NMOS IS-04

Discovery and Registration

SMPTE Event hosted by Atos
25 October 2018

Chris Gil & Cristian Recoseanu
Introduction

Why do we need NMOS?

AMWA NMOS Specifications
Discovery and Registration
Connection Management
Event & Tally

Demonstrations
IP Before NMOS

Broadcast Infrastructure already moving to IP - first audio followed by video

New installations can use COTS network switches instead of video and audio routers

Sources are offered as multicast streams - destinations subscribe to streams

Without a router port table how can the sources and destinations be identified?

How can connections be managed?

(There are no “crosspoints”)
Pre-NMOS Solutions: Router Emulation
Pre-NMOS Solutions: Router Emulation

Vendor specific IP Routing Orchestration can expose a logical router on a Northbound interface using a traditional router protocol

- Sender devices are assigned a source index
- Receiver devices are assigned a destination index
- The Control System sends router crosspoint commands to the router interface
- Router Emulator sends tally notifications to the Control System

Example solution:
BNCS controls Bloomberg IP Routing via GV using SW-P-08 protocol
Pre-NMOS Solutions: Router Emulation issues

1. Vendor specific IP Routing Orchestration must support control of each receiver endpoint device
   - including 3rd party devices that may be supplied by competitors

2. New endpoints that are discovered need to be assigned a new index
   - applies to both sources and destinations

3. The control system has no visibility of which endpoints are “offline”
   - a NACK response to a crosspoint request is a basic fallback
Pre-NMOS Solutions: Direct Control

Control System

Management Network

Protocol A

Device Driver A

Manufacturer A

Control Interface

Device A1

Device A2

Sender

Receiver

Protocol B

Device Driver B

Manufacturer B

Control Interface

Device B1

Sender

Receiver

Protocol C

Device Driver C

Manufacturer C

Control Interface

Device C1

Device C2

Sender

Receiver

Media Network

Manufacturer D

Device D1

Sender

Receiver
Pre-NMOS Solutions: Direct Control

The Control System is already managing parameters of many devices

Additional functions for the Control System:

- Set parameters on stream receivers to join a multicast group & port
- Match the multicast group & port to a sender
- Derive a “crosspoint” tally to report a receiver has subscribed to a sender
  A.K.A. a destination has a source routed to it

Example solution:
BNCS controls Sky Studios MCR IP Routing using native protocols on devices
Pre-NMOS Solutions: Direct Control issues

1. Control system must support control of each receiver endpoint device
   - drivers for each new product may not be immediately available and
   proprietary protocol may never be disclosed

2. Direct parameter control of a device via public API may not apply changes to
   subscription parameters immediately
   - immediate crosspoints may not be achievable

3. Responsiveness of devices to parameter changes can be variable and
   asynchronous change notifications are not always available

4. A multicast profile may be used by the wrong sender
Pre-NMOS Solutions: Hybrid Control
Pre-NMOS Solutions: Hybrid Control

In principle a Control System could use both methods for routing:

- Route to endpoints that have no public interface via Router Emulation
- Route to endpoints that do have a 3rd party control interface by directly setting parameters

Issues:

- All issues mentioned before for Router Emulation and Direct Control still apply
- Additional issue - Synchronisation of sender profiles
**NMOS Overview**

- Network Media Open Specification
- Interoperability between devices from different manufacturers
- Simplified Integration - one implementation works with all other vendors
- Key features:
  - Dynamic Central Registry for all equipment
  - Content Identity can be traced
  - Unified Connection Management
  - Entirely Open with the aim of achieving maximum interoperability
NMOS – a family of specifications

- IS-04 – Discovery and Registration Specification
- IS-05 – Device Connection Management Specification
- IS-06 – Network Control Specification
- IS-07 – Event & Tally Specification
NMOS IS-04 Discovery and Registration

- Central Registry
- Resources: Nodes, Devices, Sources, Flows, Senders & Receivers
- Identity: GUID for every resource

```
"id": "bfdf3b2b-7abf-21e8-8db8-40a36ba01e76"
```
NMOS IS-04 Interfaces

- Registration Interface for Nodes
- Query Interface for Controllers
NMOS IS-04 Registry Browser
IS-04 Main Features

- Establishes core entities and their registration/discovery mechanics
- Defines the relationships between entities
- Includes the concept of identity and content identity
- Has a mechanism for tracking when entities have been modified
- Provides asynchronous subscriptions for interested clients to keep up to date and in sync with the state of the system
- Acts as a core building block for subsequent specifications in the NMOS suite
- Heartbeats are used to handle nodes being lost from the system
- Timing – timestamps are used to synchronise streams
NMOS IS-05 Connection Management

- Send Connection parameters to Receiver Device via IS-05
- Notification via IS-04 websocket
NMOS IS-05 Connection Management

- A common API for connecting IP transports
- A control system can send “route” instructions to a device
- Depends on IS-04 model
NMOS IS-05 Connection Management

- Obtain Transport Parameters from Sender Device
  - RTP: multicast group, port

- Send Transport Parameters to Receiver Device

- Notification from Receiver via IS-04 subscription Websocket – “tally”

- Immediate activation vs Staged activation

- Bulk API for multiple level connections on a Receiver Device
  - 2110 Video + Audio(s) + Data
IS-05 Main Features

