Fundamentals of IP in Broadcast Production

Wes Simpson
Telecom Product Consulting
wes.simpson@gmail.com
+1 203 376 3372
https://www.telecompro.tv

Ed Calverley
Q3 Media Training
ed.calverley@q3media.co.uk
+44 20 3475 0250
https://q3mediatraining.co.uk

Agenda (90-minute Session)

- IP / Networking Basics
- Media Transport Over IP (ST 2110-10/20/30/40 Deep Dive)
- PTP: Timing & Synchronisation
- ST 2110-21 Traffic Shaping
- New parts of ST 2110 (ST 2110-22/23)
- ST 2022-7 Redundant Transport
- JT-NM TR-1001 / NMOS
Why IP?

• We must start by recapping the obvious!
• 2 main drivers for switching to IP Infrastructure:
  – Flexibility
    • Reconfigurable & infrastructure not limited by resolution/formats
    • Operational functions easier to relocate and evolve over time
    • More efficient architectures (don’t simply think about replacing SDI)
    • Software on generic IT servers rather than vendor-badged systems
    • may have to accept some compromises to change workflow
  – Costs > Use of more COTS hardware/software
    • minimise custom development and branched code makes systems easier to support
    • Up-front cost may not be lower (overall lifecycle costs may be lower)
    • Move to software-as-a-service (SaaS) – pay only for what you use when you need it!
COTS Confusion

COTS: Commercial Off-The-Shelf
- Defined Costs
- Standards based
- Choice (interoperable)
- Generic Features
- Stable
- Supported
- Upgradable

Proprietary / Bespoke System
- Variable Costs
- Vendor lock-in
- Interfacing/Conversion
- Custom Features
- Higher risk of issues
- Fixed function
- Limited/Costly to evolve

However:
- Not all IP hardware is equal
- Networking for broadcast media production is specialist
- Good system architecture & workflow planning is essential

The Rise of Software

- Dedicated Hardware still strong in production or wherever real-time processing with low-latencies are important
- Media processing with Software is growing in capability
- Software enables new architectures that don’t have equivalents in SDI
- Motivation for vendors is changing to create products which can be sold in scale
- IP Media Standards need to work for both software & hardware
**IP is all about Packets!**

**SDI Signal Routing**
- Dedicated wire per signal
- Routed via crosspoint switching

**IP Signal Routing**
- Multiple signals per wire
- Packet-level switching
- Full-Duplex (bi-directional)

**SDI** is a dedicated link with constant data rate
- SD-SDI: 270Mb/s
- HD-SDI: 1.5Gb/s
- 3G-SDI: 3Gb/s

**Ethernet links** can have higher data rate so more data can be carried in the same time
- 10GE: 10Gb/s
- 25GE: 25Gb/s
- 100GE = 4x25GE
- 40GE = 4x10GE

Data is sent in packets. Network Switches & Routers manage the sending of packets across the network.
How do we send data in packets?

- Prepare Data
- Choose Protocol
  - UDP
  - TCP
- Address it
- Send It

TCP Header

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Number</td>
<td>Acknowledgement Number</td>
</tr>
<tr>
<td>Data Offset</td>
<td>Window Size</td>
</tr>
<tr>
<td>Total Length</td>
<td>Urgent Pointer</td>
</tr>
<tr>
<td>Options</td>
<td>Padding</td>
</tr>
</tbody>
</table>

UDP Header

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Checksum</td>
</tr>
</tbody>
</table>

- Link Handshaking
- Transmission Acknowledgments
- Automatic resend on packet loss
- Perfect for FILES

- ‘Fire & Forget’
- Minimal Data Overhead
- Simple error detection
- Perfect for REAL-TIME STREAMS
How do we send data in packets?

• Prepare Data
• Chose Protocol
  • UDP
  • TCP
• Address it
• Send It

Application Layer
Transport Layer
Internet Layer

Port: 20000
IP Address: 192.168.0.1
How do we send data in packets?

