paid versions. The free version will only show you 10 minutes worth of ICMP Pings to the endpoint, while the paid version can show you more than a day.



PingPlotter 5 – Standard

PingPlotter is a graphical tool that does a very similar job as the command line tool Tracert. Unlike Tracert it is faster and plots the result in a way that is informative and visual.

When you set up your remote camera endpoint we strongly recommend you turn on ICMP support and allow pinging. Like Tracert, PingPlotter relies on ICMP support and the ability and permission of each node along the route to respond to pings. Today many IT managers prefer to turn off ICMP Ping support and keep all endpoints hidden from view. If a few of the hops along the router do not respond to pings they will show up with asterisks.



In the example above two hidden nodes do not respond to pings. This is ok because nodes afterwards do indicate that the hidden nodes are passing the pings just fine.

If the final node did not respond to pings then this tool becomes much less useful since we can no longer visualize the full route to the endpoint. Therefore, we recommend leaving full ping support on at each camera endpoint.

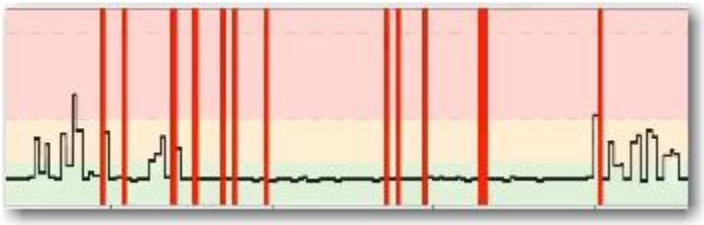
we recommend leaving full ping support on at each camera endpoint

Problem Indicators in PingPlotter

There are several things you may find when running PingPlotter on an endpoint with poor camera performance:

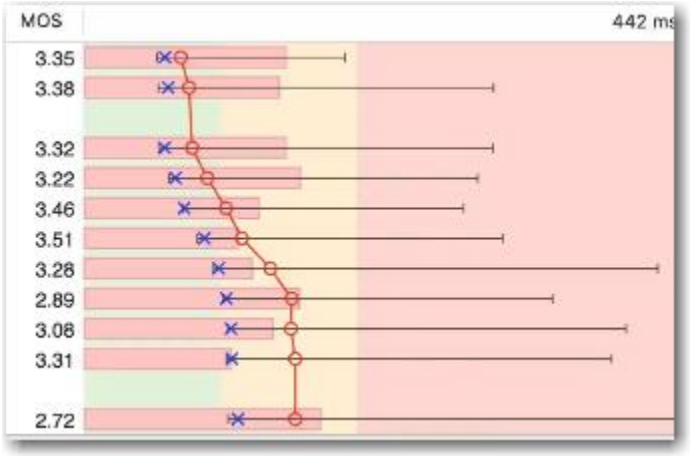
- Dropped packets
- Extremely delayed packets
- Inconsistently delayed packets
- Flaky nodes on the tracert path

Dropped packets become obvious in PingPlotter.



Long red lines indicate one or more packets were lost. The fatter the red line, the more packets lost. This is bad. As stated earlier, lost packets mean video from a camera will freeze for one or more seconds – often longer. There may be a clear indication in the list of hops where the packets are getting lost – sometimes it is not obvious. Either way – you will need to work with your ISP to clear this issue up. Having commercial service from an ISP is helpful in this case. Consumer level support often has neither the personnel nor expertise to address this.

Delayed packets are packets that took an inordinately long time to return.



PingPlotter shows three levels of packet delay: current, avg and max. if there are a significant number of packets that take longer than 150 ms to return from one or more nodes, this is an indicator that the node is overwhelmed. Consistent long return times of 60 to 100 ms is not a major concern. Inconsistent return times that range from say 30 ms to 250 ms is a concern as this means the buffers on the decoder will likely become depleted. A *depleted buffer* forces the video output to freeze while it "catches up." If you observe a significant number of inconsistent delays you will need to report that node to your ISP if this is a persistent problem.

Flaky nodes are those that *sometimes* indicate they are not responding to pings. If you observe PingPlotter over time and see that a specific node sometimes responds, and sometimes does not – this

node is most likely overwhelmed with traffic or has defective hardware. This must be reported to your ISP.

A Few Common Solutions

Before you contact your ISP it is wise to clean house first. Make sure your camera and any networking equipment at your endpoint demarc are in proper working condition. Upgrade firmware.

Here are some common issues you should look to address at your endpoint:

- Power Supplies (including POE Injectors)
- EMI/RFI Noise

One common problem is with power supplies. Wall Warts often age poorly, and it is wise to make sure these are not going bad, or injecting noise. They are cheap to replace. POE Injectors also go bad. I can tell you that on more than one occasion a problematic camera experiencing lost packets was entirely addressed by replacing the POE injector with a new one. Videstra recommends you keep at least one spare POE injector on-hand.

EMI/RFI can be addressed by placing a power conditioner at the demarc of the camera. Many cameras are on rooftops and often are near HVAC equipment and/or elevator motors. This equipment will throw spikes into the local power at what will appear as random times. As we explain in our Installation/Deployment White Paper – a relatively inexpensive power conditioner is a good investment in reliability.

Don't be Pedantic – But Insist on What You Are Paying For

Technically the number of acceptable lost TCP/IP packets is ZERO. That's probably not realistic though. An occasional lost packet is unavoidable. Several lost packets every 10 minutes is a problem that should be addressed. Work with your ISP to get these issues cleared up – having tools like PingPlotter and the results of upload bandwidth tests at-the-ready can convince your ISP that the issue is on their side. It is always wise to first make sure your endpoint is clean before contacting your ISP though.