

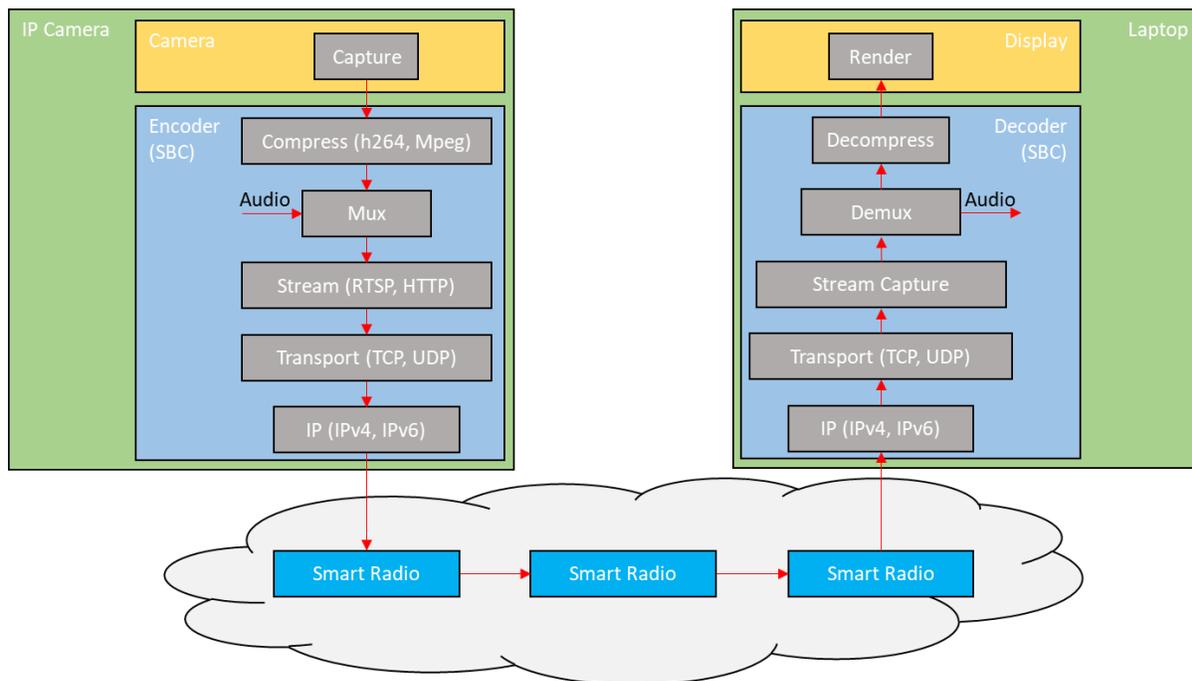
# Smart Radio Video Streaming Tutorial

## Introduction

This document is a basic tutorial on how to get started with video streaming using the Smart Radio. We will cover a few important steps. For this tutorial we consider a typical use case as an Unmanned Vehicle sending a video feed to a Control Station.

1. Video System Block Diagram
2. Hardware Setup
3. IP Camera Configuration
4. Smart Radio Configuration
5. Video Player Configuration
6. Results

## Video System Block Diagram



**Figure 1 – Video Data Flow**

The block diagram above illustrates the data flow path of a simple streaming application. The cloud of Smart Radios represents a mesh situation where the video feed may hop across a node before arriving at the destination Smart Radio. The light blue blocks represent a possible hardware setup where an SBC such as an Nvidia Jetson or Raspberry Pi might be used for network and video processing and the camera or display is separate. The same task could be performed using an IP camera on the transmit side, and a laptop on the

receive side. It is also possible to split the system in other ways with other dedicated hardware such as video encoder/decoders, video streamers, or cameras with built in compression. Using a system with built in encoding/decoding and streaming such as an IP camera offers simplicity at the expense of control. Writing a streaming application on a suitable SBC/SOM allows for control over the type of codec, the streaming protocol, and how they behave.

Below are some important parameters to consider.

## Camera

Some important points to consider are

- Sensor – CCD or CMOS sensor, the size of the sensor, the resolution, color or monochrome...
- Frame rate
- Shutter – global or rolling.
- Interface – USB, CSI, HDMI...
- Lens – focal length, aperture.
- Size and Weight

For UAV guidance for example, a global shutter may not be required, but a high frame rate and CSI interface might be important for low latency. As the requirements are unique to each application, we will not go into any detail.