1. Prepare Data
2. Choose Protocol
   - UDP
   - TCP
3. Address it
4. Send It

- Application Layer
- Transport Layer
- Internet Layer
- Link Layer

Sending SDI Over an IP Network

**Example:** SMPTE ST 2022-6

<table>
<thead>
<tr>
<th>Name</th>
<th>Standard</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDI</td>
<td>Serial Digital Interface</td>
<td>SMPTE 259M, 292M, 424M</td>
</tr>
<tr>
<td>HBRMT</td>
<td>High Bitrate Media Transport</td>
<td>SMPTE 2022-6</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-Time Transport Protocol</td>
<td>RFC 3550</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
<td>RFC 768</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol (v4/v6)</td>
<td>RFC 791 / RFC 2460</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control (e.g. Ethernet)</td>
<td>IEEE 802.3</td>
</tr>
</tbody>
</table>
UDP Multicast

Multicast allows a second device sending on the multicast group (same address/port)

239.10.0.25:20000

IP Camera

239.10.0.25:20000

Network Switch

IGMP Subscribe

IGMP v3 > Source Specific Multicast
IGMPv3 Introduced Source Specific Multicast (SSM) which ensures only multicast packets from a specific source address are received

Multicast address range to use SSM may be limited to 232.0.0.1/8

What are we sending?

SMPTE ST 2022-6

1 RTP flow

SMPTE ST 2110

2110-20 x 1

2110-30 x16 (e.g.)

2110-40 x 3 (e.g.)

~20 RTP flows
Media Transport over IP
SMPTE ST 2022-6 (‘SDI over IP’)

- Take entire SDI signal and encapsulate it in IP stream
  - Includes audio and embedded data signals
- Easy to maintain audio/video synchronization
  - Hard to process just one part of a stream
Media Transport over IP
SMPTE ST 2110

- Each media type in a separate packet stream
  - Easy to process individual components
  - Signals need to be resynchronized after processing
- PTP (Precision Time Protocol) used for packet timestamping

ST 2022-6 / ST 2110
Audio Processing Packet Flow

Using SDI/ST 2022-6

Using ST 2110
How Did We Get Here? – ST 2110 Evolution

Compressed

ST 2022-1,2

ST 2022-3,4

ST 2110-22

TR-01

Pro-MPEG CoP3

Uncompressed

ST 2110-10

ST 2110-20

ST 2110-30

ST 2110-40

ST 2110-10 System and Timing

ST 2110-20 Uncompressed Video

ST 2110-21 Video Stream Packet Shaping

ST 2110-30 Uncompressed Audio

ST 2110-31 AES3 Audio Streams

ST 2110-40 Ancillary Data
SMPE ST 2110 – Elements (NEW!)

- OV 2110-0 Roadmap for the 2110 Document Suite
- ST 2110-10 System and Timing
- ST 2110-20 Uncompressed Video
- ST 2110-21 Video Stream Packet Shaping
- ST 2110-22 Constant Bit-Rate Compressed Video
- RP 2110-23 Single Video Essence Transport over Multiple ST 2110-20 Streams
- ST 2110-30 Uncompressed Audio
- ST 2110-31 AES3 Audio Streams
- ST 2110-40 Ancillary Data

ST 2110-10
System Timing and Definitions

- Maximum UDP datagram size: 1460 octets, including UDP header
  - Extended UDP datagram allowed with up to 8960 octets
- SMPTE ST 2059-2 PTP Profile of IEEE 1588-2008
  - If interchanging audio with AES67, then compatible parameters must be used
- RTP timestamps are tied to the media
  - For video, RTP timestamps of all packets for video frame are the same
  - For real-time sources, this should represent the Image Capture Time
  - For SDI converters, RTP timestamp is moment when video frame alignment point arrives at device input (SMPTE ST2059-1 defines alignment points)
- All media clocks must have an offset of zero
  - This makes it easier to recover from loss of signal or unexpected system restart
ST 2110-20 Video Encapsulation

- Multiple video pixel groups (pgroups)
- RTP Payload Header applied
- Inserted into an RTP packet
- Placed into UDP packet
- IP packet header attached
- Wrapped into Ethernet Frame

• Same idea as before – just draw differently (and still not to scale!)
ST 2110-20 Pixel Groups

- Pixels formed into pgroups
  - pgroup size depends on sampling format
  - Must be integer number of octets
  - Pixels that share samples must be in the same pgroup

- Example: 4:2:2 10-bit
  - 2 pixels in 5 octets

Pixel Group Sizes

- Every supported video format listed in ST 2110-20 tables
  - Tables also include order of samples within each pgroup
ST 2110-20 Video Packet Header

