4K IP Delivery

Streaming and D-Streaming 4k

Scott Grizzle, M.A

12/9/2013

Recently we were asked to help someone create a proof of concept of streaming 4K content over high speed internet.

Not that long ago 1080 streaming was thought to be impossible and now 1080p streaming is almost mainstream. In a previous life I was the first to be able to demonstrate 1080p streaming at NAB 2007. Now with the upcoming advancements in Digital Cinema, UHD and 4K, new challenges have arisen. 4K isn't created by just multiplying a 1080p 30 x 4. It can offer 1080p 60fps via 3G x4 which is 8 times as much information as the normal 1080p.

With the generous help of JVC we were able to test 4K using the limited release GY-HMQ30 camera. This is a still under market testing but at this time it has not been released in the USA. This camera has an 8 megapixel 1.25 sensor that outputs 4K at 60fps and 24fps. This camera also offers 4 HDMI's out and a quad split on the output. So the 4k Stream can be split into 1080p quadrants and cam recorded onto 4 SD cards via the 4 SD slots in the camera body.

To demonstrate 4K encoding we decided to use the JVC GY-HMQ30 along with our HCoder Quad™ (with 4 channel H.264 encoder with mux), and a D-Streamer Magnus IP™ / Dig Quad. As a first step we elected to test the decoding output on a 4 HD input quad LCD monitor. A 4K monitor or Display plus a media processor or blender like the Sony MPE 200, AnalogWay Ascender, or Vista Systems Spyder would have presented a non segmented picture but none were available at the time of testing.

Our first goal was to encode the 4 HD streams converting the HMQ-30 output 4K via the 4 individual HDMI connectors each carrying a 1080p payload.

We used our HCoder Quad h.264 Encoder with a built in multiplexer which is crucial since when implemented in the real world the 4 sets of packets would travel together.

We used our D-Streamer Magnus Quad as the decoder since it can ingest the arriving multiplexed stream and send to 4 different HDMI or HD-SDI outputs after decoding the 4 streams.

Below are pictures of setup:

We connected the Quad Encoder to the camera and looked out the output via our 4 output decoder feeding the Quad 1080p monitor.

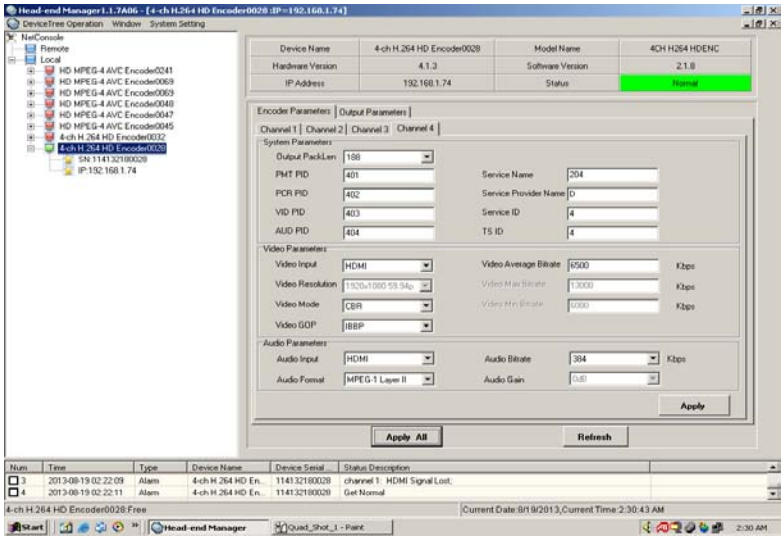
We took the 4 HDMI 1080p60 (1 per quadrant) from the JVC camera and fed the Quad Encoders 4 inputs. The Quad encoded each input/ quadrant to 6.5 mbps H.264 UDP. The 4 6.5 mbps streams were then multiplexed (muxed) into a single 30 mbps (null packets included to help with delivery).

Out of the Quad Encoder the muxed IP (quad 1080 videos) went to the decoder Quad Via IP and Internet. The decoder Quad demultiplexed the streams into individual HD-SDI channels that were all in sync. These 4 Channels were displayed on a quad 1080p display. So each quadrant was labeled. We wanted to see any movement from quadrant to quadrant.