## Video Compression

The first important point to take into consideration is type of video compression being used. Attempting to stream raw video data is not recommended as it will only be possible to stream at very low resolution.

It is possible to significantly reduce network overhead by choosing a codec which includes compression in both the image space, and time. H.264 or H.265 are the most common streaming codecs being used today. We suggest choosing hardware that supports the H.265 codec H.265 HEVC (High Efficiency Video Compression) can be used to deliver 1080 p30 video quality at only around 3.4 Mbps while H.264 codecs need around 4.5 Mbps bandwidth

The table below shows the bandwidth required for various video resolutions with H.264 and H.265 HEVC codecs. We recommend choosing hardware which supports H.265.

**Table 1: Network throughput required for different compression schemes**

Video Resolution @ 30 fps	H.264 codec – High Quality. Required Throughput (Mbps)	H.265 codec – High Quality. Required Throughput (Mbps)
SD (480p)	0.7	0.5
HD (720p)	2	1.5
Full HD (1080p)	4.5	3.4
Ultra 4K HD (2160p)	18	13

## Latency

Each stage of the system adds a certain amount of latency which affects the overall “glass-to-glass” latency. Latency can arise for various reason. At the receive side, it is common to buffer captured packets up to a certain number of frames in order to smooth out jitter in the lower layers.

The Smart Radio is optimized to minimize the impact of a weak wireless link, and in latency sensitive applications we recommend reducing any buffering on the receiver. If you are using an IP camera, choose a model which is optimized for low latency as some IP cameras include transmit buffering when they are trying to stream over a choppy link.

## Streaming Protocol

Various protocols have been designed specifically to cater to media streaming [1]. Tradeoffs exist between the protocols such as support for adaptive bitrates, ease of deployment, and end-to-end latency. RTP is a simple streaming protocol that works over UDP or TCP, and is extended using RTSP to include control over the video stream. This tutorial uses RTSP as it is a good choice for point-to-point links offering low latency and simplicity. RTSP is supported by common video applications and frameworks such as VLC, FFmpeg, and GStreamer. Further details on streaming protocols can be found here [2].

## Transport Protocol

There are only two transport protocols to choose from, UDP or TCP. The reliability of UDP packet transmission is the same as the underlying network. It cannot correct for corrupt or lost packets, but latency is minimal as the receiver is not required to acknowledge received packets. This makes UDP ideal when the end-to-end network latency is long (over the Internet for example) or when multicast is required.

TCP offers flow control and error correction through re-transmission. This results in a very robust link with control over network congestion. **Overall we recommend TCP for Smart Radio links and only suggest UDP if multicast is required, or if the video must be directly streamed across Internet.** Over a single-hop, the additional latency added by TCP over UDP is negligible, and for UDP, it may be necessary to increase the receive-side buffering to compensate for the poorer quality.

## Hardware Setup

For this guide, we are using an IP camera to stream video. Look out for our companion guide on getting started with video streaming on the Nvidia Jetson Nano.

The hardware setup is shown below. This is just a basic representation, and it is assumed that the user has followed the [Smart Radio Integration Guide](#) and good practices outlined there.

The IP camera's Ethernet port is connected to ETH0 of the remote Smart Radio. Likewise, the Laptop's Ethernet port is wired to ETH0 of the local Smart Radio. It is possible to buy a video encoder separately from the camera but in this document, we assume that the video encoder is integrated with the camera.

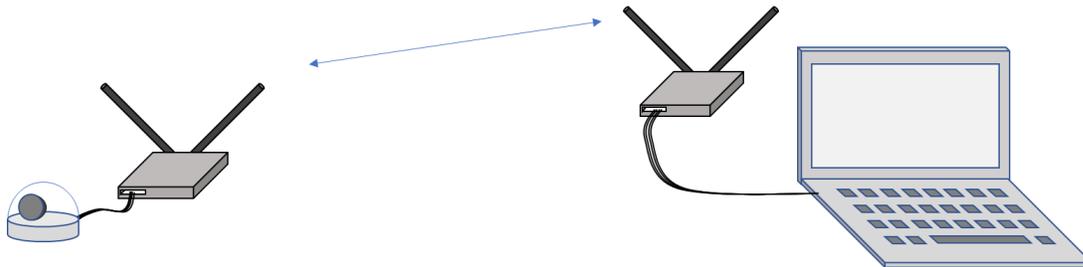


Figure 2 – Experimental Setup

## IP Camera Configuration

### Network Setup

Before configuring the video settings, make sure that the IP camera is on the same subnet as the receiving laptop. We recommend using the 10.223.0.0/16 subnet so that you can configure the Smart Radios through the same interface. Before trying anything at all with the Smart Radios, we recommend hooking up the laptop directly to the IP camera. From there you can test the glass-to-glass latency of the video feed without the Smart Radios in the chain. You can also optimize the video performance to your liking.