- 32-bit Sequence Number (16 bit Sequence number would wrap in less than half a second for Gigabit-class payloads)
- Length of Sample Row Data = Number of octets from scan line in this datagram. Must be multiple of pgroup
- F = 0 for progressive scan and first field in interlace video
- F = 1 for second field in interlace video
- Video Line Number = Video scan line number, starts at 0 for first active line of video (note difference from SDI line numbering)
- C = 1 if more than one line is in datagram, set to 0 for last line in each datagram
- Sample Row Data Offset = Location of first pixel of payload data within scan line = 0 if first pixel in scan line; counts by pixels

ST 2110-20 Sample Row Data

- Packet 0
  - Sample Row 0
  - Offset 0
- Packet 1
  - Sample Row 0
  - Offset X
- Packet 2
  - Sample Row 0,1
  - Offset 2X, 0

Embedded Audio (HANC)

Ancillary Data (VANC)
ST 2110-30 Audio Encapsulation

- Multiple Audio Samples (16 or 24 bit)
- Grouped into one RTP packet
- Placed into UDP packet
- IP packet header attached
- Wrapped into Ethernet Frame

ST 2110-30 Audio

- Based on AES67
  - 48 kHz, 24-bit linear encoding must be supported in all devices
- Zero Offset Media Clock
  - Forces all media clocks to be tied to common time base
- Audio Channel Grouping
  - How audio channels relate to each other in a stream
- Receiver Classifications
  - Three levels of receiver performance
- Packet size limit \(1440 = 1460 - (12 \text{ (RTP)} + 8 \text{ (UDP)})\)
- No need for SIP or other connection management
Importance of “ptime”

- Audio streams are divided into fixed duration packets
  - Common size is 1 msec, signaled using “a=ptime:1” attribute
- Number of samples from a channel depends on sampling rate
  - For example, 48 kHz has 48 samples in 1 msec
  - Each sample could be 2 bytes (16 bit audio) or 3 bytes (24 bit audio)
  - Thus, 1 msec of 48 kHz, 24-bit audio is 48 * 3 = 144 bytes
- Number of channels in a packet limited by payload size
  - Total RTP audio payload is 1440 bytes
  - Jumbo frames not allowed for audio

ST 2110-30 Receiver Classifications

<table>
<thead>
<tr>
<th>Required Sampling Rates and Packet Times</th>
<th>A</th>
<th>AX</th>
<th>B</th>
<th>BX</th>
<th>C</th>
<th>CX</th>
</tr>
</thead>
<tbody>
<tr>
<td>48 KHz, 1 msec</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>48 KHz, 125 μsec</td>
<td></td>
<td></td>
<td>8</td>
<td>8</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>96 KHz, 1 msec</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96 KHz, 125 μsec</td>
<td></td>
<td>8</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ST 2110-30 Audio
Channel Grouping Symbols

<table>
<thead>
<tr>
<th>Channel Grouping Symbol</th>
<th>Quantity of Audio Channels in group</th>
<th>Description of group</th>
<th>Order of Audio Channels in group</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>1</td>
<td>Mono</td>
<td>Mono</td>
</tr>
<tr>
<td>DM</td>
<td>2</td>
<td>Dual Mono</td>
<td>M1, M2</td>
</tr>
<tr>
<td>ST</td>
<td>2</td>
<td>Standard Stereo</td>
<td>Left, Right</td>
</tr>
<tr>
<td>LfRt</td>
<td>2</td>
<td>Matrix Stereo</td>
<td>Left Total, Right Total</td>
</tr>
<tr>
<td>51</td>
<td>6</td>
<td>5.1 Surround</td>
<td>L, R, C, LFE, Ls, Rs</td>
</tr>
<tr>
<td>71</td>
<td>8</td>
<td>7.1 Surround</td>
<td>L, R, C, LFE, Lss, Lrs, Lrs</td>
</tr>
<tr>
<td>222</td>
<td>24</td>
<td>22.2 Surround</td>
<td>Per SMPTE ST 2036-2, Table 1</td>
</tr>
<tr>
<td>U01...U64</td>
<td>Undefined</td>
<td>Unn where nn is the number of channels in group</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

ST 2110-31 AES3 Audio Streams

• AES3 streams can be used for non-PCM audio applications
  – Dolby E Compressed Audio is one common application
  – Other signals defined in SMPTE ST 337/338 (e.g. AC-3 compressed audio)
  – Has also been used for non-linear audio, one-bit audio and SACD
  – Also know as AES/EBU Audio

• For these applications, transparent carriage across IP is a must
  – Cannot change any bits within the stream
  – Data cannot be interpreted as uncompressed linear audio signals

• ST 2110-31 should NOT be used for linear 16-bit or 24-bit audio
  – Those should be carried using ST 2110-30 packet streams
ST 2110-31 Packet Format

- Based on AM824 data format
  - 32 bit sub-frames, each with 8 bits of signaling and 24 bits of data
  - Can hold all data plus signaling bits from AES-3 (B, F, P, C, U, V)

<table>
<thead>
<tr>
<th>V</th>
<th>P</th>
<th>X</th>
<th>CC</th>
<th>M</th>
<th>PT</th>
<th>RTP Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time Stamp (32 bits)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Synchronization Source (SSRC) Identifier</td>
</tr>
<tr>
<td>B,F,P,C,U,V data bits</td>
<td>AM824 sub-frame 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B,F,P,C,U,V data bits</td>
<td>AM824 sub-frame 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B,F,P,C,U,V data bits</td>
<td>AM824 sub-frame 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ST 2110-40 Ancillary Data

- Extract ancillary data packets from VANC or HANC
  - Captions, time code, ad triggers, etc.
  - Place them into RTP packets with custom header
- Line numbers are based on SDI line numbering
  - Don’t match 2110-20 line numbers
ST 2110-40 ANC Packet Format

- Each ANC packet in the RTP payload has its own header
- Color channel flag: C=1 – ANC packet is from HD color difference channel. C=0 in all other cases
- Line Number and Horizontal Offset refer to SDI raster values
- S=1 Multiple streams comprise the format of the original video signal containing the ANC packets
- Stream number indicates where the ANC packets were located within a multi-stream signal
- DID, SDID, Data Count, Packet Payload and Checksum are exact 10-bit values from ANC packet
- For each ANC packet within the RTP payload, padding makes the total number of bits a multiple of 32

<table>
<thead>
<tr>
<th>C</th>
<th>Line Number (11 bits)</th>
<th>Horizontal Offset (12 bits)</th>
<th>S</th>
<th>Stream Num (7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DID (10 bits)</td>
<td>SDID (10 bits)</td>
<td>Data Count (10 bits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANC Packet Payload</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANC Packet Payload</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Padding to 32 bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SDP – Session Description Protocol

- Standardized format for describing video and audio content
  - RFC 4566
- Provides key data needed to process content
  - Structural metadata for each type of media stream
  - Connection information for each stream
  - Clock and timing information
  - Stream associations for closely coupled streams (ST 2022-7 hitless and RP 2110-23 subdivided streams)
- Text file
  - Can be accessed through NMOS APIs, other means
Parts of a Session Description

Session Information

Timing Information

Media Information

Connection Information

SDP Example

v=0
o=wes 203763372 89 IN IP4 10.201.33.19
s=Example of a 1080i29.97 video signal
t=0 0
m=video 31008 RTP/AVP 101
c=IN IP4 239.201.33.11/32
a=source-filter: incl IN IP4 239.201.33.11 10.201.33.19
a=rtpmap:101 raw/90000
a=fmtp:101 sampling=YCbCr-4:2:2; width=1920; height=1080; interlace; exactframerate=30000/1001; depth=10; TCS=SDR; colorimetry=BT709;
PM=2110GPM; SSN=ST2110-20:2017;
a=mediaclk:direct=0
**Time / Synchronisation**

- Synchronous signals are essential for production to allow clean-switching/mixing and capture to a common timebase

Date <> Time-of-day <> Frequency <> Phase

- **SMPTE-309M**
- **SMPTE-12M**
- **B&B/Tri-Level Sync**

- ‘Offline’
- ‘Real-time’
  
  (genlock)
Time / Synchronisation

- **PTPv2 (IEEE 1588-2008)**
  - Defines mechanism for accurately setting a local clock via exchange of a few simple messages
  - PTP timestamps are 80-bits in size: 48 bits Seconds : 32 bits nanoseconds
  - Can handle all our needs from Date/Time Timecode through to Frequency (Genlock)
  - PTP Profiles defined by **AES67 / SMPTE 2059-2**
    -(see AES-R-16-2016 for compatibility recommendations)

- **Epoc 1970-01-01 00:00:00**

  - **NTP** – accuracy ~200 micro seconds
  - **PTP** – accuracy ~1 micro second

  ![Image of PTP Architecture](image-url)

  *PTP ensures all slaves have the same time regardless of their distance (signal transit time) from the master*

---

We learned from Y2K! 48bits is approx. 9 million years!
1-Step Sync with End-to-End Delay

- **Sync Message:**
  - “The time now is XX:XX” (sec:nanosec.)
  - If slave immediately updated clock to be the time in the sync message (T1), it would still be offset equal to the time taken for the message to transit the link

- **Delay Request & Response:**
  - Measures time taken to transit network link
  - Assumes symmetrical delay

PTP Implementation Options

**Time-Stamping Mechanism**

- **1-Step**
  - Accurate timestamp written into packet at the point of egress
  - Requires precision hardware implementation

- **2-step**
  - Message formed and passed to hardware for egress
  - Actual time of egress reported and inserted in a follow-up message

- **Messages affected:**
  - Sync Message
  - Peer Delay Response Message *(will cover later!)*
2-Step Sync with End-to-End Delay

- **Sync Message:**
  - “The time now is XX:XX” (sec:nanosec.)
  - If slave immediately updated clock to be the time in the sync message (T1), it would still be offset equal to the time taken for the message to transit the link
  - Actual time of egress sent in a follow-up message

- **Delay Request & Response:**
  - Measures time taken to transit network link
  - Assumes symmetrical delay

PTP Architecture

- **Unicast Mode**
- **Multicast Mode**
- **Mixed Mode (Hybrid)**
PTP Architecture

- Unicast Mode
- Mixed Mode (Hybrid)  
  \textit{Sync/Follow-up Multicast}  
  \textit{Delay Req/Resp Unicast}
- Multicast Mode

All messages Multicast  
Delay Req/Resp ignored by other slaves
Passive Network Switches

- Any network switch which is not ‘PTP-Aware’
- Variable delay can be introduced as messages transit
- QoS can be used to improve performance
- The assumption of symmetry used in the End-2-End delay mechanism can result in clock jitter
- Passive switches can be used but their impact needs to be considered
IEE 1598-2008 defines the following device Types

- **Ordinary Clock**
  - Device with a single PTP port, can be grandmaster-capable or can be slave-only

- **Boundary Clock**
  - has multiple PTP ports, synchronises network segments

- **End-to-End Transparent Clock**
  - has multiple PTP ports, modify timestamps in PTP messages
  - residence times measured/added to the correctionField of Sync / Delay_Req (one-step) or Follow_Up / Delay_Resp (two-step)

- **Peer-to-Peer Transparent Clock**
  - has multiple PTP ports, modify timestamps in PTP messages
  - residence times plus egress-link delays measured/added to the correctionField of Sync (one-step) or Follow_Up (two-step) messages

- **Management Node**

---

### Boundary Clock Network Switch

![Boundary Clock Network Switch Diagram](image-url)
Boundary Clock Network Switch

- One port on the network switch will be a PTP Slave
- All other ports become PTP Master
- Provides independent links to downstream clocks
- Allows expansion through multiple tiers and with common grandmaster
- Individual links use End-to-End delay mechanism

Transparent Clock Network Switch

2 functional modes:
- End-2-End
- Peer-2-Peer

Peer-2-Peer delay mechanism also requires slaves to be appropriately configured
PTP Implementation Options

Delay Mechanism

• **End-to-End**
  – Slave measures the delay between itself and the master
  – Messages: Delay_Req / Delay_Resp

• **Peer-to-Peer**
  – Each network element measures the delay between its port and the device on the other end of the link
  – Measured delays for each network element added to SYNC message as it transits between master and slave
  – Messages: Pdelay_Req / Pdelay_Resp (+Follow-up for 2-step)

Transparent Clock (TC) Network Switch (E2E)

• PTP messages can be modified on the way through the switch

• ‘**Residence Time**’ (the time spent in the switch) is measured and added to the correction field in PTP message:
  – 1 Step: Sync / Delay_Req
  – 2-Step: Follow_up / Delay Resp

• Each TC **adds** its measured value to correction field value (i.e. tracks the cumulative delay)
Transparent Clock (TC) Network Switch (P2P)