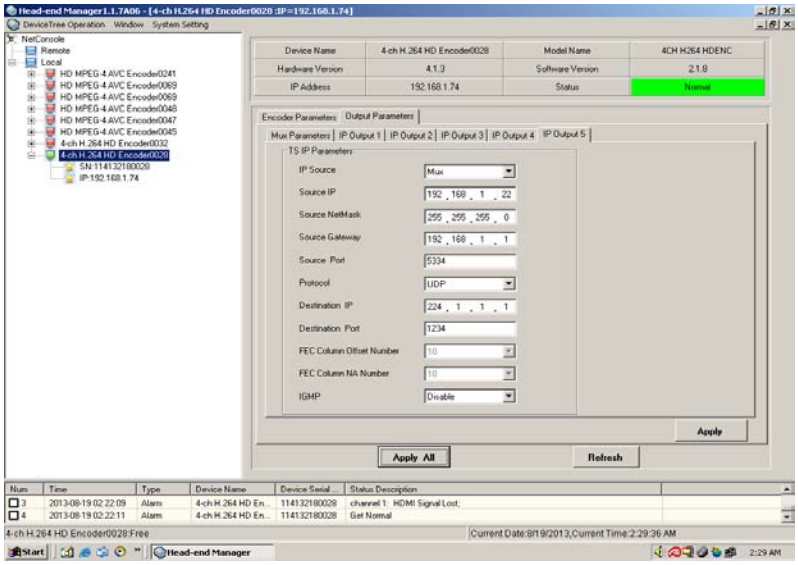
We were able to use the camera in live mode and see a panoramic view. But the test was most noticeable from content that was shot earlier in the week from Huntington state beach. We were able to see the boats going from quadrant to quadrant and then finally a plane flew in the shot. The plane was carrying a banner. So the plane flew in the bottom of quadrant 4 then flew up through quadrant 2 and out quadrant 1. The focus was being changed from good to bad to show that the content was not canned or staged.

The feeds coming from the camera through the Quad Encoder to the decoder Magnus the test was on.

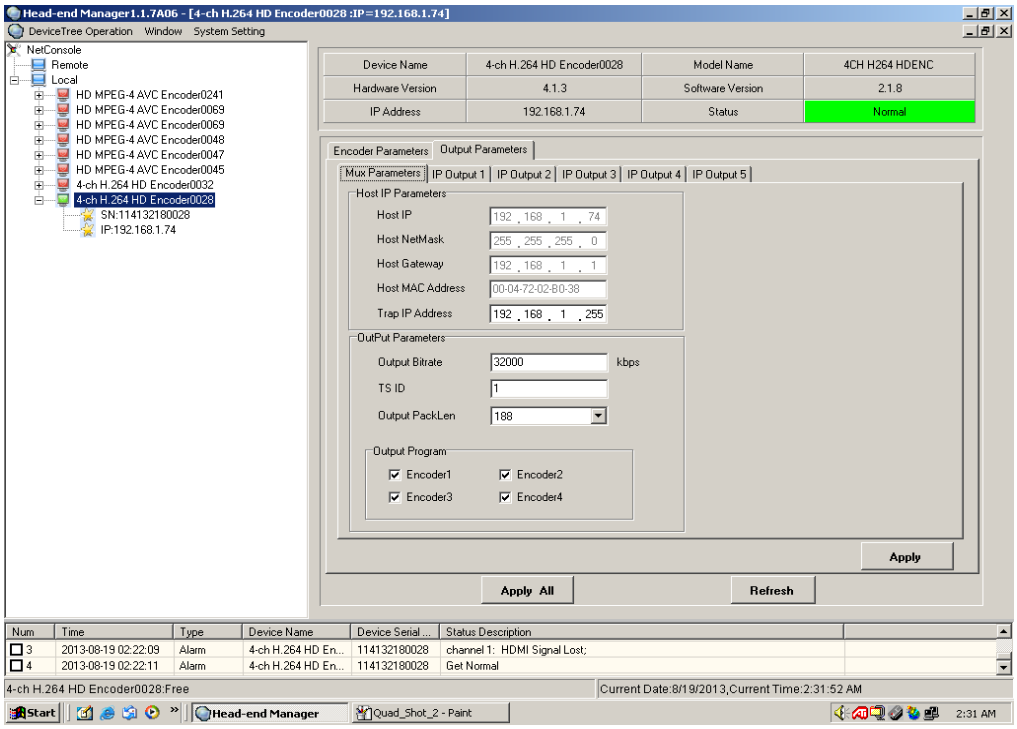
The first step in the encoding process was setting up the HCoder Quad's 4 Simultaneous HDMI inputs.



