THE BROADCAST BRIDGE

Connecting IT to Broadcast

Video Over IP - Making It Work

Essential Guide

EG

ESSENTIAL GUIDES

Sponsored by Riedel Communications

Introduction from Arne Bonninghoff, Head of IP Research, Riedel

Hello, and thank you for downloading this second installment of the Essential Guide to Broadcast IP Infrastructures.

After the first installment built a solid foundation in Audio over IP, this ebook will lay out some of the basic principles of Video over IP. The first article talks about SMPTE 2110 and the PTP timing protocol. Next, we address timing and jitter. The third article explores network capacities and buffering. Finally, we present a distillation of a technical paper that I wrote last year about NMOS for automated discovery and connection management.

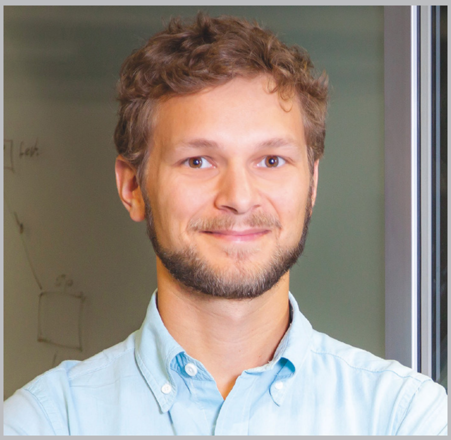
Here at Riedel, we strongly feel that NMOS IS-04 and IS-05 are key factors to the widespread adoption of IP. Currently, it is still quite difficult to create entirely IP-based broadcast infrastructures. Once they are implemented, additional technologies must come from a single vendor to have any chance at an easy install. We all look forward to the day when we can plug any new piece of gear into a network, have it instantly and automatically identify itself, and then make its I/O known to the network.

And, the best news is that it's not too far away!

I hope you enjoy this next chapter in the Essential Guide to Broadcast IP Infrastructures and encourage you to share it with your colleagues. Only through sharing knowledge will we push the technology forward.

Best regards,

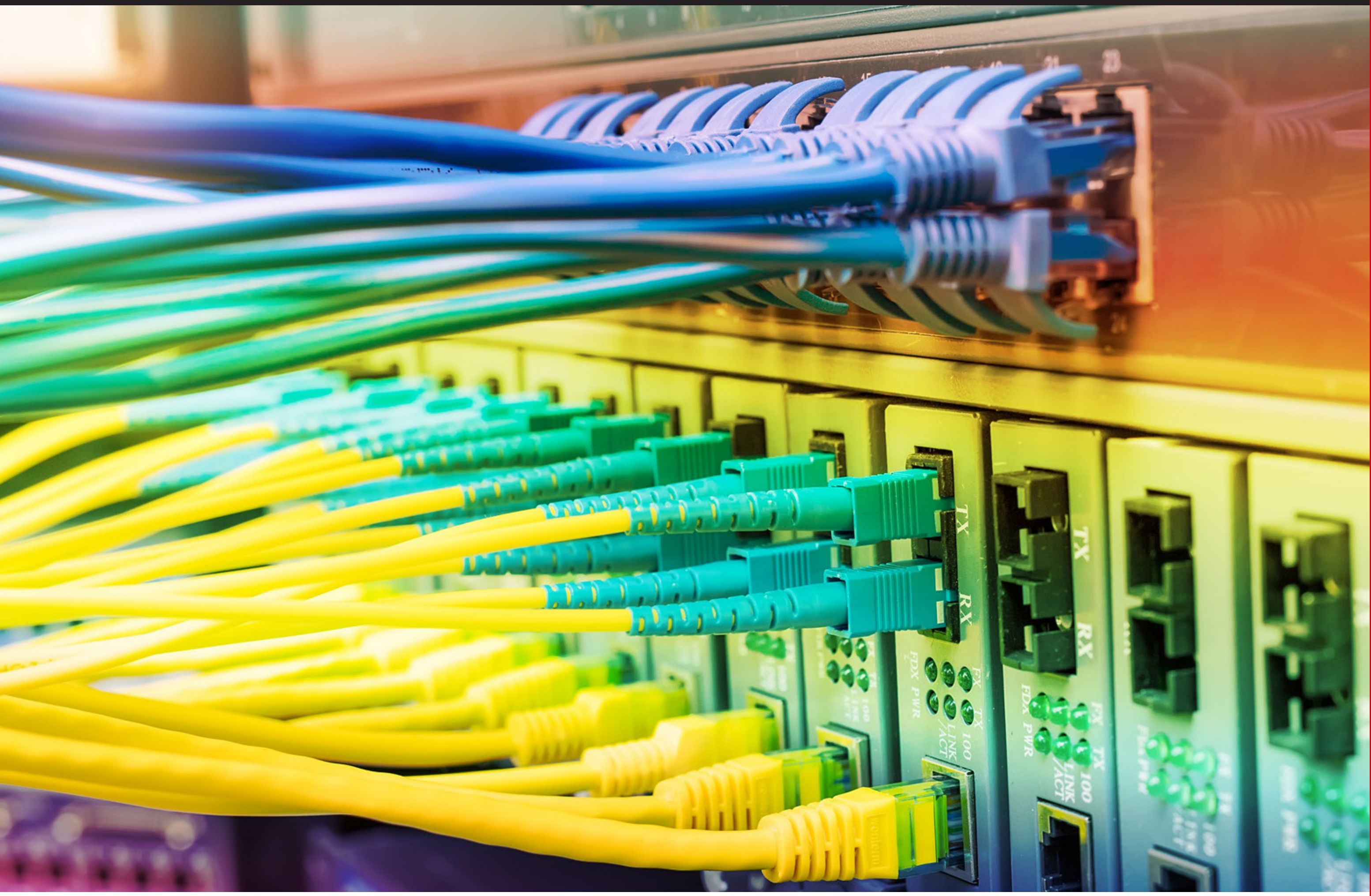
Arne Bonninghoff



Arne Bonninghoff, Head of IP Research, Riedel Communications.

Sponsored by Riedel Communications

Video Over IP - Making It Work



By Tony Orme, Technology Editor at The Broadcast Bridge

Part 1

For the first time in the history of live television we can now abstract away the video, audio, and metadata streams from the underlying hardware. This innovation presents unprecedented opportunities empowering broadcasters to deliver flexibility, scalability, and highly efficient workflows.

SDI is a synchronous system and is intrinsically tied to the underlying hardware to maintain clock and data accuracy. Consequently, extracting the video from the stream is complex and restrictive as it requires special hardware interfaces dedicated to SDI distribution and unique to broadcasting.

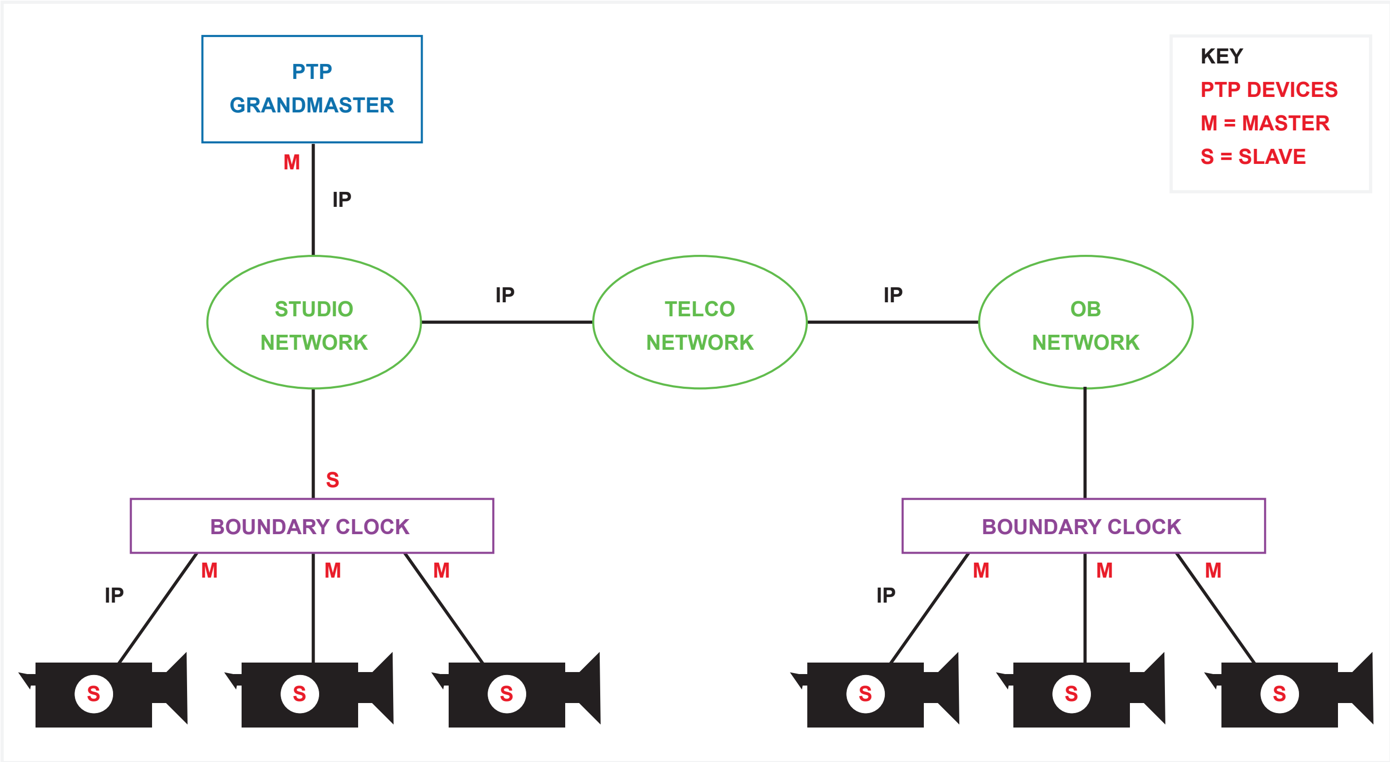


Diagram 1 – PTP synchronizes broadcast equipment in IP systems replacing SPG's in traditional SDI infrastructures.

SDI Lip-Sync Errors

Maintaining SDI audio-video synchronization has its own challenges, especially when the audio is distributed independently of the SDI video. Even embedded audio can lose lip-sync when the signal is processed by a frame synchronizer. Audio and video are often separated by the synchronizer and if the correct delay has not been applied to the audio, then lip-sync errors can easily occur.

IP generally uses asynchronous networks, such as Ethernet, to distribute packetized data. This is true of video, audio, and metadata. Packets leaving a camera are transferred across an Ethernet fiber to a switch. At the point where packets leave the camera, the traditional SDI timing is lost.

SMPTE ST2022

Initially, SMPTE provided the ST2022 specification. ST2022-6 packetized blocks of SDI data and grouped them into UDP datagrams to allow distribution over an Ethernet network. ST2022-6 maintains the TRS (timing reference signal) information from the SDI network, so the packets are easily reconstructed by the receiver into the original video, audio, and metadata streams.