### Video Setup

For our tests, following settings were used.

- H.264 or H.265 Compression – As mentioned before, H.265 provides better performance and now easily available.
- Baseline Profile – Codecs provide different performance profiles, sometimes proprietary. It is possible to use them. Just make sure that the profile is compatible with the video player being used. For our camera, we used the baseline profile for H.264, but only one profile was possible for H.265.
- 8 Mbps Bitrate – In general, higher bitrates provide better video quality at the expense of higher network bandwidth. However, after about 3 Mbps bitrate, the gains are marginal.
- 1080p Resolution – Higher video resolutions are possible with larger network bandwidth. Use a bandwidth calculator [3] and make sure Smart Radio link throughput is able to accommodate it.
- 25 fps – Generally 25-30 frames per second is considered very good. Higher fps is possible and requires higher bandwidth.
- I-Frame Interval = 15. With 25 fps, this is a 0.6-second interval. So even in case an I-frame is lost in transmission, the video will recover quickly.

Considering the 8 Mbps constant video bitrate, ensure that the wireless link supports a throughput of more than 12 Mbps in order to handle the changing wireless environment. The Smart Radio supports wide bandwidth at long range and is an ideal choice for streaming video over long distances.

## Smart Radio Configuration

Below are the simple steps to setup the Smart Radio for video transport. If you are not sure how to make any of these configuration settings, then please consult the [Smart Radio Configuration Guide](#).

1. WDS AP/Client Mode. Use the Simple Config interface in the web GUI to set the remote Smart Radio (UAV side) as the AP, and the control station Smart Radio as the Client. In general if mesh is not required, it is better to use the WDS AP/Client network mode. If your application requires multiple video feeds to a central node, then you should choose the central node as the AP. An AP/Client network is a star configuration, and all traffic must pass through the AP.
2. Non WiFi-Compliant Mode. This step only applies to RM-2450 model to avoid the interference from other WiFi nodes in the area. You can switch to any channel bandwidth other than 20/40 MHz to avoid WiFi compliance. This mode tends to lead to a more stable latency.
3. In the Xtreme models, reduce the distance setting to the maximum required in your application. This setting is found in the advanced tab of the wireless configuration.
4. Keep the TX Power in Auto mode. This provides the best range.
5. Make sure that the Smart Radio link is able to support the required throughput for the desired video resolution. Iperf can be used to measure throughput. We suggest to keep an extra 50% headroom for the interference margin. Refer to Table 1 for throughput requirements.
6. In the differentiated services UI page, enable video optimization, and add a new rule to send packets originating from the IP address of the IP camera to the video queue (screenshot below). In our example, we are simply putting all traffic coming from the IP camera in the video queue. It is possible to filter traffic by the IP addresses, protocol, or port number.

Note that Doodle Labs Mesh Rider uses special radio parameters for this queue to optimize the video transmission over wireless medium in high interference areas.

## Differentiated Services

With [DiffServ](#) you can mark network traffic selected by addresses, ports or services to be subjected to QoS.

Describe your traffic using the Source/Destination Host, Port and Protocol. The Differentiated Service CodePoint (DSCP) field in the IP Header will then be rewritten according to your configuration. The DSCP field is also evaluated by the WiFi stack to prioritise different types of traffic and use the four WMM Queues of IEEE 802.11. Some DSCP fields are marked in the configuration:

- CS6 is assigned to Voice, Command & Control (the highest priority 3)
- CS5 is assigned to Video traffic (priority 2)
- BE is assigned to WLAN best effort traffic, which is the default (priority 1)
- CS1 is assigned to WLAN background traffic (the lowest priority 0)

Voice/Command & Control Queue can be optimized for Ultra Reliable Low Latency Communications (URLLC) operation.