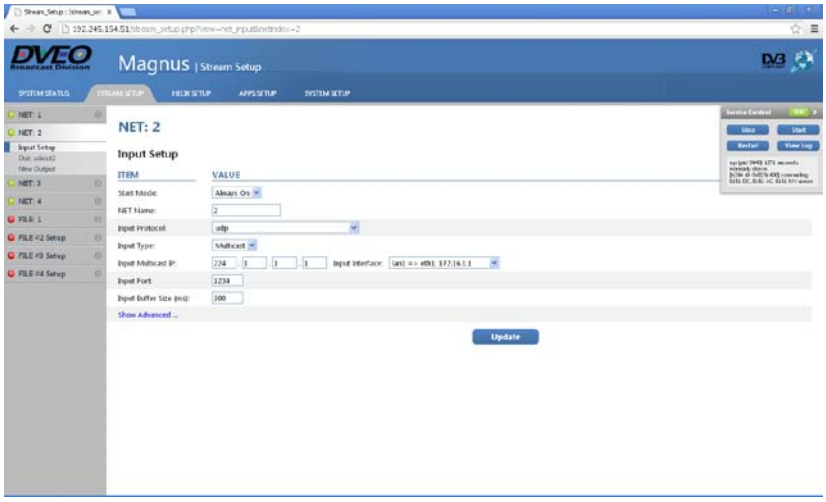
Each HDMI (quadrant) input has its own solo output.



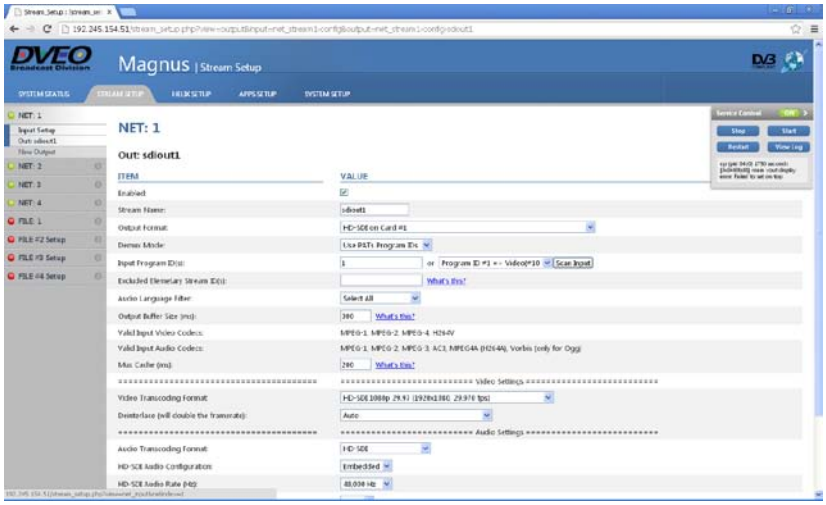
The 4 solo outputs muxed into a single synced multiprogram stream.

