discovering | designing | delivering |

control

The practical challenges of navigating from SDI to IP

A Pebble White Paper by Cristian Recoseanu - Development Team Lead

While the move towards IP production is gathering pace rapidly, there are still a few areas where the industry as a whole is not as advanced as it should be. For those looking to set up IP-native workflows, there are a few gotchas still lurking in the details.

pebble

Gone are the days of plugging in some SDI cables and immediately seeing the picture from your camera on the monitor. IP flows between a sender and a receiver need to be first initiated and then maintained and monitored at all times. Meanwhile, endpoints need to be configured and provisioned with IP addresses, multicast addresses, and port numbers. And other considerations need to be addressed too, such as security.

In an SDI-based system, security effectively started and stopped with locking the door to the apparatus room. With an IP-based system, security is paramount as the attack surface to hacking becomes much larger. That means that authentication and identity checking, as well as action authorization, need to be enforced.

In other words, while there are many noted benefits to the wholesale move to IP, it also comes along with the unwelcome spectre of increased complexity. So, how do we navigate from one state to the other?

We will illustrate the issues involved with the following simple system:

- **4 cameras with a connected microphone each**
- **1 large monitor display**
- **1 multiviewer device with 32 inputs and 4 outputs**
- **multiviewer output 1 has a predefined quad 4 x tile layout, where tiles 1-4 are connected internally in sequence to inputs 1-4**

The goal is to connect all of the 4 cameras to the multiviewer inputs and be able to visualise multiviewer output 1 on a large display.

For simplification, we are going to assume in both SDI and IP cases that the SDI router matrix and the network fabric have been installed and configured. We are also going to assume in the IP analysis that the NMOS suite of specifications is being used.

One of the first steps is to install the 4 cameras and cable them into sources 1-4 on the SDI router. We then proceed to do a similar thing for the multiviewer and connect its 4 outputs into sources 5-8 on the SDI router and the 32 inputs in destinations 1-32. Finally, the monitor display input can be connected as destination 33.

With the cable plugging out of the way, connections need to be issued. This can be achieved through a front panel if the SDI router has one or through a dedicated broadcast controller. In the latter case, the broadcast controller would need to interface with the SDI router controller using a specific protocol (sometimes proprietary). The end-user would then use the broadcast controller interface (software or hardware) to make the following connections:

- **Sources 1-4 (cameras) to Destinations 1-4 (first 4 multiviewer inputs)**
- **Source 5 (multiviewer output 1) to Destination 33 (monitor display input)**

At this point, we should see an image on the display monitor which contains the multiviewer output 1 in a quad layout depicting 4 tiles, each one showing cameras 1-4.