Although ST2022-6 is reliable and is in use in many installations throughout the world, it is wasteful of precious bandwidth as it maintained the line and field sync information and didn't take full advantage of the opportunities IP networks offer. However, ST2022 was an effective and safe step into IP for broadcasters until ST2110 became available.

SMPTE ST2110

SMPTE's ST2110 family of specifications was released in the fall of 2017 and is the real game-changer, providing full utilization of IP networks. When packetizing video frames, ST2110 removes the TRS so redundant timing data is no longer distributed in the packet. Active video is encapsulated in a datagram, which in turn is appended with a unique timestamp accurate to a few nanoseconds. Receivers can reconstruct the video frame using these timestamps.

To maintain the demanding levels of timing accuracy needed to ensure the optimal viewer experience, SMPTE have mandated the use of IEEE's 1588 Precision Timing Protocol V2 (PTP). Industry has been using PTP for many years to synchronize machinery used on precision manufacturing production lines.

In a true IP system, PTP replaces the traditional black-and-burst sync-pulse-generator (SPG). But SPG's are still needed in hybrid systems where SDI and IP co-exist, or to support legacy broadcast equipment.

PTP Replaces SPG's

PTP timestamps can be thought of as a continuous counter that is incrementing every nanosecond. The absolute value is referenced to the epoch-time at midnight on 1st January 1970. In other words, the timestamp is the number of nanoseconds that have elapsed since the epoch. When a video, audio, or metadata packet is created, a PTP timestamp is appended to it, enabling the receiver to know exactly when the packet was created to allow it to reconstruct the frame of video, samples of audio, or reference the metadata.

Providing PTP timestamps creates unbelievable opportunity for broadcasters. No longer are we constrained by the underlying timing dictated by the synchronous networks of SDI, AES, and MADI. We no longer think in terms of video line and frame timing but can now think in terms of "events " or "grains". Each video frame or audio sample is a grain with a unique timestamp. It is possible to collect many different grains via different protocols or networks together using PTP as an absolute time of origin reference.

PTP uses a master-slave architecture to achieve a synchronous time reference throughout a LAN and WAN. All master clocks require an oscillator, but they vary in their accuracy. For example, a GPS referenced clock is more accurate than an NTP (network time protocol) referenced clock. To maintain synchronous timing, all slave clocks within the network will synchronize to the master.

No More SPG Changeovers

The Best Master Clock (BMC) algorithm, part of the PTP IEEE 1588 specification, negates the need for the A-B SPG changeover used by broadcasters in SDI systems. Several masters may exist in one network to give redundancy. But BMC allows them to identify the most accurate time source.

If we consider a network with two master clocks, master-A and master-B, and both GPS locked. The system administrator will need to set their priorities to be different. Master-A would be priority "0", and Master-B would be priority "1". If master-B is powered up before master-A, it will listen out for Announce messages on the network, as there won't be any, it will assume itself to be Grand Master and start periodically sending its own Announce messages. The Announce message contains much information about the clock, including its accuracy and its priority, in this case "1".

When master-A comes online it also starts to listen and receives master-B Announce messages. Master-A determines its priority is higher because it is "0" and starts to send its own Announce messages containing priority of "0". Master-B receives this message, accepts master-A has a higher priority, goes into listen mode, and stops sending announce messages.

All slave clocks on the network will receive these messages and automatically select master-A as their new time source.

If master-A was to lose its GPS reference, then it's accuracy would be degraded, and it would update its Announce message with this information (at this point it's still Grand Master). Master-B will be continually receiving the messages and determine that it now has a more accurate clock, thus assuming Grand Master status, and start sending Announce messages. Master-A would also receive these messages, acknowledge master-B is more accurate, stop sending Announce messages, and go into receive mode.

All slave clocks on the network will receive these messages and automatically select master-B as their new time source.

Use PTP Enabled Switches

Although PTP master clocks tend to be hardware devices, end points, such as camera's, sound consoles, and playout servers, can all use software solutions to sync to the master. The quality of the end users network interface card (NIC) is important as the buffers within the NIC influence the delay of the PTP messages and affect how well the device locks to the PTP master.

To maintain PTP accuracy and keep timing jitter low, PTP-enabled switches must be used wherever possible. IEEE 1588 provides a system to update PTP messages with the delay incurred in the switch, thus enabling slave devices to take the time spent in the switch into consideration.

Not all switches are PTP aware. If they are used, PTP messages may be randomly delayed and the slave devices syncing to the master will experience clock jitter and offset from the master time. To rectify this, delay may need to be added to the received video, audio, or metadata streams, resulting in unacceptable delays for operational staff.

Improved Frame Accurate Metadata

High Dynamic Range (HDR) is taking the broadcasting industry by storm. But to fully provide an immersive experience for viewers, frame accurate metadata must be created, processed, and broadcast to viewers. Compliant TV's use this frame-accurate data to dynamically configure their screens to provide the most optimal viewer experience possible.

In SDI systems, creating and maintaining frame-accurate metadata from a camera all the way through the production and transmission chain, is a complex and challenging task. A system would be awash with SDI embedders and de-embedders, along with multiple interface systems to convert, delay, and package the data accordingly.

IP networks, specifically using ST2110, allow broadcasters to create frame-accurate metadata and maintain its timing relationship to video and audio throughout the studio, production, and transmission chains.

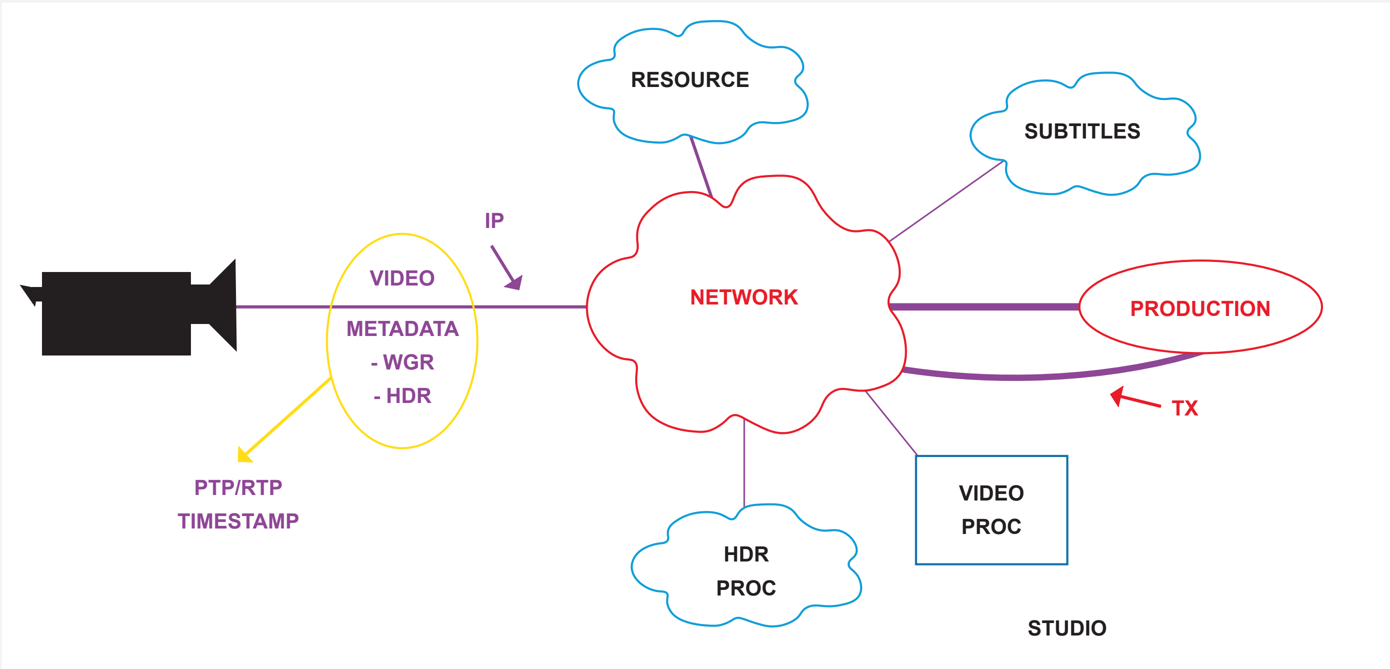


Diagram 2 – Using ST2110 allows broadcasters to process video, audio, and metadata independently of each other to provide flexible workflows.

Thinking of video, audio, and metadata as event-timed data packets allows broadcasters to process the data streams anywhere they like. Leaving them open to new and more efficient working practices, on-prem and off-prem datacenters, and Cloud infrastructures suddenly become available to creatively process streams, assuming network pipes are fast enough.

IP Provides Efficient Workflows

Creating multilingual subtitles traditionally requires language specialists to work at a broadcaster's studio. A Scandinavian broadcaster based in London might require native speakers from Sweden, Norway, and Denmark to work at the studio, clearly a massive expense.

Using ST2110-IP, the subtitling linguists could work from their respective homes with just a simple internet connection.

A compressed proxy stream of the broadcast would be sent to them with the original PTP timestamps, and they could use a simple PC/MAC computer to subtitle the program. The software would be able to stamp each subtitle with the associated video using the PTP time stamp.

REMI (remote-integration model) "At-Home" or backhaul outside broadcasts are easily achieved due to the adoption of ST2110. Rather than send a complete crew and production team to a stadium for a sports event, broadcasters just dispatch the essential camera and sound operators. Video, audio, and any associated metadata is streamed back to the studio over IP circuits provided by Telco's allowing the whole production to take place at the studio.

This has an obvious advantage in terms of accommodation and subsistence costs for crews. However, by switching the IP circuits from subsequent stadiums to the studio, one studio crew can cover several football matches or events.

Imaginative Workflows

Through the adoption of ST2110, broadcasters can unleash the power and opportunities IP networks provide for modern media distribution and broadcasting. Never in the history of television have we had the freedom to process, distribute, and monitor frame accurate video, audio, and metadata independently of each other and the broadcast infrastructure. Now, we are only limited by our imagination.

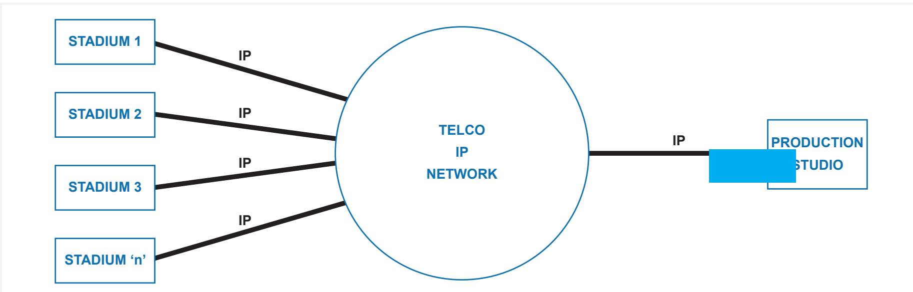


Diagram 3 – Using remote-OB techniques, broadcasters can now service many sports events from one studio to save on crew and deployment costs.