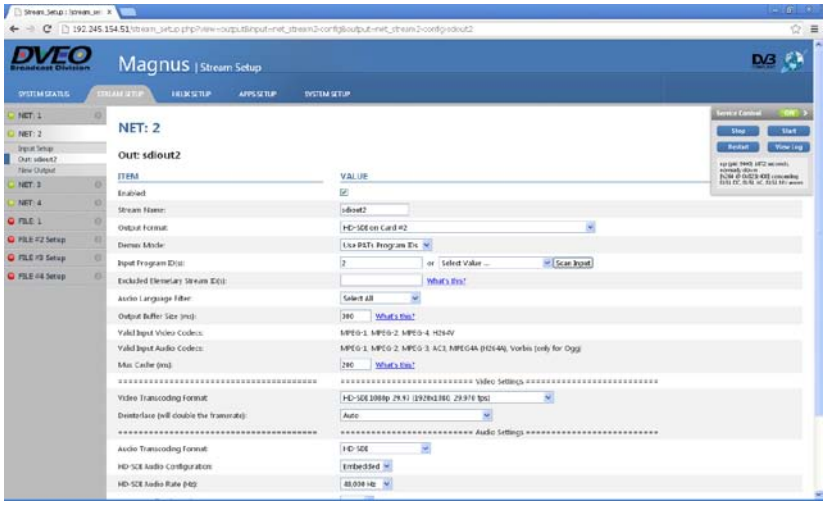


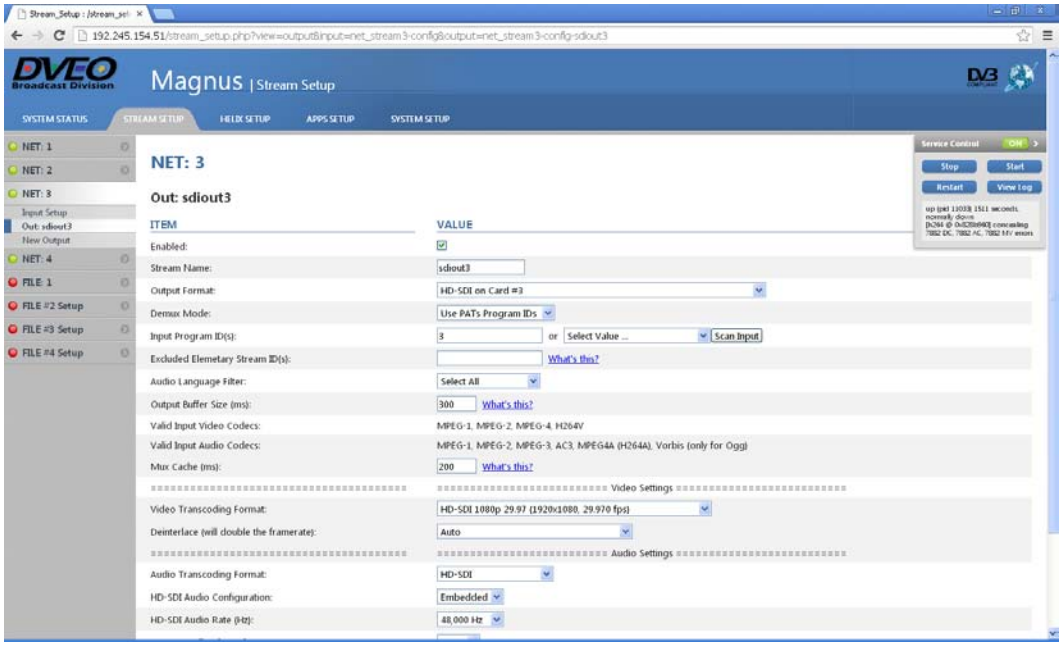
After the Quad Encoder is done encoding the 4 1080p muxed stream, the muxed stream is sent to the quad channel decoder. Now the reverse of the of the process happens. The muxed (4 1080p's) video is demultiplexed (demux). After the demux it's set into separate outputs so each quadrant can be sent to its own HD-SDI output.

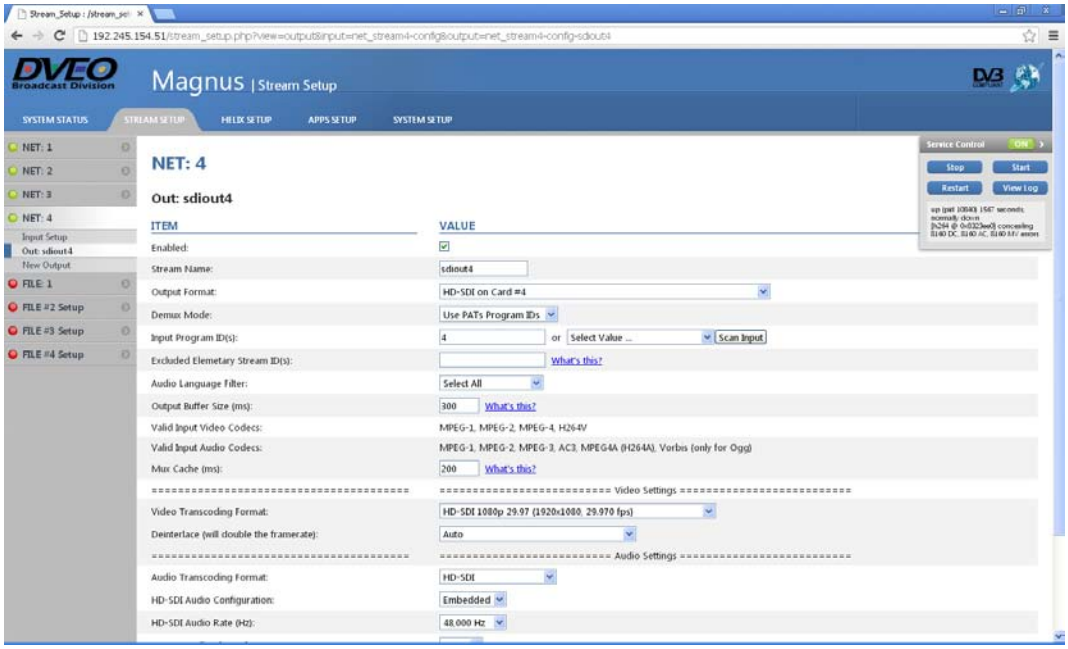


After the input of the D-Streamer Magnus is set, each of the corresponding outputs is set via the GUI.






Stream_Setup : /stream_set
192.245.154.51/stream_setup.php?view=output&input=net_stream3-config&output=net_stream3-config-sdiout3
DVEO Broadcast Division
Magnus | Stream Setup
SYSTEM STATUS
STREAM SETUP
HELIX SETUP
APPS SETUP
SYSTEM SETUP
NET: 1
NET: 2
NET: 3
Input Setup
Out: sdiout3
New Output
NET: 4
FILE: 1
FILE #2 Setup
FILE #3 Setup
FILE #4 Setup
NET: 3
Out: sdiout3
Service Control
ON
Stop
Start
Restart
View Log
ITEM
VALUE
Enabled:
Stream Name:
sdiout3
Output Format:
HD-SDI on Card #3
Demux Mode:
Use PATs Program IDs
Input Program ID(s):
8
or
Select Value ...
Scan Input
Excluded Elemetary Stream ID(s):
What's this?
Audio Language Filter:
Select All
Output Buffer Size (ms):
300
What's this?
Valid Input Video Codecs:
MPEG-1, MPEG-2, MPEG-4, H264V
Valid Input Audio Codecs:
MPEG-1, MPEG-2, MPEG-3, AC3, MPEG4A (H264A), Vorbis (only for Ogg)
Mux Cache (ms):
200
What's this?
Video Settings
Video Transcoding Format:
HD-SDI 1080p 29.97 (1920x1080, 29.970 fps)
Deinterlace (will double the framerate):
Auto
Audio Settings
Audio Transcoding Format:
HD-SDI
HD-SDI Audio Configuration:
Embedded
HD-SDI Audio Rate (Hz):
48,000 Hz