1. SDI workflow

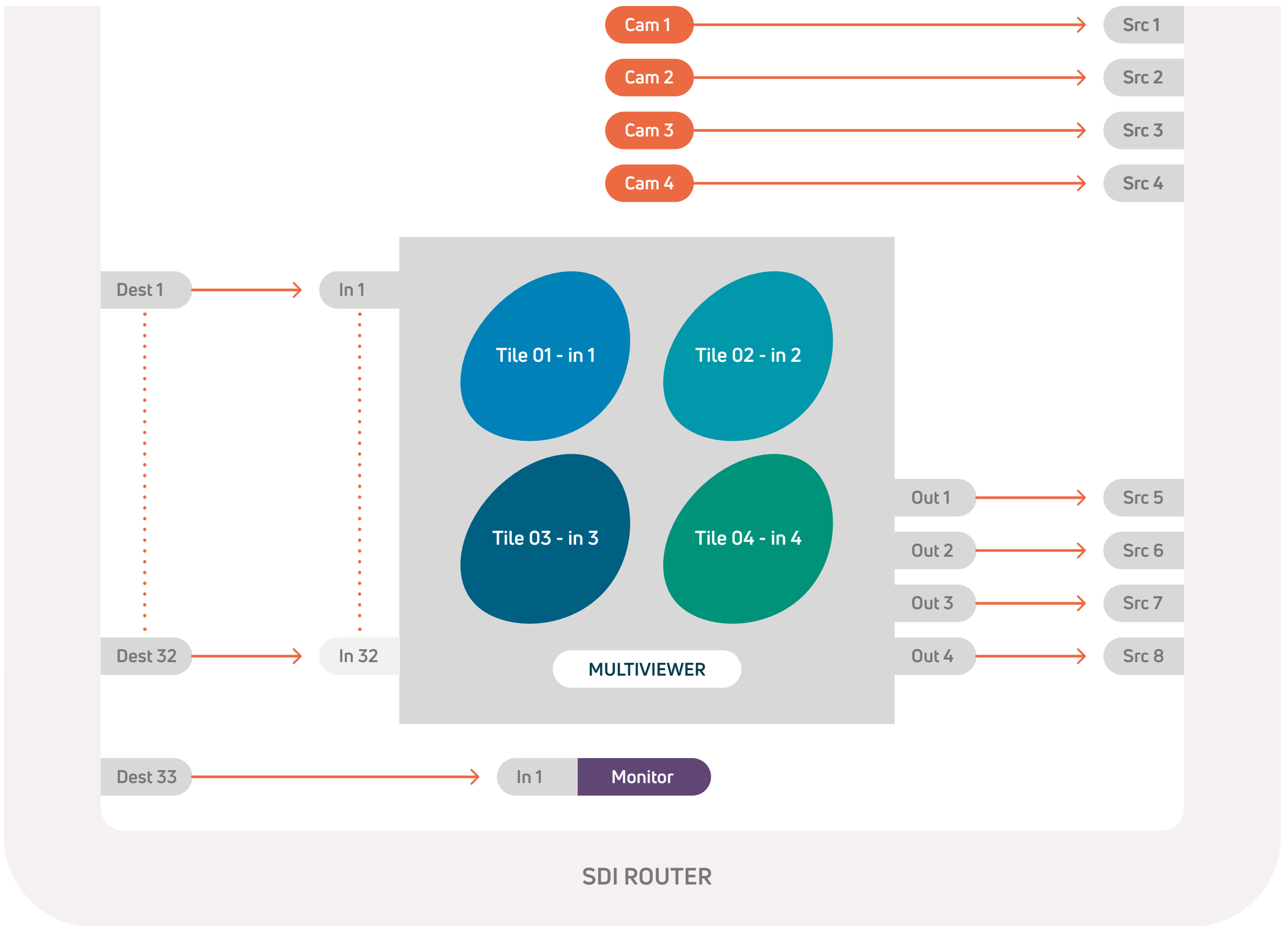

Figure 1 – SDI connections

2. The IP/2110/NMOS workflow

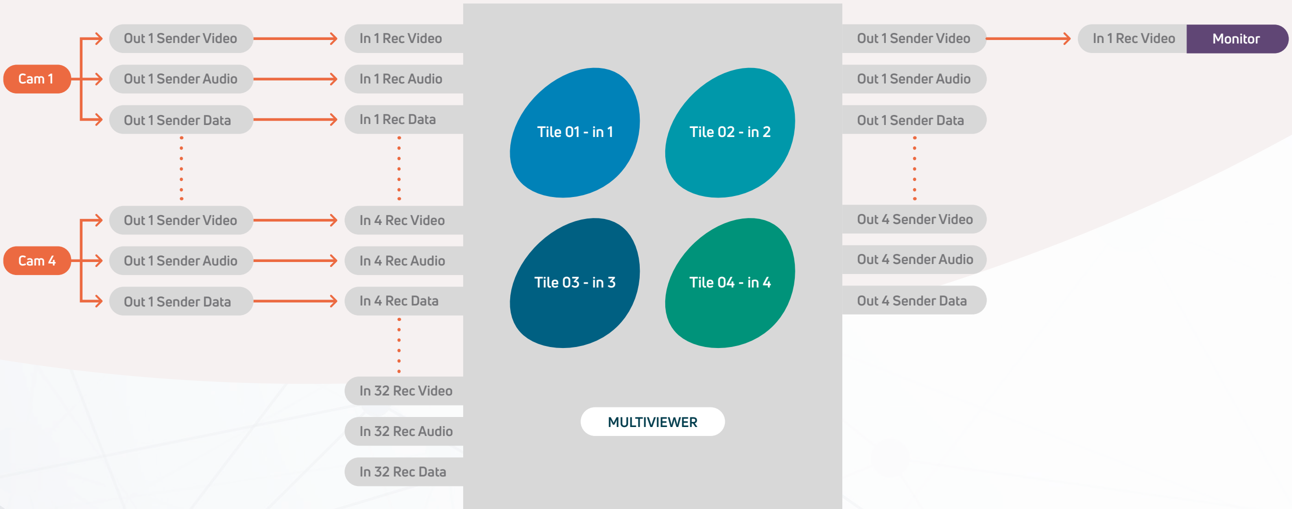

Figure 2 – Required IP connections

In this case, the first step would be to provision a DNS server and an NMOS registry service that has an appropriate DNS-SD record associated with it. This allows other NMOS nodes to discover the registry and register themselves (following the mechanism described in IS-04).

At this point, we have complete visibility of all of the nodes and their senders and receivers in the NMOS registry. To have the ability to connect them, however, all of the senders need to be configured. They need to be enabled and their stream parameters need to be provisioned (source address, multicast address, port number, etc.). This is a mandatory step that is best done through a control system's configuration/management UI.

Upon successful configuration of the senders, connections may be issued for compatible entities. Ideally, a control system would present the user with an intuitive interface that can highlight compatible senders and receivers. After a connect action is sent from the UI, the control system needs to use the IS-05 API to retrieve an SDP file for each sender (a description file that contains detailed parameters including flow parameters like source address, multicast address, and port number). This is then used in a subsequent API call against a receiver to tune it into the stream emitted by the desired sender. If receivers are on the same device, the control system may choose to issue the connect commands as a single bulk command.

After all the connection requests have been sent, all of the receivers will change their subscription and update their entries in the NMOS registry to reflect this. A well-integrated control system will make use of the IS-04 registry subscription mechanism to get WebSocket notifications when resources change. This means it can track all of the changes happening to the system dynamically and report what sender each receiver is connected to.

And at this point, as with the SDI system, we should see an image on the display monitor which contains the multiviewer output 1 in a quad layout depicting 4 tiles, each one showing cameras 1-4.

Five challenges

The scenario above highlights the increased importance of a competent control system in IP workflows. There are five key takeaways from our analysis of the two different methodologies:

- **Senders need to be provisioned**
- **Resource in the NMOS registry has to be made visible in a UI that can highlight compatible items**
- **Video, Audio and Data flows need to be connected for each virtual input/output and their relationship needs to be maintained without fault (we don't want the audio from camera 2 mixed with the video from camera 1). Feature-rich control systems can simplify this by allowing the creation of logical constructs which can group related resources**
- **Resource changes need to be tracked and reported dynamically (a receiver connected to a new sender or a device dropping off from the registry)**
- **As opposed to SDI, IP connections need to be actively re-asserted and monitored as there is no default signal path through the network**

Deeper future systems integration

Moving forward, there are clear desires for an infrastructure as a service world where platforms are cloud fit. This can only be achieved by not only embracing modern technologies of the web but also building from the ground up using interoperable architectures like NMOS.

Platforms of the future will make these valuable traits a part of their DNA as they strive to increase productivity, reduce risk and reduce costs. With that vision in mind, it is worth examining the following integration example. Here a powerful suite of interoperable NMOS specifications is at play (IS-04, IS-05, and IS-07), implemented by different systems in the facility (control system, building management system, and digital signage, as well as all the NMOS compatible media nodes).