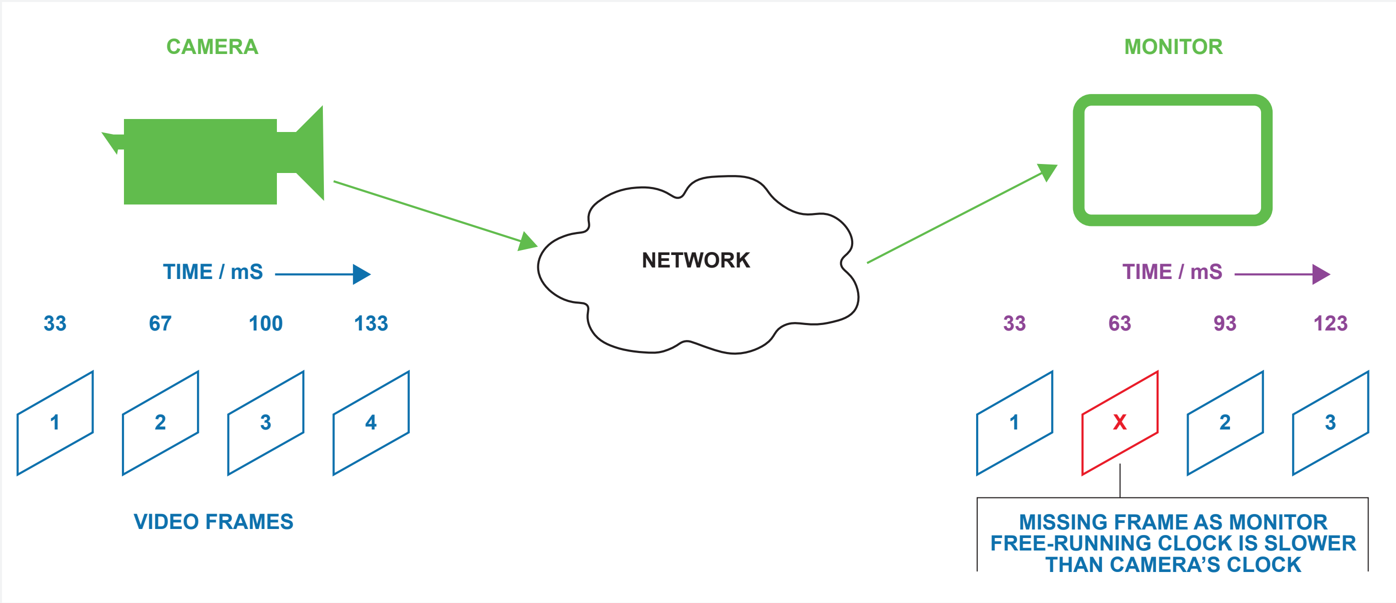


Diagram 4 – if the monitor is not phase and frequency locked to the camera, then frames will either be lost or duplicated. If excessive jitter occurs, both loss and duplication will happen causing the picture to be unstable.

Part 2

Timing is the most fundamental aspect of broadcast television. From the early days of tube cameras, to modern OLED displays, timing continues to play a crucial role for maintaining an optimal viewer experience. Picture stutter and breakup are just a few of the symptoms of synchronization and timing errors.

Broadcast systems have relied on synchronization since the first television services were transmitted in the 1930's. Line, field, and frame pulses were used to keep cathode ray television screens synchronous with the image created by the scanning tube camera.

Accurate lip-sync was assumed as there were no appreciable delays created by processing equipment to destroy the timing relationship. But this all changed with the advent of signal digitization.

Processing Lip-Sync Errors

In the early days of digital processing, the adoption of frame synchronizers witnessed the first lip-sync errors. However, they were confined to the locality of the frame synchronizer so could be easily fixed using digital audio delay. The same lip-sync errors followed compression, with the amount of delay just increasing.

Where video and audio were distributed together, embedded SDI was used to maintain the video-audio timing relationship. The underlying technology of all video and audio digital distribution at this time relied on bit-clocks being embedded in the transport signal itself, thus establishing and maintaining sample synchronization between the sender and receiver.

Packet switched networks, such as IP, destroy this relationship. It's the fundamental price we pay for the flexible, scalable, and highly efficient workflows that IP offers.

Unlocked Oscillators Cause Instability

If the monitor oscillator was running slightly fast with respect to the camera, then there would not be enough video samples entering the monitor and the picture displayed would become unstable. The opposite is true if the monitors oscillator was running slow; there would be too many samples and the monitor would not be able to display them all.

A difference of just 30Hz on an SDI connection between the camera and monitor oscillators could result in one sample per frame being lost. If this occurred during frame blanking, the picture would become unstable and unusable.

Relying on free-running oscillators is guaranteed to make broadcast systems catastrophically fail.

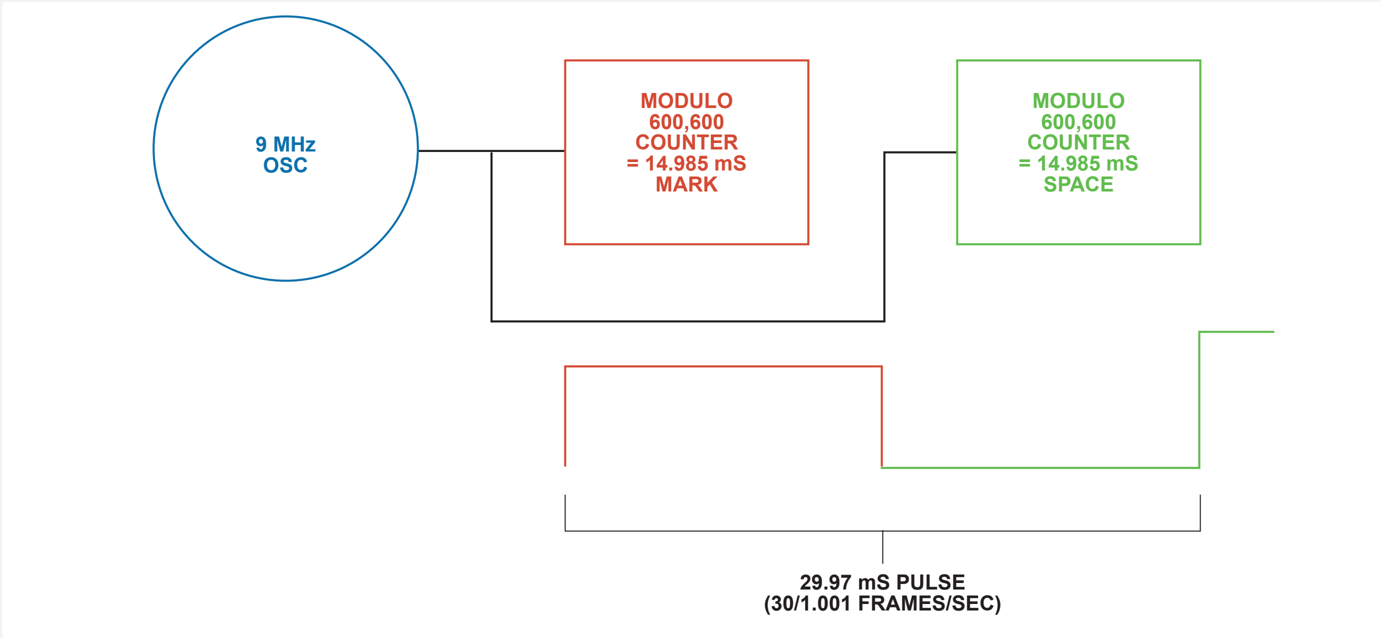


Diagram 5 – digital oscillators achieve lock by changing counter values to provide a cyclical waveform. Feedback filters are used to dampen the rate of change of the values so that the system is stable and converges to the desired frequency. This diagram provides an NTSC frame pulse, changing each of the mark-space counter values to 720,000 will provide a PAL frame pulse at 25Hz.

Enemy Jitter

Clock jitter is the enemy of distributed synchronous systems. Short-term jitter causes the frequency of the slave oscillator to change quickly, and long-term jitter causes drift.

Digital master-slave clocks are analogous to phase locked loop oscillators used to synchronize color sub carrier frequencies in NTSC and PAL analogue transmissions. A feedback mechanism compares the television oscillator and the incoming signal from the broadcaster. This creates an error voltage which is used to vary the frequency-controlled oscillator. Filters dampen oscillations in the error voltage to keep the system stable.

Although digital filters do not change the frequency of their local oscillator, they do create a periodic pulse to form the slave clock synchronized to the master. Counters vary the mark-to-space ratio of the slave clock to correct its phase and frequency to obtain convergence.

Digital filters are used extensively in feedback mechanisms for slave clocks to reduce both long-term and short-term jitter. But there is a price to pay; if the filter time constants are too long then the slave will take considerable time to sync-up and may never reach convergence. And if the filter time constants are too short, then the slave clocks will behave erratically resulting in unstable pictures and distorted sound.

Buffering Masks Jitter

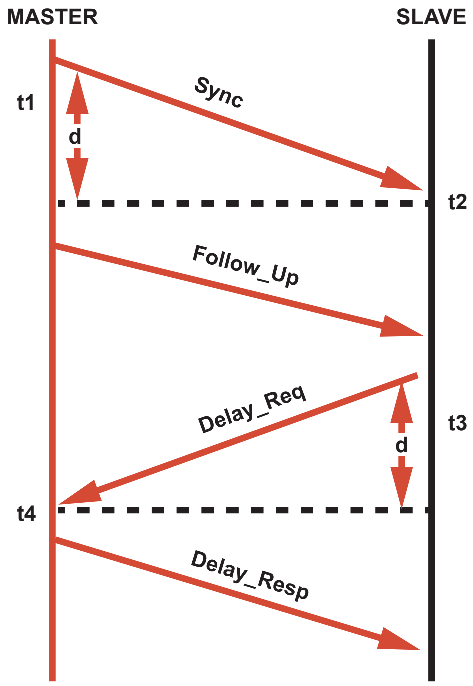
One more tool is available to help overcome jittery synchronization and that is buffering. Writing input samples into a buffer gives some breathing space for the output side to read the data at a constant rate. If the long-term frequencies are synchronized and correct, the sender and receiver will have exchanged video and audio data happily. However, if the buffer windows are too long, the video, audio, and metadata will be significantly delayed causing lip-sync issues, or unacceptable operational delay.

SMPTE adopted the IEEE 1588-2008 standard to synchronize clocks between devices on IP networks for their ST-2110 specification. Otherwise known as PTP (Precision Time Protocol), SMPTE borrowed this standard from industry as they had already been grappling with the issue of precision synchronization for many years.

Epoch is Key

Using well designed Ethernet networks, it is possible to achieve sub-microsecond accuracy using PTP. Coherent-phase alignment between devices is achieved using the Epoch time; a unique timestamp value available every nanosecond, referenced to the beginning of the Epoch.

Initially, the accuracy of the PTP network relies on an ordinary clock assuming the role of the Grand Master clock. The highest accuracy clocks available are atomic maintaining an accuracy of 1nS, but these are generally not available to most broadcasters and so they lock-to GPS satellites with onboard atomic clocks. The most reliable PTP-GPS clocks can achieve accuracies of < +/- 40nS.



$$\text{Delay (d)} = \frac{(t2 - t1) + (t4-t3)}{2}$$

$$\text{Offset} = \frac{(t2 - t1) - (t4-t3)}{2}$$

Message Type	Description	Slave Knows Values Of
Sync	Master initiates process and sends “sync” to slave. Slave measures time it receives the sync message (t2). Master can only accurately determine the time of the sync message (t1) after it has left.	t2
Follow_Up	This message sends the data value of the exact time (t1) “sync” left the master.	t1, t2
Delay_Req	Combined, these help to determine the propagation delay of the network. PTP assumes this to be the same in both directions.	t1, t2, t3
Delay_Resp		t1, t2, t3, t4

Diagram 6 – PTP clock synchronization.

The PTP Grand Master transmits “sync” messages approximately once every second. The send frequency varies depending on the type of network used and the speed of pull-in required in the slave devices. These messages include much data, but critically include the number of seconds and nanoseconds passed since the start of the Epoch.

SMPTE’s ST 2059-1:2015 “Generation and Alignment of Interface Signals to the SMPTE Epoch”, defines the reference to be midnight, January 1st 1970, International Atomic Time (TAI). Audio samples, video frames, and metadata is appended with a timestamp value that represents the number of seconds and nanoseconds that have elapsed since this time.

Using this elapsed time, we can determine the day, month, year, hour, minute, second, and fraction of a second, anywhere from January 1st 1970 to now, with an accuracy dependent on our Grand Master.

Low Jitter for High Accuracy

The aim of network synchronization is to make the Grand Master and all attached slave clocks contain values that describe the same number of seconds and nanoseconds elapsed since the Epoch at any one time. The accuracy with which this can be determined is based on the resolution of the master and slave counters, and the amount of jitter the network introduces into the timing messages.

Although simply sending time messages from the master to the slaves will provide some synchronization, it is a naive approach as the distance and delay between the slave and master is unknown. To correct this, three more messages are exchanged between the Grand Master and slave. Diagram 6 demonstrates how the protocol works.

When a slave first synchronizes to a master, its time values will be wildly different to the master, and a period of synchronization starts. Convergence may take any time between a few seconds to ten minutes, depending on the configuration adopted and network design.

The “1588 Default profile”, designed for general applications, uses slower message rates and might lead to long lock times. For the broadcast industry, devices need to lock to PTP quicker, to facilitate fast exchanging devices in a live production (e.g. camera). AES67 Media Profile and SMPTE 2059-2 profile uses faster message rates between Master and slave, enabling lock times of a few seconds.

Software PTP Causes Jitter

If the PTP stack is implemented completely in software. Without hardware assisted network interfaces, operating systems, IP stacks, and CPU response times all conspire to reduce the accuracy of the timestamps in both the master and the slave. Consequently, for broadcast applications, we will always use a hardware assisted PTP generator. If accurate time-of-day timecode lock is required or the master and slaves are geographically separated, the PTP generator will be locked to a GPS source.

Accurate slave devices, such as camera's, should provide hardware circuits in the Ethernet network interface card to extract and insert timestamps in the UDP/IP datagrams as close as possible to the physical layer of the network. Using this method removes timing errors and jitter created by the operating system, IP stack, and CPU response time.

Certain COT's Network interface cards also offer hardware-based timestamping for usage in server chassis, offering accurate synchronization for software applications running on standard servers.

Networks Must be Symmetrical

As diagram 6 demonstrates, networks must be symmetrical for the protocol to work correctly. That is, the time taken to send an IP packet from master to slave, must be the same as the time taken to send it back.

This is a reasonable assumption in well-designed Ethernet networks, however, this assumption is not necessarily valid in resilient routed networks as the datagrams may take different send and receive routes. The Delay_Req and Delay_Resp messages are averaged to determine the overall network delay. If one is longer than the other, the average will be skewed, possibly randomly, and excessive jitter will occur at the slave device.

In a broadcast network, there may be hundreds of slave devices, such as camera's and switchers, all receiving their timing information from one master. From diagram 6, we can see that each slave must send a Delay_Req message back to the master in response to the Sync and Follow_Up messages. This has the potential to overload the Grand Master CPU causing time inaccuracy and jitter.

Application	Layer 3		Layer 2
	DiffServ	DSCP	COS
Routing	CS6	48	6
Clock	**EF**	**46**	**5**
Media	**AF41**	**34**	**4**
Conferencing Video	CS4	32	4
Mission-Critical Data	AF31	26	3
Call Signaling	CS3	24	3
Transactional Data	AF21	18	2
Network Management	CS2	16	2
Bulk Data	AF11	10	1
Scavenger	CS1	8	1
Best Effort	**0**	**0**	**0**

Diagram 7 – Mapping of DSCP to COS according to recommendations in AES67:2018. PTP Traffic marked with DSCP 46 will use a higher queue (e.g. 5) than the Audio or video traffic, marked with 34.

Additionally, standard switches might queue PTP messages to prioritize other messages, resulting in a less-than-ideal delay measurement.

QoS

PTP messages need to be prioritized over all other messages, as correct timing is crucial for any other media (Audio, video, control) to function correctly. Quality of service queues configured in the network devices help enforce this.

Usually, switches capable of IEEE 802.1p can sort all incoming packets into a maximum of seven queues before leaving the switch on another port. Like boarding a plane with priority permission, packets with Class of Service (COS) markings can traverse the switch faster than others.

In IP Media networks, DSCP (Differentiated Service Code Point) markers in the IP Header can be used to mark PTP packets created in Master or slave. Switches will then sort up to 64 different DSCP tags into the 7 queues offered in most switches.

Using higher DSCP tags for PTP than for Audio helps achieving accurate synchronization for audio networks, as switch introduced jitter is not extremely high, as the bandwidth utilization of Audio over IP is not significant.

Boundary Clocks Keep Jitter Low

To overcome network jitter in Video over IP applications, where data rates of multiple Gigabit per seconds occur, boundary clocks are used in the network to distribute the message load away from the Grand Master. PTP allows each port to act as either a master or slave. One port on the boundary clock will be configured as a slave type, connected to the Grand Master, and the second port will become the master for all the slave devices connected to it.

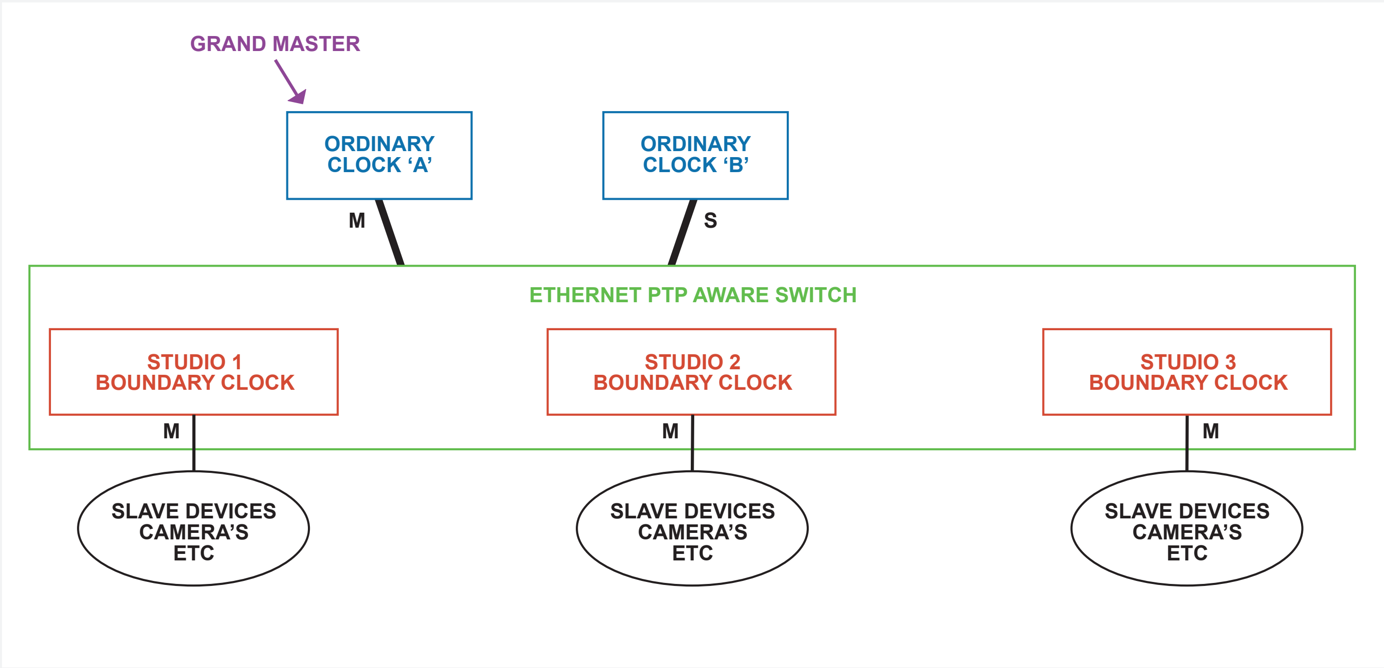


Diagram 8 – Operation of boundary-clocks to move protocol load from the Grand Master. In this case, ordinary clock 'A' is the Grand Master and ordinary clock 'B' is in slave mode. If 'A' was to degrade or be disconnected from the network, clock 'B' would become the Grand Master, as either it would no longer see "announce" messages from 'A', or 'A' would announce its clock had been degraded. All boundary-clocks would see these "announce" messages and switch over to clock 'B'.

The boundary clock must internally synchronize its own clock to the Grand Master to enable accurate jitter-free synchronization of the slave devices connected to it.

Switch Must be PTP Aware

Where boundary clocks are either not required or not accessible, then PTP provides transparent clocks. These are used in PTP-enabled switches to take into consideration the packet delay incurred within the switch to reduce the risk of excessive jitter.

Although transparent clocks are essential to keep jitter low, they do not provide timing information other than to update the delay fields in the sync messages. Therefore, they do not take any load off the master clock.

Achieving accurate PTP timing is critical for making distributed television work over asynchronous IP systems. Jitter is our enemy and we must be constantly vigilant when designing networks.
PTP frees us from the rigid constraints of SDI, AES, and MADI, empowering broadcasters to deliver new and more efficient workflows.

Part 3

Point to point connections dedicated to delivering video and audio signals have dominated the broadcast industry since the 1930's. But migrating to IP is compelling engineers to think differently about packetized signal distribution. In this section we investigate the potential sources of congestion and the effects of buffering.

Latency has been slowly creeping into broadcasting since the first frame-synchronizers were used to time remote cameras into studios. If the audio was correctly delayed and the lip-sync relationship maintained, broadcasters weren't too concerned about latency as the times involved were very small. But migrating to IP has delivered many new challenges and we must be vigilant when dealing with latency and jitter.

By the very nature of IP, data is divided into packets and moved through a network independently of any other packet. Many devices in a network will buffer data and as they do, they introduce unpredictable delay resulting in jitter and potential loss of data.

Not all devices within a network will operate at the same Ethernet speeds. A camera may use a 10GbE connection, an audio processor may use 100Base-T, and an IP Multiviewer might use a 100GbE. Assuming we're using a centralized switch topology, any 100Base-T devices will need to be converted to fiber to facilitate connection to the Ethernet switch.

Changing Data Rates

If we assume a scenario where a well-behaved camera with a 10GbE connection is creating an evenly gapped video, audio, and metadata stream, then converting the HDR metadata content with an average data-rate of 5Mbits/s from the 10GbE to 100Base-T is seemingly straightforward as solutions such as media-converters provide this.

Effectively, the media-converter is providing two functions, its gearing the clock speed from 10Gbit/s to 100Mbit/s and changing the physical layer from fiber to twisted copper pair.

To change clock speed, the complete received Ethernet frame must be loaded into a buffer before it can be clocked out at the lower rate. And the opposite is true when converting from 100Base-T to 10GbE.

The Detail is in the Burstiness

All buffers are a fixed length and if packets from the 10GbE are being written into the buffer faster than the data is being read out for the 100Base-T connection, then buffer overrun occurs. That is, the frame received has nowhere to go, so it gets dropped.

Average measurements in networks do not tell us much of what is going on. The real area of interest is in the tails of the distribution bell curve.

Although the overall data-rate may be within the specification of the PC, media converter, and 100Base-T Ethernet link, there could be lost frames due to burstiness and jitter as the frames may not be able to be processed quickly enough.

Measuring the burstiness of a connection is notoriously difficult and must be achieved using a hardware device. Network Interface Cards (NIC's) will buffer data as soon as it arrives on the physical connection, removing the temporal relationship between the frame and wire. A hardware monitoring unit will tag the frame with an accurate clock as it is received off-the-wire and before it is written to the buffer. The time-tag can be used to determine the exact burstiness of the data and provide some meaningful buffer management.

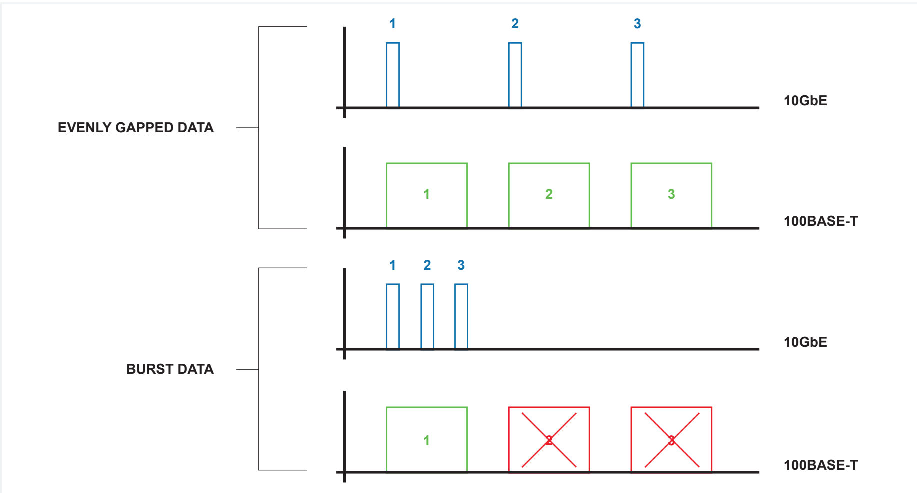


Diagram 9 – Each green and blue block shows the relative time duration to send consecutive frames with the same average data rate on 10GbE and 100Base-T connections. A media-converter will receive the frame into a buffer at 10Gbit/s, and then send it out at 100Mbit/s assuming the long-term average is less than 100Mbit/s. Evenly gapped data is easily converted to a 100Base-T connection from 10GbE without packet loss. But burst data shows frames 2 and 3 are lost on the 100Base-T connection if a small buffer is used. If a large buffer is used then all frames from the 10GbE connection will be correctly sent on the 100Base-T connection, assuming the long-term average is less than the capacity of the link.

Buffers to the Rescue

IT solutions regularly deal with burstiness and tend to average out the data using buffers. In the scenario described in diagram 9, a buffer that can store at least three frames would have resulted in no lost packets. However, latency would increase.

SDI, AES, and analogue switching matrices have no buffering, thus any input can be routed to any output with only a few nanoseconds of processing and propagation delay. But this is not always the case with Ethernet switches. In IT terms, the non-blocking switch is defined as a switch-fabric capable of handling the theoretical total of all ports, such that any routing request to any free output port can be established successfully without interfering with other traffic. Smaller and budget switches do not always facilitate non-blocking.

Delivery Without Collision

Switch-fabric, or just “fabric”, is the function within the Ethernet switch used to transfer frames from the ingress port to the egress port. The fundamental role of the fabric is to achieve switching without collisions or loss of frames.

The buffer management within a switch is one of the principle factors that governs frame throughput. Ethernet connections fundamentally differ from video and audio as an Ethernet circuit may be carrying more than one media-essence stream, whereas a video or audio circuit is usually carrying just one.

For example, the 10GbE IP stream leaving the camera will have frames containing video, audio, and metadata. The video output might consist of multiple HD and SD feeds for program and monitoring. Return video, talkback, tally, and lens control streams will all be moving in the opposite direction simultaneously.

No Buffers is Best

Ideally, switches would not use buffers at all and would just be one big silicon block of shared memory on a single chip supporting a thousand simultaneous serial connections. Commercially this is impractical as current technology only allows 100 – 200 simultaneous serial connections to a chip. Although building a device may be possible, the manufacturing yield rate would be prohibitively low and the expense would be enormous.

Switches regularly move frames from many ingress ports to one egress port. A quad-split viewer would require multicast feeds from four different cameras, all presenting on different ingress ports on the switch. The fabric must individually move each frame from ingress ports 1 to 4 for each camera, to the same egress port for the quad-split viewer. If all the frames simultaneously arrive at ports 1 to 4, the switch must store three of them and send each frame in turn, through the fabric, to egress output 1, otherwise frames would be dropped.

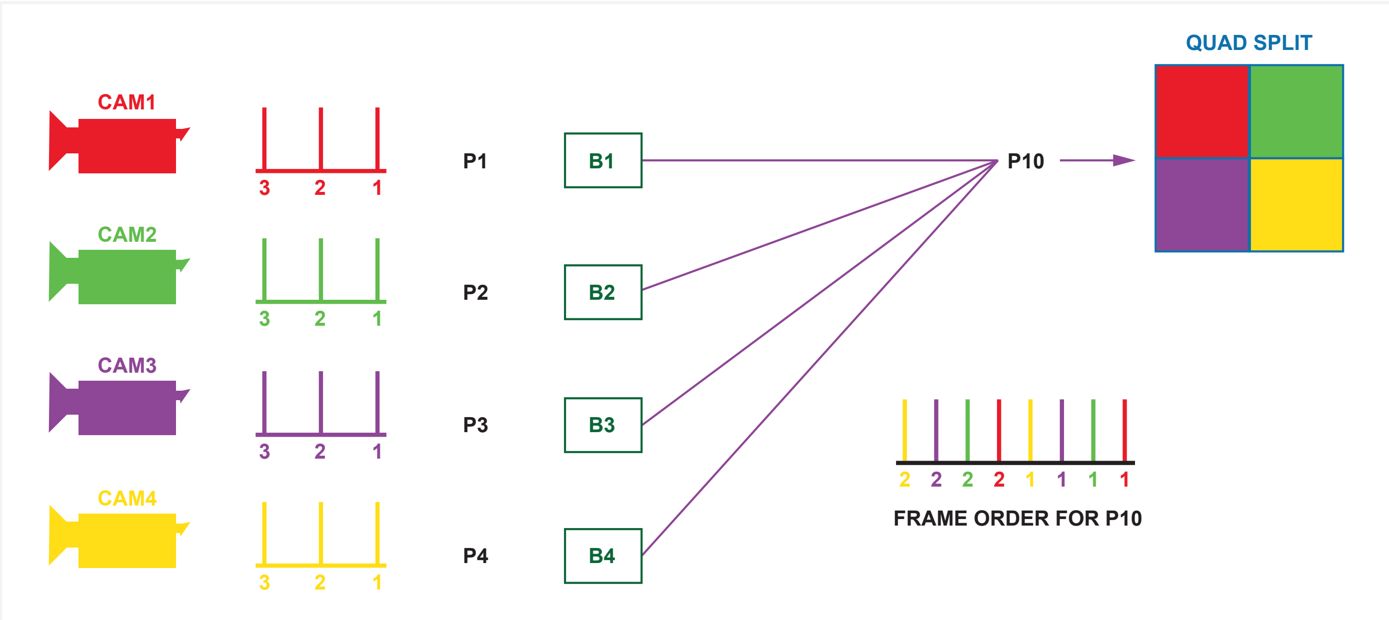


Diagram 10 – Cameras 1 to 4 present their frames to ingress ports 1 to 4. Each of the frame streams is being multiplexed onto the egress port 10. Many frames arrive at the same time, or close to each other, and if the egress port P10 was not empty, frames would be dropped. Buffers B1 to B4 provide temporary storage for the frame streams from the cameras so that they may be correctly ordered and presented as a single stream to the Quad-Split on P10.

To Understand why switches are blocking or non-blocking requires a better understanding of buffer strategies.

Placing input buffers on a switch seem the most obvious choice. If a FIFO (First in First Out) is associated with each ingress port, then frames presented to the port will be immediately written to the FIFO memory. Assuming the long-term average of the link was not breached, and the egress port capacity was not breached, then the FIFO would be able to deal with short term burstiness and not drop any packets.

Algorithms in the switch read the MAC destination address of the frame in the FIFO and determine which egress port to move the frame to through the fabric.

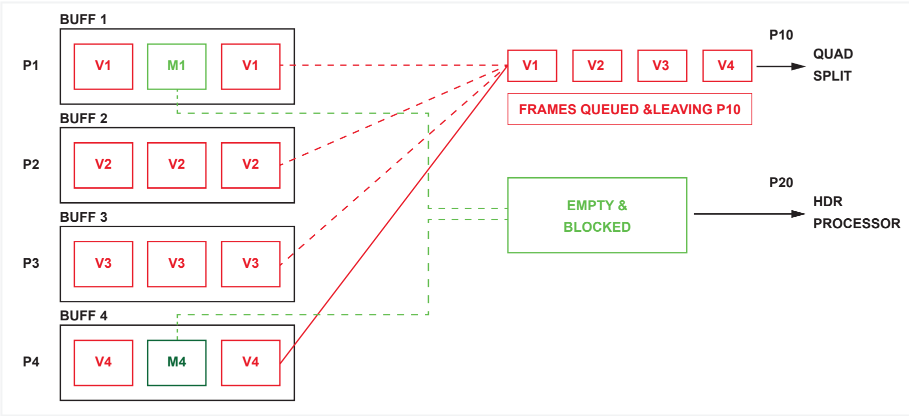


Diagram 11 – Ports 1 to 4 are receiving Ethernet frames for video (Vn) from cameras 1 to 4 with associated HDR meta-data (Mn). As Buffers 1 to 4 are FIFO's, M1 and M4 cannot be moved to P20, and hence processed by the HDR processor, until frames V1 and V4 have been moved to P10 for the quad split. Even though P20 is empty, and the link has the capacity to send frames to the HDR processor, it cannot, until V1 and V4 have been moved. In this scenario, M1 and M4 frames are blocked. HDR processing becomes dependent on the Quad-Split monitoring, this is unacceptable behavior and will result in HDR to video sync errors or dropped packets.

Head of Line Blocking

A fundamental problem occurs with this method. The Ethernet stream presented to the ingress port will contain frames with destination MAC's for different ports.

For example, each of the cameras on ports 1 to 4 would also contain audio which might be destined for port 11 going to the sound console. These frames will be queued behind the video frames and must wait until the video frames have been moved to their respective egress ports before the audio frames can be moved. If the video egress port is operating at a high capacity, then the audio frames will be blocked.

This phenomenon is referred to as Head-of-Line-Blocking (HLOB). FIFO's are cheap to implement and used in small switches, and many applications in IT do not require the expense of non-blocking architectures as they rely on higher protocols such as TCP to compensate for any dropped packets.

Broadcasters use UDP as they cannot afford the latency that TCP protocols present. In using UDP we've removed the protection TCP provides, especially as switches using ingress FIFO's drop packets during times of high congestion. Non-blocking switches use much more advanced buffer management solutions such as VOQ (Virtual Output Queues).

Intelligent Buffering

VOQ is an intelligent memory buffer. Each ingress port will have one VOQ buffer for each of its egress ports. For example, if the switch has 16 ports, each ingress part of the port will have 16 VOQ buffers associated with it. When a frame is presented to the ingress port, the VOQ algorithm determines the destination MAC of the frame and moves it into the VOQ buffer for that frame. When the egress port becomes empty, the VOQ algorithm will decide which of the ingress-VOQ buffers to move the next frame from.

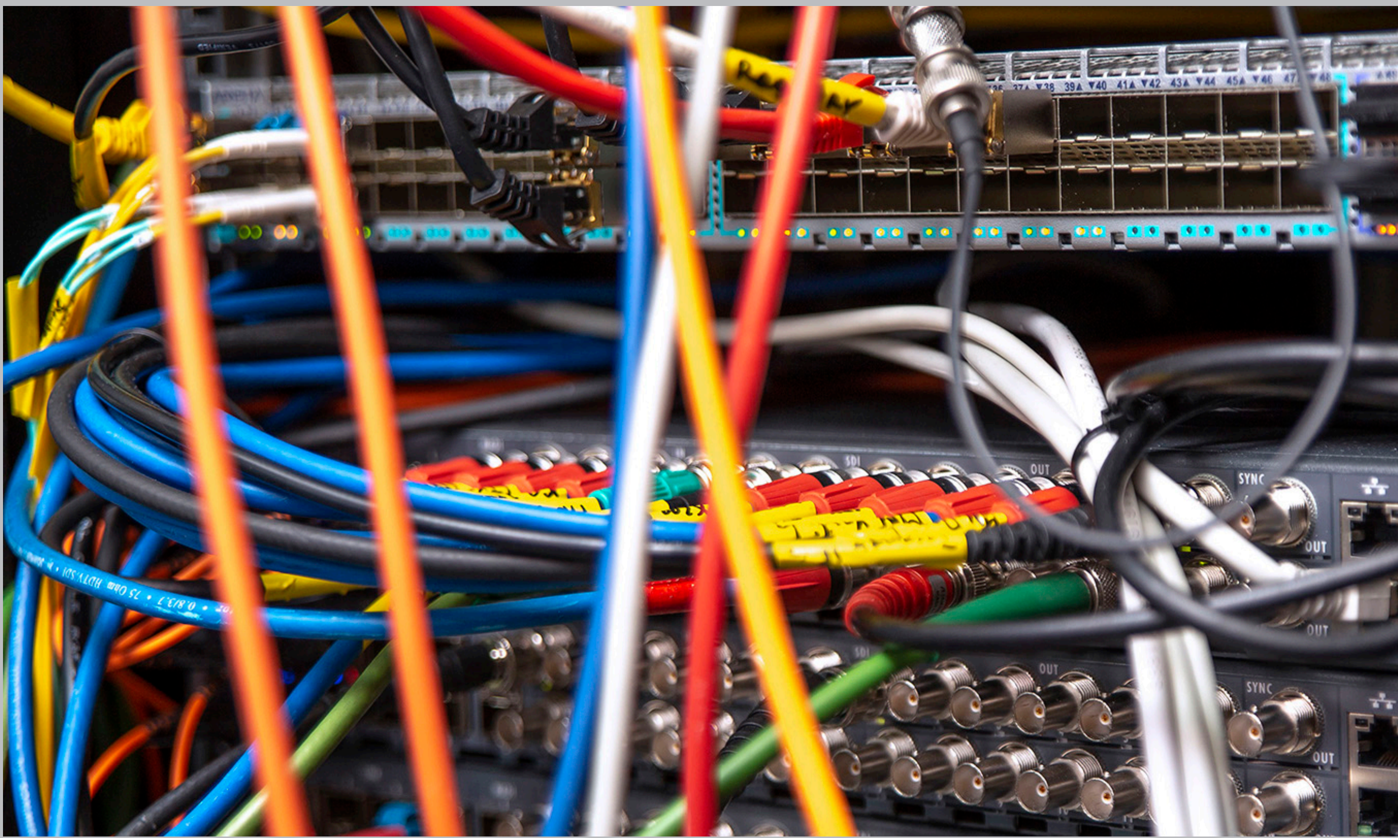
Due to the complexity and high bandwidth storage management needed to make VOQ work efficiently, it is only found in expensive, low latency, high end switches. VOQ also removes HLOB while allowing network administrators the option of applying QoS (Quality of Service). VOQ algorithms prioritize certain frames by applying priority weighting to buffers. This could be useful for prioritizing PTP traffic to and from cameras.

Timing Constraints

Calculating network capacities is not as straight forward as it may first seem. Burstiness can cause excessive packet jitter and even dropped packets. ST2110-20 provides tight timing constraints for burstiness and if they are not obeyed, downstream equipment may not be able to correctly display video and audio.

Riedel - The Sponsors Perspective

Standardized Connection Management for Essences and Network Flows in ST2110 and AES67



MediorNet MicroN nodes in Riedel's IP Laboratory.

With the release of the core parts of the SMPTE ST 2110 suite, the confusion around different transport standards for audio and video over IP is now settled. However, the adoption of these solutions for day-to-day use is still far from easy. While there are more and more pure IP facilities and OB trucks now in service, they take significant time to set up and are only practical when a single vendor interface is used.

This article will discuss the recent advances of AMWA NMOS and explain how IS-04 Registration and Discovery and IS-05 Connection Management can lead to plug-and-play-like installations. The use of IP devices and signals with automatic address assignment are key factors in this discussion.

Workflows: Old Versus New

In a baseband broadcast scenario, a Sender and a Receiver have always been individually identifiable by their physical connectors. Thus, the "one-to-one" connection between a Sender and a Receiver has always had an underlying, physical "one-to-one" connection. Assuming that the routing from the SDI output of a camera to the SDI input to a screen is fixed, an operator can connect source and destination simply by connecting a cable.

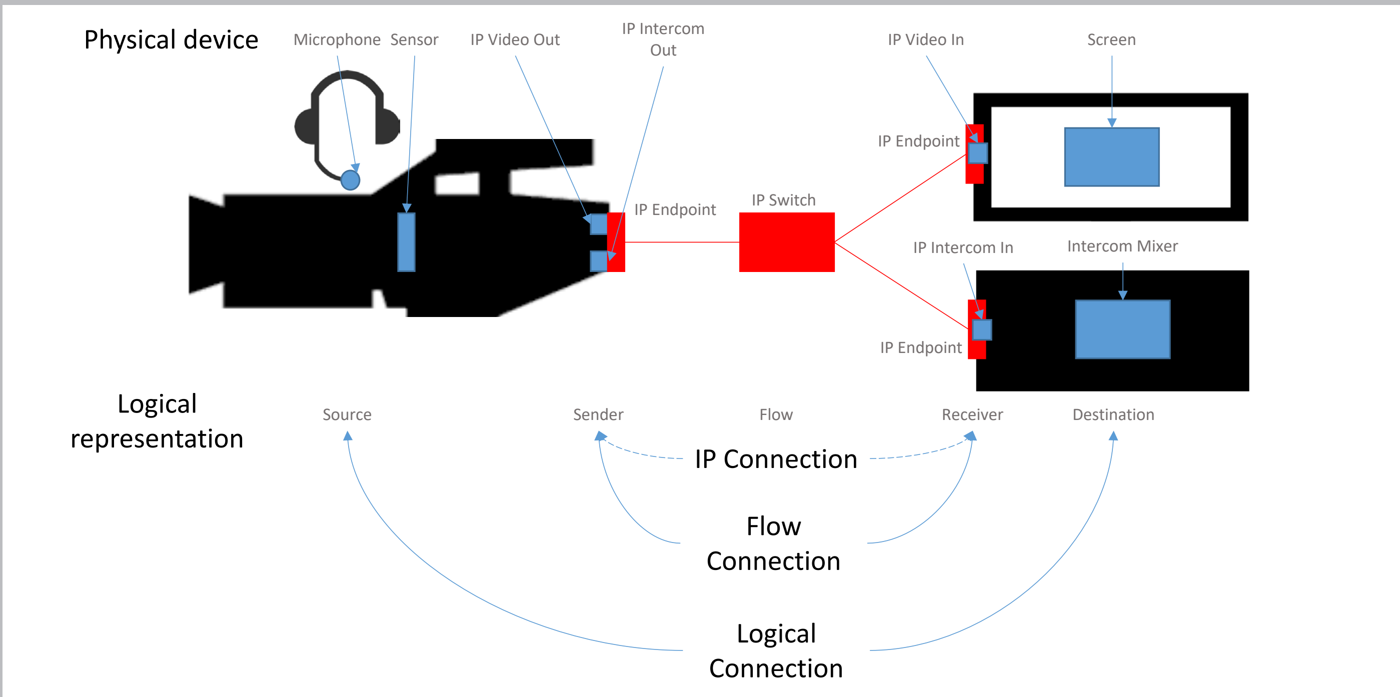


Figure 1: IP-based signal flow.

With IP, this relationship differs. Since the shared medium of Ethernet allows transportation of AES67 or ST2110-30, together with ST2110-20 or other traffic in parallel, a physical port must be treated separately from the essence Sender and Receiver.

Regarding the connection of a Sender and Receiver, the underlying physical connection can no longer be taken as an identifier. Instead, the identification of a Sender or Receiver now needs to be defined by a combination of the source IP address, the destination IP address and, in the case of 2110 or AES67 traffic, the UDP port*. But operators should not have to be bothered to type in IP addresses and take care to maintain them in a broadcast environment.

Similar to a PC that retrieves IP addresses automatically and is also identifiable by its unique hostname, IS-04 and IS-05 can allow automatic workflows for broadcast.

* UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) and is used primarily to establish low-latency and loss-tolerating connections between applications on the internet.

IS-04

The AMWA IS-04 data model currently describes human-readable identifiers that hold the information and configurations of a broadcast device. A device that implements the current open-source APIs will “advertise” all available “Sources” (Stage Announce), “Senders” (2110-30 Out 1), “Receivers” (2110-30 In 1), and “Flows” (uncompressed 24bit 48kHz). Overarching these elements is the “Device” (Artist Client Card AES67-108-G2) and the “Node” (Artist 64 Frame).

IS-04 offers three APIs:

- **Node API:** Exposed by every IS-04 Node and describes the above data model. Can be used to get information about a Node directly from the Node itself.

- **Registration API:** Used by a Node to register all resources of its current, particular data model to a central registration service. This API is exposed by a central service with database capabilities. Each time the configuration of the Node changes, for example, when streams are activated or deactivated, the Node will update its representation via the Registration API.

- **Query API:** Exposed by the same registration service and used by any control surface or user interface to retrieve information from any Node via a single interface. In this manner, the control systems do not have to query all Nodes on the network for information. They retrieve live updates from the central source of information.

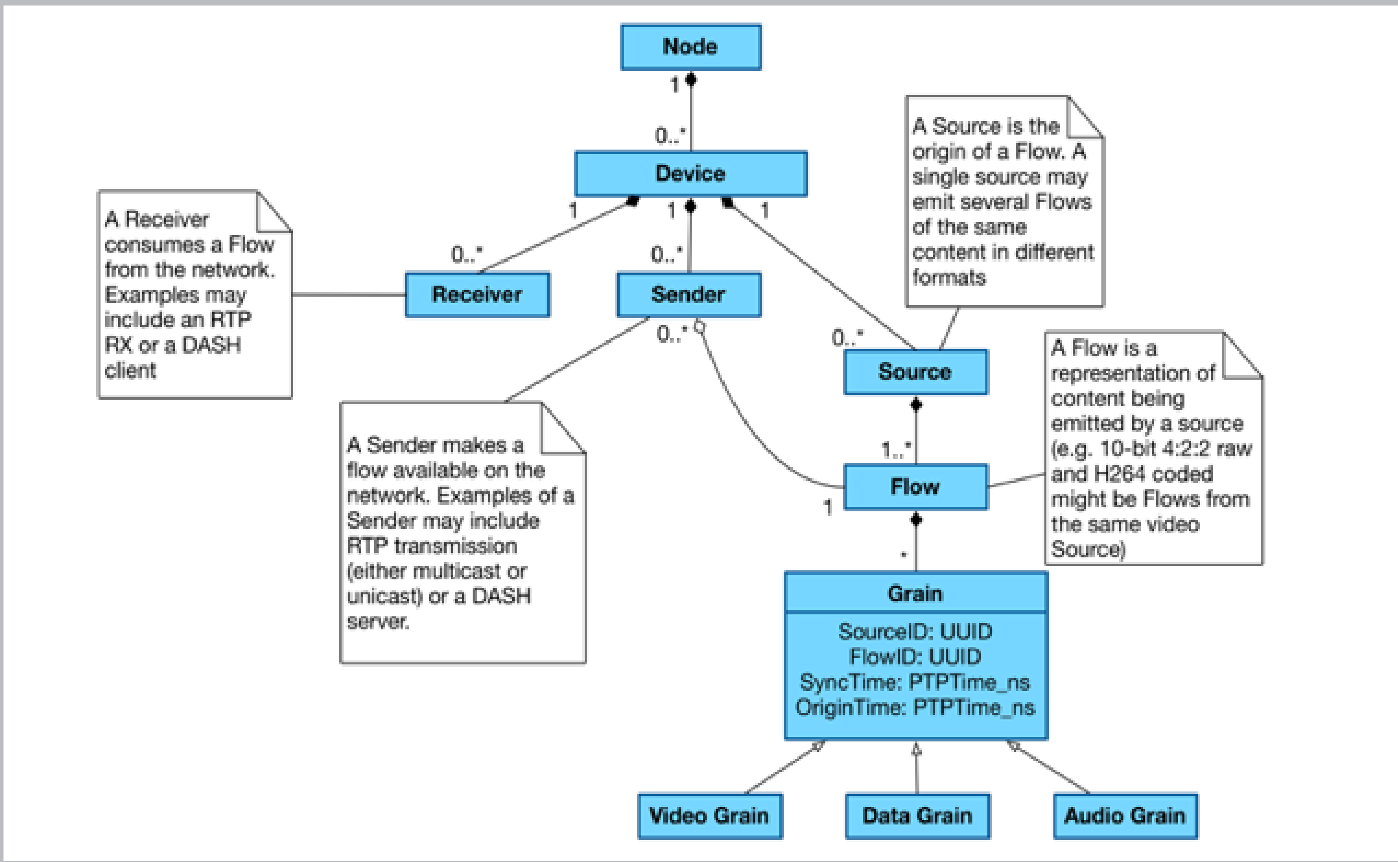


Figure 2: IS-04 Data Model.

By employing a control system that uses the IS-04 Query API, all IS-04 Devices can be discovered and modeled into a user interface, assuming that Nodes have already registered via the Registration API. The user interface can then list all available Senders and Receivers, similar to a physical patch panel of coax connectors. Riedel's NMOS Explorer is an example of one user interface and is freeware, available at **https://myriedel.riedel.net.**

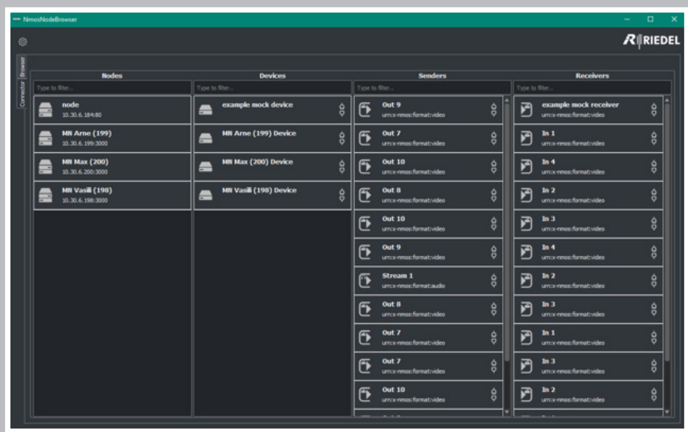


Figure 3 – Riedel NMOS Explorer showing the registered Nodes and Devices with the state of their Senders and Receivers after querying the central registration service.

To connect a Sender and a Receiver, the identification needs to be known. For dynamic connection management, it is also necessary to start and stop Senders and Receivers in order to avoid unwanted traffic. Additionally, a Receiver needs a description of the underlying flow to be successfully received and in order to reserve sufficient bandwidth.

IS-05

IS-05 specifies the transport Parameters of IP addresses and ports and allows them to be staged and activated via the same type of API as IS-04. As soon as a control system has knowledge of available Senders and Receivers, it can configure addresses, regardless of make and brand.

For the Flow description, IS-05 specifies a "transportfile" address. An HTTP GET on that address delivers the Session Description Protocol (SDP) of the underlying flow to the control system. The SDP needs to resemble the flow currently connected to that Sender and must be compliant with the underlying standard. For an audio stream following AES67, the SDP needs to comply with AES67:2015. For a video stream following SMPTE ST 2110-20, the SDP needs to comply with the SMPTE ST 2110 suite of standards. The API can also be adapted for future Senders that may use different transport mechanisms or compression simply by storing and altering the identification in the "transportfile." The transport parameters do not change when a flow is compressed.

By connecting a Sender to a Receiver via IS-05, a control system will retrieve the "transportfile" from the Sender and send it to the IS-05-compliant receiver. The Receiver is then configured for the stream about to be coming in.

The physical act of connecting a cable to a coaxial port in the IP world is described and executed by activating the "Master Enable" switch via IS-05. The receiver will issue an IGMP join, and the stream is connected.

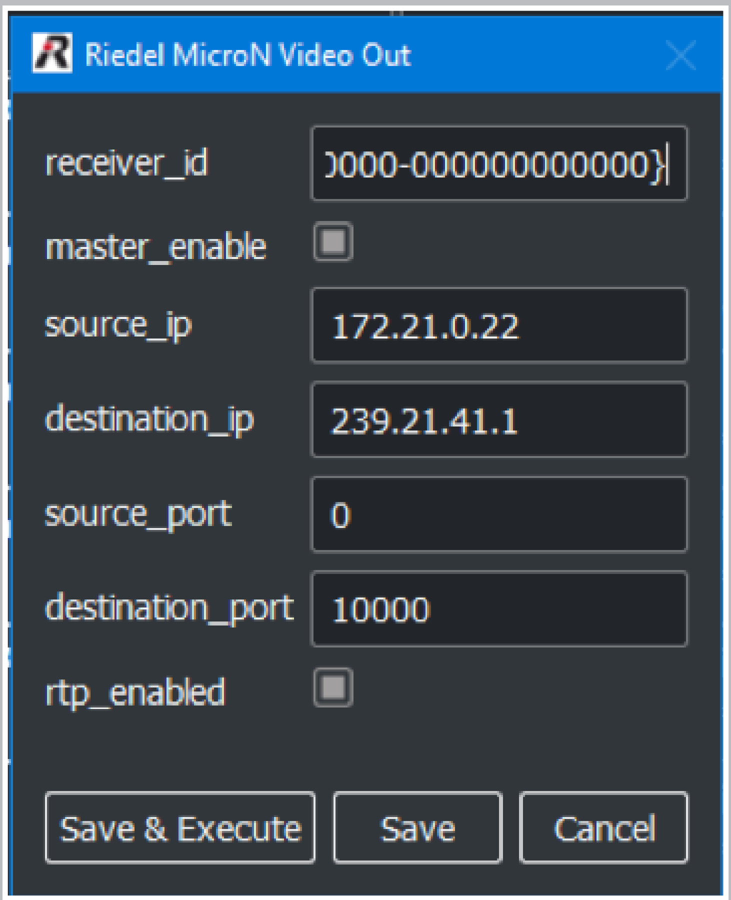


Figure 4 – Staging and activating Source IP, Destination IP, and Port via the NMOS Explorer on a Riedel MediorNet MicroN IP Output.

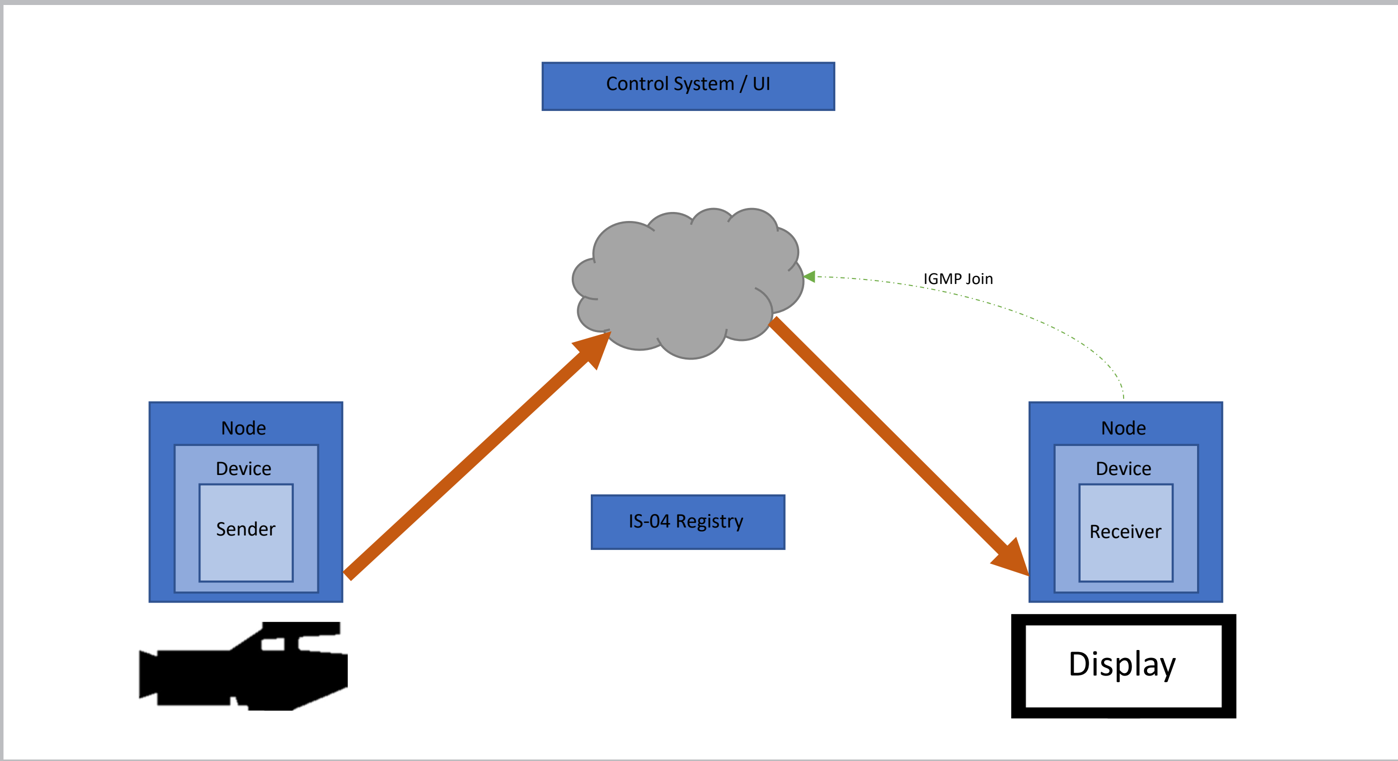


Figure 5 – Receiver issues an IGMP join after being configured and activated by an IS-05 controller.

“Plug-and-play” workflow with the help of IS-04 and IS-05

For connecting Senders and Receivers, IS-05 offers all of the needed parameters and allows the ability to alter them through a unified API, and then IS-04 identifies which Senders and Receivers exist. Therefore, if a user wants to use an IP-based control panel to connect Senders and Receivers, the following steps are implemented into a broadcast control system by IS-04 and IS-05:

1. Devices retrieve configuration port and media port IP addresses via DHCP.

2. IS-04 Node implementation discovers IS-04 registry in the network by mDNS or DNS entry.

3. IS-04 Node on devices registers NMOS Node for advertising the API via Config Port.

4. IS-04 implementation on devices registers all Senders and Receivers including information about IS-05 control.

5. The control system learns of all Senders and Receivers via the IS-04 Query API and can then automatically define multicast addresses for all Senders via IS-05 staged transport parameters.

6. A control panel can then be populated with all Senders and Receivers using the connected source and destination information of the Node API as label information.

7. Through user interaction via drag/drop or push of destination and source buttons, the control system retrieves the “transportfile” of the Sender representing the source and POSTs it to the desired Receiver.

This interaction can be implemented for all devices supporting IS-04 and IS-05 and offers interoperable connection management. The following figure visualizes the above interaction between a control system, two nNodes, an IS-04 Registry, and a DHCP Server.

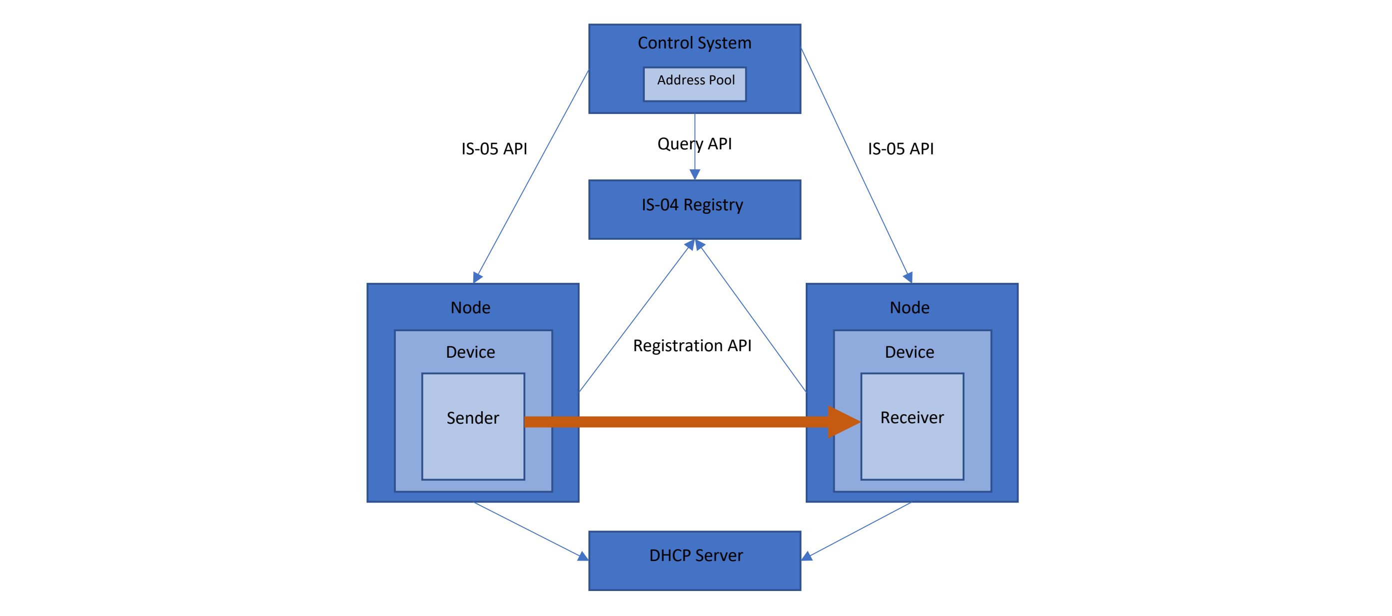


Figure 6 - IS-04 API relationships (including IS-05).

Security

AMWA's specifications define the APIs and the workflows to be used. The correct use of authentication, authorization, or accounting (AAA) mechanisms are currently outside the scope of IS-04 and IS-05. Nevertheless, recent activities within AMWA work on specifying additions to the currently used IP techniques of IS-04 and IS-05.

The use of TLS (HTTPS) allows obscuring the exchange of data between Senders, Receivers, the registry, and control systems. To enable TLS, all participating devices need to exchange keys beforehand to ensure proper decrypting and encrypting of messages.

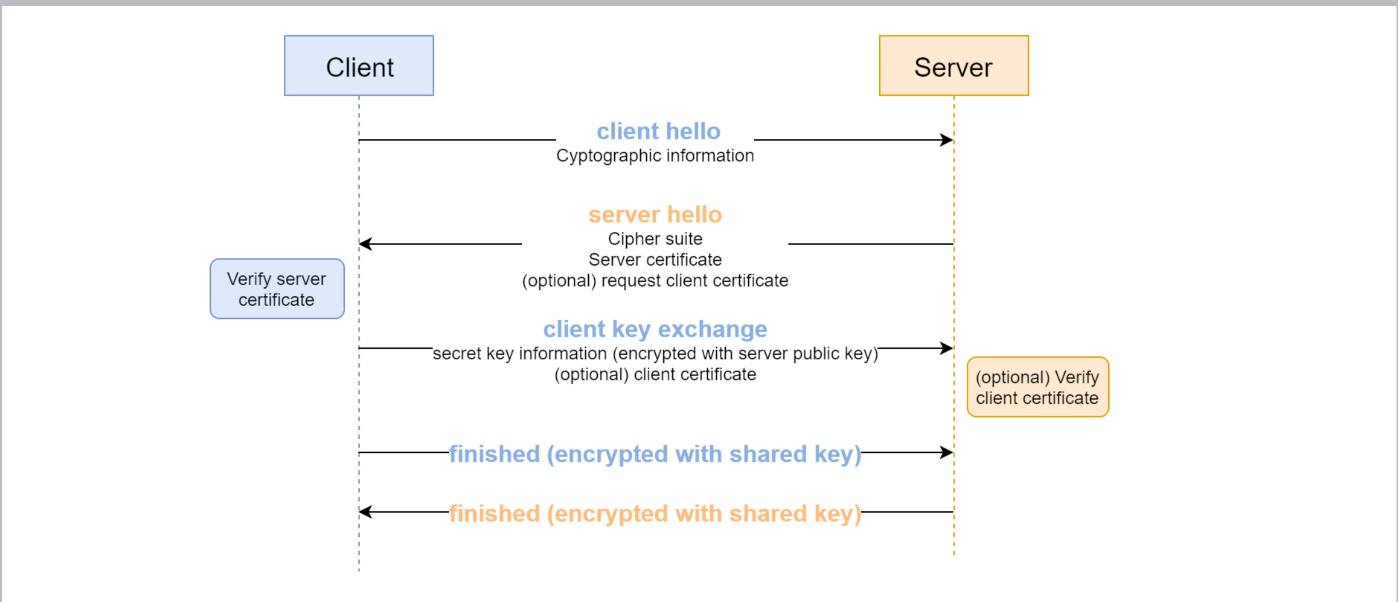


Figure 7 - Key exchange mechanism (simplified) to enable TLS traffic.

More information about HTTPS and key exchanging can be found in a recent whitepaper published by the BBC's R&D department that focuses specifically on the broadcast market and which methods are already being used in new AMWA activities. (https://www.bbc.co.uk/rd/publications/whitepaper337).

An authorization method is also needed as a second critical way of securing NMOS IS-05. By introducing an authorization server to the network that monitors all online Nodes and control systems, "permissions" can be issued to controllers based on user requirements. An IS-05 control system from a rogue laptop in the facility could then be prevented from starting or stopping streams.

Conclusion

In this article, we described the abstract architecture of a broadcast system for traditional baseband systems as well as IP-based systems. Clearly, additional means for identification and discovery are needed. AMWA IS-04 offers a representation of broadcast media devices and an open-source description of the API, along with recommended practices that are available and easy to implement. To use IP media essences in a dynamic fashion, the IS-05 API was introduced for connection management.

IS-05 offers all the necessary parameters to connect an AES67 and ST 2110-2, -30, or -40 Sender to a similar Receiver, both in unicast and multicast. While some additions would provide even more usability for the user, the current specification can serve as a de-facto standard for SMPTE and AES-based devices. Through application of the widely used HTTP protocol, IS-04 and IS-05 can easily be secured using common techniques.

Adding standard IT mechanisms like DHCP and DNS can offer "plug-and-play" workflows that do not require manually configuring IP addresses and would also make the transition to IPv6 much easier.

Sponsored by Riedel Communications

WP
WHITE PAPERS

EG
ESSENTIAL GUIDES

MEDIA

CS
CASE STUDIES

Find Out More

For more information and access to white papers, case studies and essential guides please visit:

thebroadcastbridge.com

8/2018

Sponsored by Riedel Communications

RIEDEL