- Peer-2-Peer Delay Mechanism
- Sends/responds to Peer Delay Request messages on each port to assess delay on local link
- Residence Time + Peer Delay on Ingress Link added to Correction Field

Real Networks
Electing a Grandmaster

- Best Master Clock Algorithm (BMCA) is used to elect a Grandmaster
- attributes are evaluated in the following order:
  - Priority 1 (lowest number wins)
  - Clock Class (GPS, free-run, etc)
  - Clock Accuracy (accuracy to UTC)
  - Clock Variance (jitter and wander)
  - Priority 2 (lowest number wins)
  - GMID (similar to mac address)
- These attributes are advertised in the PTP Announce Message (typically 1 - 4 per second depending on config)
- NB: all other equipment should be set to be PTP Slave only to prevent accidental election of inappropriate masters
Switching Packets Requires Buffering
Wireshark captures showing some 2110-20 flows from different hardware devices

Software processes typically work with frame buffers so this could get a lot worse!

2110-21 Traffic Shaping
Networking Detail
ST 2110-21 Timing Models

- Senders can’t burst out all of their data at once
  - Overloads receivers and network switch buffers
- Some variability is necessary
  - HANC/VANC gaps, software-based senders

Two Constraints for ST 2110-20 Senders

- Network Compatibility Model
  - Ensures streams will not overflow buffers inside network devices
  - Scaling factor $\beta$ of 1.1 means buffers drain 10% faster than they fill
- Virtual Receiver Buffer Model
  - Buffer is modelled as input of every receiver device
    - Note: Must be included in end-to-end system delay
  - Packets read from buffer perfectly, based on video format
  - Buffer not allowed to overflow or underflow
- All senders must comply with both models
ST 2110-21 Gapped, Linear Packet Schedules

- **Type N (Narrow)**
  - designed for real-time capture and processing
  - Maximum required receiver buffer is about 9 packets in gapped mode
  - Model assumes $T_{OFFSET}$ of a couple of video lines from SMPTE Epoch
  - Small buffer means limited delay passing through each device in systems
  - Pixels inside packets “roughly” in sync with pixels in SDI

- **Type NL (Narrow Linear)**
  - linear version of $N$
  - no gaps corresponding to SDI VANC

- **Type W (Wide)**
  - designed to support software-based video sources
  - Maximum receive buffer is 720 packets in some popular formats
  - Larger buffer can handle packet bursts more easily
  - Bursty transmission is more common to software-based senders
ST 2110-21
Sender/Receiver Compatibility

<table>
<thead>
<tr>
<th>Receiver Type</th>
<th>Type N Sender</th>
<th>Type NL Sender</th>
<th>Type W Sender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type N Synchronous Narrow</td>
<td>Mandatory</td>
<td>Optional</td>
<td>No</td>
</tr>
<tr>
<td>Type W Synchronous Wide</td>
<td>Mandatory</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Type A Asynchronous</td>
<td>Mandatory</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

- Synchronous Receivers must have clock locked to Sender
- Synchronous Narrow Receivers are only required to work with Senders that use the default TROFFSET

New parts of 2110
Forthcoming: SMPTE ST 2110-22

- Current Title:
  “Professional Media over Managed IP Networks:
- Constant Bit-Rate Compressed Video”
  – Supports CBR compression formats such as VC2
  – Must be a registered RTP media type as per RFC 4855
  – RTP Clock rate of 90 kHz
  – Must conform to either “NL” or “W” network compatibility model of ST 2110-21; virtual receiver buffer model does not apply

Forthcoming: SMPTE ST 2110-22

- Visually lossless compression cannot be seen by observer
  – Some data must always be removed
  – Done so as to be invisible to human viewer
  – Can have very low latency – using slice-based compression

- Popular codecs available
  – VC-2 DIRAC from BBC – RFC 8450
  – Also JPEG XS – draft-lugan-payload-rtp-jpegxs-01

- 2:1 to 8:1 compression ratios
  – 3Gbit/s SDI compressed to 1.5 to 0.5 Gbit/s
Forthcoming: SMPTE ST 2110-22
SDP for ST 2110-22

- Format parameters (a=fmt) statement must include
  - Image height in lines
  - Image width in pixels
  - TP of either 2110TPNL or 2110TPW
  - Optional value of CMAX if different from default

- Bit rate parameter “b=AS:<bandwidth>“ must be included
  - Bandwidth is in kilobits/second calculated over one frame period

- SDP must include a frame rate statement, either
  - a=framerate xx.yy (as a decimal number)
  - exactframerate=M/N (as a ratio of two integers) in “fmtp”

Forthcoming: SMPTE RP 2110-23
Recommended Practice

- Working Title:
  “Single Video Essence Transport over Multiple ST 2110-20 Streams”

- Idea is to have a system where multiple low-bandwidth streams can be used to transport one high-bandwidth signal
  - High resolution streams, such as UHD1/4K or UHD2/8K
  - High frame rate streams, such as those over 100 fps
  - Also known as “multiport”

- Each sub-stream is a valid ST 2110-20/2110-21 stream
  - Timestamps tied to original frames
  - Comply with timing models
Three Methods to Split Stream

1. Temporal De-Interleave
2. Square Division
3. Two-Sample Interleave

2022-7 Redundant Transport
2022-7 Hitless Protection Switching

- Send identical signal on two separate paths
  - Identical packet timestamps, sequence numbers
  - Receiver aligns packets using buffer
  - Transit time equal to delay of longest path

- SMPTE 2022-7 Standard
  - “Seamless Protection Switching of SMPTE ST 2022 IP datagrams”
  - Published 2013

Route Diversity using ST 2022-7

- Diverse routes with SMPTE ST 2022-7 Hitless Protection Switching
Standards & Bodies (Related to 2110)

Industry Bodies
- EBU: European Broadcasting Union
- AMWA: Advanced Media Workflow Association
- VSF: Video Services Forum
- IABM

Industry Bodies with working groups defining specifications
- EBU
- AMWA
- VSF
- IABM

Standards
- EBU
- AMWA
- SMPTE
- AES: Audio Engineering Society

Standards
- General
- IETF: Internet Engineering Task Force
- IEEE: Institute of Electrical and Electronics Engineers

Copyright 2019 - No part of this presentation may be reproduced or repurposed without permission of the authors: telecompro.tv / Q3 Media Ltd.
JT-NM TR-1001-1:2018 v1.0

“System Environment and Device Behaviours for ST 2110 Media Nodes in Engineered Networks – Networks, Registration, and Connection Management”

- The goal of this document is to enable the creation of network environments where an end-user can take delivery of new equipment, connect it to their network, and configure it for use, with a minimum amount of human interaction.

EBU: “The Technology Pyramid for Media Nodes”

The minimum stack of endpoint technologies to build an IP-based media facility

- **MEDIA TRANSPORT**
  - Single link video (ST 2110-20)
  - Software-friendly video receivers (ST 2110-21)
  - Multichannel, low-latency audio (ST 2110-30)
  - Stream protection (ST 2022-7)

- **DISCOVERY & CONNECTION**
  - Discovery and registration (AMWA IS-04)
  - Connection management (AMWA IS-05)
  - Audio mapping (AMWA IS-08)
  - Topology discovery (LLDP)

- **TIME & SYNC**
  - PTPv2 configurable within SMPTE/AES profiles
  - Multi-interface PTP redundancy
  - Synchronisation of A/V and data essences

- **CONFIG & MONITORING**
  - IP Assignment (DHCP)
  - Open configuration management (e.g. API, config file, SSH CLI etc.)
  - Open monitoring protocol (e.g. syslog, agent, SNMPv3 etc.)

- **SECURITY**
  - Security tests (EBU R 148)
  - Security safeguards (EBU R 143)
  - Secure HTTPS API calls


NMOS

• A set of specifications/protocols created by AMWA’s Network Media Incubator working group
• Developed since 2015 alongside standardisation of SMPTE ST-2110 (which evolved out of VSF’s TR-03 format)
• Modelled around concepts outlined in JT-NM’s Reference Architecture document published at IBC 2015 (http://jt-nm.org/RA-1.0/)

https://amwa-tu.github.io/nmos/

IS-04 Discovery & Registration

• Keeping track of all the flows, what is generating them, and what can consume them
• Protocols defined for both Peer-to-Peer discovery and discovery via central Registry
• mDNS discovery (within subnet)
• HTTP-based protocols for Node/Registration/Query API’s (JSON payload)
• Flow properties/parameters via SDP file (RFC 4566)
### IS-04 Discovery & Registration