pebble

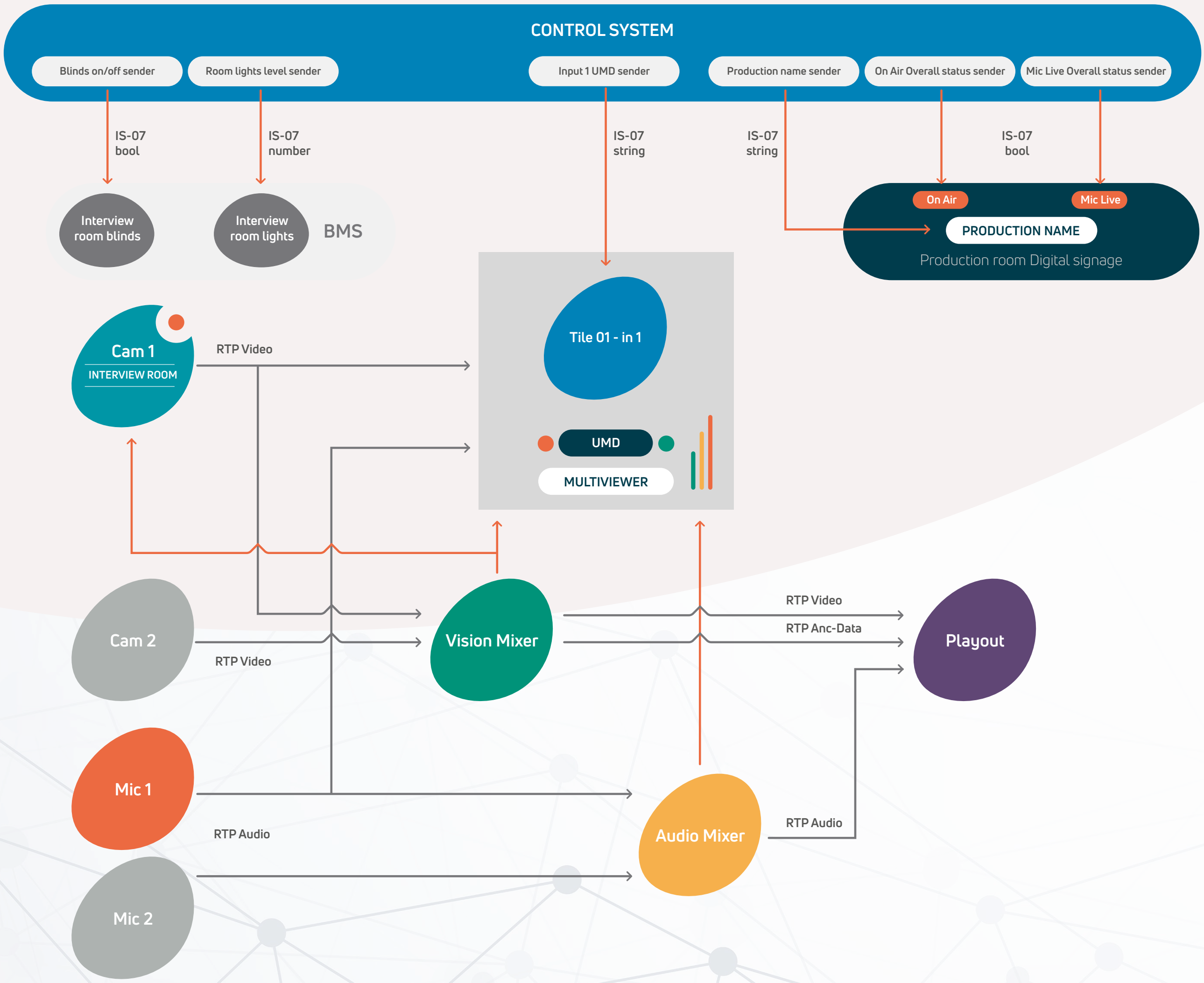

Fig 3. System diagram

The example showcases how detailed workflows can be engineered with multiple parts all being interconnected and architected by the control system. Connections are issued using IS-05 and all changes are tracked dynamically using IS-04 mechanics. IS-07 allows for raw or processed states to be sent out to different interested parties to increase the visibility of certain elements of the production process.

On-air tally lamps can be switched on, UMD labels can be set and interconnected building signage systems can be updated to reflect the current production taking place.

Furthermore, because everything is based on technologies that underpin the internet, there is no reason why some elements of the control can't be orchestrated from an MCR which is separated geographically from the production room. Remote productions are becoming more widespread and with that also the need for better integrations and control. MCR may be able to lower the blinds down in the interview room on a particularly sunny day or adjust the level of lights at night time.

Moving towards the future

The migration to IP-based facilities has already begun. The immediate and long-term benefits are clear with vast opportunities for deeper systems integration, improved scalability, developing infrastructure as a service, empowering remote productions, and finally dematerialising facilities and separating the productions from the data centre.

All of these opportunities have to be encapsulated by platforms of the future which are built on interoperable and web-friendly technologies, with security as a core and not an afterthought (indeed the NMOS community is currently working on a suite of security specifications which aim to provide just this core capability).

Control systems need to reduce complexity, reduce risk, increase productivity, while at the same time offering seamless integrations to other facility platforms. All of these have to be backed by the commitment of companies, their development teams, and their technology strategists to shape the next generation of products.

pebble

discovering | designing | delivering |

pebble

Unit 1, First Quarter, Blenheim Road
Epsom, KT19 9QN United Kingdom
Tel: +44 1932 333 790 | Email: info@pebble.tv

www.pebble.tv