### General Settings

- Enable Differentiated Services
- Optimize Command & Control and Voice for URLLC
- Optimize Video Streaming

### Classification Rules

Source host	Destination host	Protocol	Ports	DSCP	Comment	Sort
all	all	TCP	2000	Voice, Command & Control (CS6)	ser2net telnet	↑ ↓ Delete
all	all	TCP	3000	Voice, Command & Control (CS6)	ser2net raw	↑ ↓ Delete
10.223.0.60	all	all	all	Video (CS5)	IP Camera	↑ ↓ Delete

Figure 3 – Differentiated Services Setup

## Video Player Configuration

For this tutorial we used [GStreamer](#), which is a framework for creating multimedia streaming applications. It is possible to use the GStreamer API to create your own application, but in this case, we used the GStreamer command line tool for video playback. The command we used was

```
gst-launch-1.0 rtspsrc location=rtsp://<user>:<password>@<source IP address> ! application/x-rtp, payload=96 ! rtph264depay ! avdec_h264 ! videoconvert ! autovideosink sync=false
```

Note that this command uses TCP for the transport protocol. We recommend TCP for point to point links, or links which do not have very significant latency. If you prefer UDP, then change "location=rtsp:///" to

“location=rtsp://”. It is possible to use different video player. Just make sure that it is compatible with the video codec and associated profile settings being used.

## Results

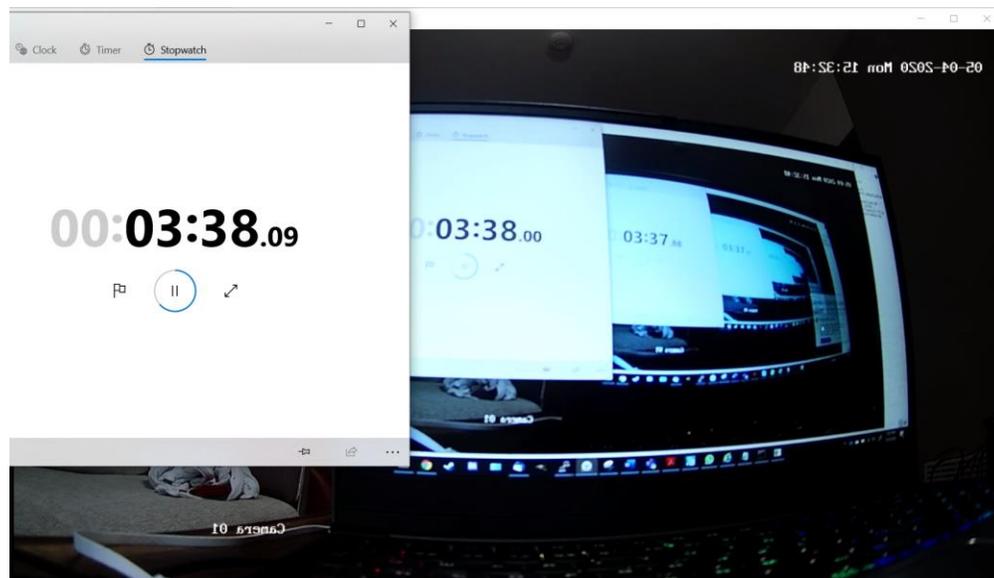
### Latency Measurements

We measured the latency for H.264, H.265, TCP and UDP over a single Ethernet cable, and over the Smart Radio wireless link. Latency was measured using a picture in picture setup. The webcam was pointed at the PC monitor (144 Hz) which had a stopwatch running, and the video stream was displayed on the same PC. As the measurement results are subject to various errors, it is important to take a large number of samples, and average.

The table below shows the results. We can see that the latency added by the Smart Radio network amounted to less than **around 10ms**. These results were taken over the course of a day, and as a result there was some variation in the results. They should be taken only as an indication of typical performance with the settings we chose, and not as a “datasheet” value.

**Table 2: Latency Results**

	Average Latency (ms)			
	H.264		H.265	
	UDP	TCP	UDP	TCP
Cable	103	106	118	111
Smart Radio Wireless	116	109	119	125



**Figure 4 – Screenshot of a Latency Measurement Data Point**

Doodle Labs experiments have concluded that these simple configuration steps provide high quality video transmission over Smart Radio wireless links.

## References

1. Streaming Protocols: Everything You Need to Know, August 26, 2019, Traci Ruether, <https://www.wowza.com/blog/streaming-protocols>
2. Cobalt Digital White Paper, “IP Video Transport Protocols – Knowing What to Use When – and Why”, [https://www.cobaltdigital.com/sites/default/files/IPTransprotocols\\_what-when.pdf](https://www.cobaltdigital.com/sites/default/files/IPTransprotocols_what-when.pdf), 2016
3. Network Bandwidth Calculator - <https://www.cctvcalculator.net/en/calculations/bandwidth-calculator/>