- **API’s defined by IS-04**
  - Node
  - Registration
  - Query
- **All devices should be PTP locked**

### IS-05 Discovery & Registration

- **IS-05 works alongside IS-04 to provide mechanism to configure senders & receivers**
- Commands can be ‘staged’ before making ‘active’ *(e.g. to allow validation)*
- Support for applying immediately or with time offset
- **Single / Bulk mechanism** *(e.g. Salvo routing)*
IS-06 Network Control

- IS-06 is an API for interfacing to an SDN controller (e.g. when using software defined network routing instead of standard IGMP-based multicast routing)

IS-07 Event & Tally

- Mechanism to emit and consume states and state changes issued by sources
- ‘GPI for IP’ – General purpose interfacing for buttons/knobs/sliders/displays/lights
- Extensible to carry additional data/values (potentially much more than triggers & tallies!)
- Clients subscribe to message queues (MQTT / Websockets)
- Small message size with very low-latency
- Multiple option timestamps for tracking action/effect times
- Signalling device reboot/shutdown (in addition to general purpose state messages)
## Network Media Open Specifications

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Spec Status</th>
<th>Version(s)</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS-04</td>
<td>Discovery &amp; Registration</td>
<td>AMWA Specification (Stable)</td>
<td>v1.3 / v1.2.2 / v1.1.3</td>
<td>nmos-discovery-registration</td>
</tr>
<tr>
<td>IS-05</td>
<td>Device Connection Management</td>
<td>AMWA Specification (Stable)</td>
<td>v1.1 / v1.0.2</td>
<td>nmos-device-connection-management</td>
</tr>
<tr>
<td>IS-06</td>
<td>Network Control</td>
<td>AMWA Specification</td>
<td>v1.0</td>
<td>nmos-network-control</td>
</tr>
<tr>
<td>IS-07</td>
<td>Event &amp; Tally</td>
<td>AMWA Specification</td>
<td>v1.0.1</td>
<td>nmos-event-tally</td>
</tr>
<tr>
<td>IS-08</td>
<td>Audio Channel Mapping</td>
<td>AMWA Specification</td>
<td>v1.0</td>
<td>nmos-audio-channel-mapping</td>
</tr>
<tr>
<td>IS-09</td>
<td>System</td>
<td>Work In Progress</td>
<td></td>
<td>nmos-system</td>
</tr>
<tr>
<td>IS-10</td>
<td>Authorization</td>
<td>Work In Progress</td>
<td></td>
<td>nmos-authorization</td>
</tr>
<tr>
<td>MS-04</td>
<td>ID &amp; Timing Model</td>
<td>Work In Progress</td>
<td></td>
<td>nmos-id-timing-model</td>
</tr>
<tr>
<td>BCP-002-01</td>
<td>Natural Grouping</td>
<td>AMWA Specification</td>
<td></td>
<td>nmos-grouping</td>
</tr>
<tr>
<td>BCP-003-01</td>
<td>API Security: Communications</td>
<td>AMWA Specification</td>
<td></td>
<td>nmos-api-security</td>
</tr>
<tr>
<td>BCP-003-02</td>
<td>API Security: Authorization</td>
<td>Work In Progress</td>
<td></td>
<td>nmos-api-security</td>
</tr>
<tr>
<td></td>
<td>Parameter Registers</td>
<td>Continuing</td>
<td></td>
<td>nmos-parameter-registers</td>
</tr>
</tbody>
</table>

---

**JT-NM TR-1001 / NMOS at IBC 2019**

Successfully met test criteria at interop event held prior to NAB 2019

Successfully met test criteria at interop event held prior to IBC 2019

Manufacturers who have implemented NMOS ('Self-certified')

Full list of both tests and successful participants here: [http://jt-nm.org/jt-nm_tested/](http://jt-nm.org/jt-nm_tested/)

---

Copyright 2019 - No part of this presentation may be reproduced or repurposed without permission of the authors: telecompro.tv / Q3 Media Ltd.
Thank you

Wes Simpson
Telecom Product Consulting
wes.simpson@gmail.com
+1 203 376 3372
https://www.telecompro.tv

Ed Calverley
Q3 Media Training
ed.calverley@q3media.co.uk
+44 20 3475 0250
https://q3mediatraining.co.uk

Thank you to our Media Partners

Thank you