Stream_Setup : /stream_set
192.245.154.51/stream_setup.php?view=output&input=net_stream4-config&output=net_stream4-config-sdiout4
DVEO Broadcast Division
Magnus | Stream Setup
SYSTEM STATUS
STREAM SETUP
HELIX SETUP
APPS SETUP
SYSTEM SETUP
NET: 1
NET: 2
NET: 3
NET: 4
Input Setup
Out: sdiout4
New Output
FILE: 1
FILE #2 Setup
FILE #3 Setup
FILE #4 Setup
NET: 4
Out: sdiout4
Service Control
ON
Stop
Start
Restart
View Log
ITEM
VALUE
Enabled:
Stream Name:
sdiout4
Output Format:
HD-SDI on Card #4
Demux Mode:
Use PATs Program IDs
Input Program ID(s):
4
or
Select Value ...
Scan Input
Excluded Elemetary Stream ID(s):
What's this?
Audio Language Filter:
Select All
Output Buffer Size (ms):
300
What's this?
Valid Input Video Codecs:
MPEG-1, MPEG-2, MPEG-4, H264V
Valid Input Audio Codecs:
MPEG-1, MPEG-2, MPEG-3, AC3, MPEG4A (H264A), Vorbis (only for Ogg)
Mux Cache (ms):
200
What's this?
Video Settings
Video Transcoding Format:
HD-SDI 1080p 29.97 (1920x1080, 29.970 fps)
Deinterlace (will double the framerate):
Auto
Audio Settings
Audio Transcoding Format:
HD-SDI
HD-SDI Audio Configuration:
Embedded
HD-SDI Audio Rate (Hz):
48,000 Hz

The decoder quad was able to accept and send out all 4 demuxed feeds via HD-SDI and displayed each video in its corresponding quadrant

The biggest challenge in this whole process is receiving all 4 quadrants in perfect sync -- because if one is off by one millisecond, the whole video will not be appealing to the eye and will actually cause discomfort. There are ways to adjust the delivery latency if there is an issue but that should be last resort if possible.

From the 4 HD-SDI outputs of the D-Streamer Magnus you could feed a device like a Sony MPE 200, AnalogWay Ascender, or Vista Systems Spyder that will do edge blending for larger 4k displays like the JVC PS-840UD Or to Projector and other display devices.

Recently AJA released their Hi5-4k which is a 4 HD-SDI to 1 HDMI converter. So this is a perfect device to add to this workflow. Attach this to the Quad HD-SDI Decoder and convert to 1 4K HDMI that can be used with a variety of 4K TV's or monitors.

Conclusion:

Streaming of 4k content is now possible provided you have the necessary bandwidth. This test scenario worked well for the quad split 4k's off camera's and other devices. So this same workflow would be useful with the Sony F55, F65, BlackMagic Design and some of the Red Digital cameras. The above workflow is great for closed circuit distribution or dedicated point to point systems. 30-40 mbps or higher isn't practical for most applications. However delivery content from special events is obviously affordable and plausible.

With a Multiprogram recorder / player you will be able to record at the encode side or at the receive side. So this would be a modified workflow. The Quad Encoder (with mux) will send the muxed feed to the MPTS recorder player. The Recorder / Player will act also as a gateway for the decoder. The Live feed will be recorded and then the live feed will be passed through to the decoder. After a period of recording, use the player/record with built in scheduler to send the previously recorded content to the decoder via the same IP address / streams it was already receiving. With the recorder at the receive side, once the recording is playing back bandwidth can saved. Only use the bandwidth when you need it and send to the 4K Decoder locally. The whole idea is to keep everything in sync and to have guaranteed delivery.

For delivering 4k point A to point B when you have the proper bandwidth this is fully doable. For example maybe a church or a concert venue needs to have a satellite facility or the venue wants to be able to display extra content or the feeds throughout the location. This could be sporting events, concerts, education or any place where you need to share the feed on a giant display at really high resolution (cinema resolution).

For general public distribution 4K is still a bit off. The bandwidth requirement is a bit high and you would want a special player that can take the quadrants and blend together. This is possible but for general 4K for public use that is where H.265 is going to be the selling point. It will give you extra tighter compression and the bandwidth requirement won't be as large. HEVC/H.265 is designed to be encoder heavy but light decode and is expected to drop the bandwidth needed between 30-50%. Plus in the near future you have to expect more bandwidth to the door.

However for niche point to point workflows this is very doable today. Just a private dedicated line or massive bandwidth will be the sticking point but that is all addressable and possible.

This camcorder was produced in limited quantity and is offered to Japanese domestic customers for 1,700,000 Yen ($17,440). There is no plan to bring it to the US market.