- Establishes a unified API for connecting various supported transports
- Establishes the means by which transport parameters can be modified on both the sending and receiving sides
- Establishes the means by which parameter changes can be staged and activated either immediately or scheduled for a later time
- Offers a bulk API for allowing multiple connections to be issued within the context of the same receiver device.
- Changes in receiver connections are reported via IS-04 Websocket subscriptions
NMOS IS-04 & IS-05 Demonstration
NMOS IS-06 Network Control Specification

- Abstracts the network from the broadcast controller and offers a unified agnostic API
- Establishes the means by which the network topology can be discovered including links between switches as well as links to devices
- Establishes the means by which media flows can be created from a sender to one or more receivers
- Establishes the means by which bandwidth can be protected for defined media flows
IS-06 Current Architecture
NMOS IS-07 Event & Tally Specification

- [Work in progress]
- Targets GPI and Serial wiring in Broadcast Facility
- Full integration with IS-04 and IS-05
- Transport options: Websocket & MQTT
- Data Types: bool, string, number, enumeration (in phase 1)
- Connections: 1-1 or 1-Many
- **NOT** a control API (no commands are sent to specific receivers)
NMOS IS-07 – Control System

- Connects publishers to subscribers
- Once connected the Control System does not need to be in the loop
- If necessary the Control System can also subscribe to monitor state of senders
- Compatibility of Event Types
IS-07 Main Features

- Establishes a mechanism by which to emit and consume state changes issued by sources
- Establishes the means by which to determine compatibilities between a source and a receiver using event types and metadata descriptions
- Establishes guidelines and mechanisms for late joining receivers to get in sync with an emitter state
- Establishes a mechanism for highlighting interruptions in the events channel (disconnections/reboots/shutdowns)
### IS-07 Event Types

<table>
<thead>
<tr>
<th>Boolean</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>● True [input present]</td>
<td>● UMD label</td>
</tr>
<tr>
<td>● False [input missing]</td>
<td>● Floor Manager name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number</th>
<th>Enumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Temperature</td>
<td>● Ok, Warn, Fail</td>
</tr>
<tr>
<td>● Error Count</td>
<td>● 625i50, 720p50, 1080i50, 1080p50</td>
</tr>
</tbody>
</table>
IS-07 Event Types - Compatibility

- Sources will always advertise a single *event type*

- Receivers advertise capabilities as an array of *event types* and can also use a wildcard *
<table>
<thead>
<tr>
<th>Boolean</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>● boolean</td>
<td>● string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number</th>
<th>Enumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>● number/temperature/C</td>
<td></td>
</tr>
<tr>
<td>● number/temperature/*</td>
<td>● number/enum/StudioCondition</td>
</tr>
<tr>
<td>● number/temperature/*</td>
<td>● number/enum/*</td>
</tr>
</tbody>
</table>
IS-07 Demonstration
IS-07 Interoperability Demo with Pebble Beach

- BNCS publishes IS-07 events when Hardware Panel buttons are pressed

- Pebble Dolphin subscribes to events and changes internal resources:
  - Video Source Selection
  - Graphic Elements On/Off
  - Logo, Clock, DVE

- Pebble Dolphin Publishes IS-07 state change events

- BNCS updates Hardware Panel LCD labels and backlight colour
IS-07 REST API

- Retrieve associated metadata information
  
  Example Event Type: number/enum/StudioCondition

  Example Payload:
  
  ```json
  {
    "values": [
      {
        "value": 0,
        "label": "idle",
        "description": "Studio condition is idle"
      },
      {
        "value": 1,
        "label": "reh",
        "description": "Studio condition is rehearsal"
      },
      {
        "value": 2,
        "label": "tx",
        "description": "Studio condition is tx"
      }
    ]
  }
  ```

- Mechanism for late joiners need to synchronise with the last state of a sender
IS-07 potential uses

- GPIs
- Camera tally lights
- On-Air and other studio status lights
- Multiviewer UMD labels and tally lights
- Vision mixer tally states
- Hardware control surfaces
- Triggers of graphics overlays
- Device signal and packet telemetry status
- Bridge to integrate with building management systems
  - Air conditioning, House lights and Blinds
IS-07 Facility Infrastructure

- Unlike a GPI wiring schematic nothing is connected by default
- No hard wiring or jumpers
- Expected connections need to be made explicitly
- Questions:
  - How do you define the expected connections?
  - Are they static or dynamic?
  - Is a salvo or scenario recall needed?
For “Normal” infrastructure behaviour all Event connections must be established and maintained.
Packages and Grouping

- Natural grouping using Grouping API – an extension of IS-04
- Bundling - packages
  - Commissioning - apply tags and labels
  - Association - build packages
  - Routing - simple operations with correct video/audio/data

- Package Levels
  - Video
  - Audio
  - ANC Data
  - Event & Tally
Packages Demonstration

Routing Packages

Package Tally

Match compatible Source and Destination Packages

Indication of missing levels
Where next?

● What users should do to increase adoption?
● Participate in Workshops – develop your own implementations
● Host labs
● Visit the web resources:
  ○ [https://nmos.tv/](https://nmos.tv/)  Introduction to NMOS and Presentations
  ○ [https://amwa-tv.github.io/nmos/](https://amwa-tv.github.io/nmos/) Specifications and Documentation
  ○ [https://github.com/AMWA-TV](https://github.com/AMWA-TV) AMWA public repositories

● This presentation was given at the SMPTE event hosted by Atos on 25 October 2018 - more information here: