



# AMWA NMOS IS-04 and IS-05 Scalability and Performance

Rob Porter  
Sony Europe Limited

**SONY**

IP SHOWCASE THEATRE AT IBC – SEPT. 14-18, 2018



## AMWA NMOS IS-04 and IS-05 Scalability and Performance

### Part 1

#### Overview of AMWA IS-04 and IS-05

- APIs
- Open Source Software

### Part 2

#### AMWA NMOS Scalability Study

- Methodology
- Results

### Part 3

#### Discussion

- Best Practice Recommendations
- Future Work

**SONY**

2



# AMWA NMOS IS-04 and IS-05 Scalability and Performance

## Part 1

### Overview of AMWA IS-04 and IS-05

- APIs
- Open Source Software

## Part 2

### AMWA NMOS Scalability Study

- Methodology
- Results

## Part 3

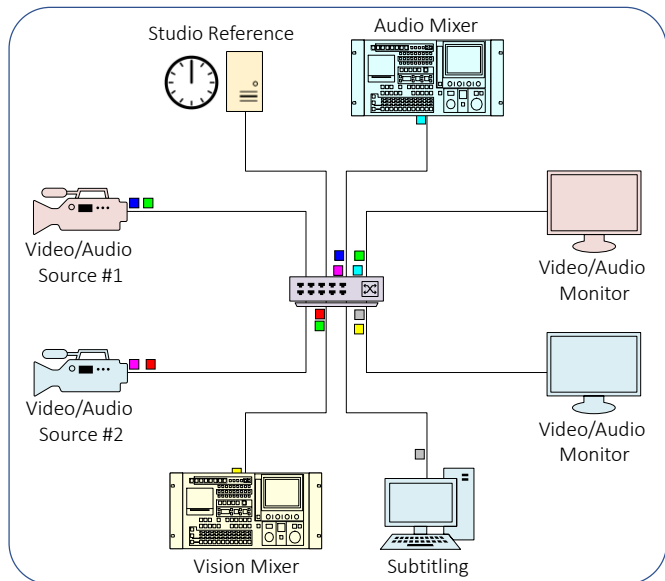
### Discussion

- Best Practice Recommendations
- Future Work



## Professional Media IP Network

- Video, audio and ancillary data carried as packetized data over IP network
- Timing provided by PTP clock
- What about managing the connections between devices?
- This is where the Networked Media Open Specifications (NMOS) come in...





### AMWA Networked Media Open Specifications

- **AMWA IS-04** for Discovery and Registration
- **AMWA IS-05** for Connection Management
- These are open specifications to allow interoperability between different manufacturers' devices
- IS-04 and IS-05 are RESTful APIs using HTTP POST, PUT, PATCH and GET as well as WebSockets for notifications of changes



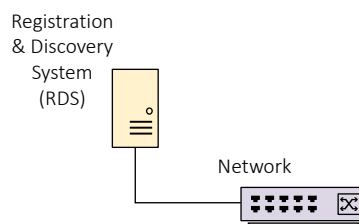
SONY

5



### AMWA IS-04 for Discovery and Registration

- Introduce an IS-04 **Registration & Discovery System (RDS)** into the network
- RDS comprises one or more Registry instances – these include a database storing all registered resources
- RDS exposes two APIs: a **Registration API** and a **Query API**
- NMOS **Nodes** register with RDS using its Registration API



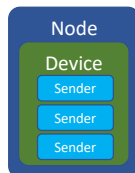
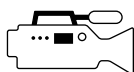
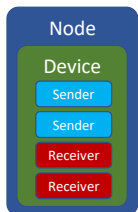
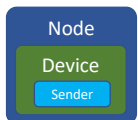
SONY

6

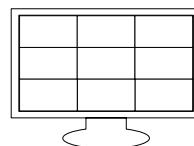


### NMOS Node

- An NMOS **Node** is a logical host connected to the network
- Can host one or more **Devices** each with any number of associated **Senders, Receivers, Sources and Flows**
- These are known as *resources* and are all defined by the JT-NM content model
- Some examples shown here, but note that there is more than one valid way to represent some of these items



SDI to ST 2110 Gateway



IP Multiviewer

SONY



### AMWA IS-04 for Discovery and Registration

- On connecting a Node to the network:

Registration & Discovery System (RDS)



Network

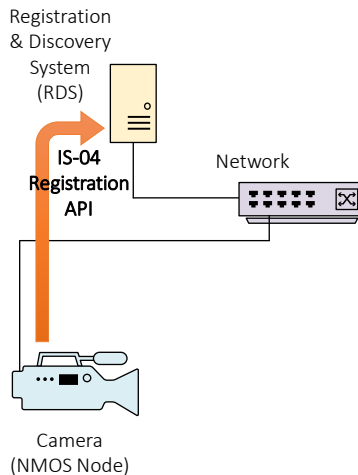


SONY



### AMWA IS-04 for Discovery and Registration

- On connecting a Node to the network:
  - Node discovers Registration APIs advertised over DNS-SD
  - Node selects a Registration API (by highest priority)
  - Node registers its Node resource with selected Registration API
  - Node registers each of its sub-resources (Devices, Senders, Receivers, Sources, Flows) and begins to post regular heartbeats

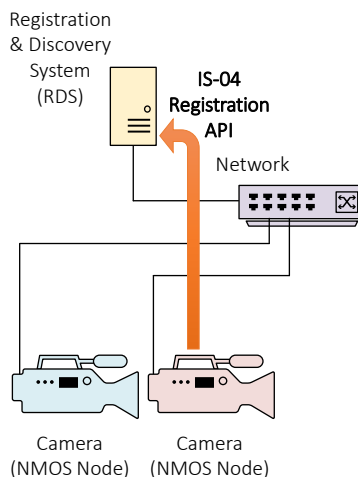


SONY



### AMWA IS-04 for Discovery and Registration

- On connecting a Node to the network:
  - Node discovers Registration APIs advertised over DNS-SD
  - Node selects a Registration API (by highest priority)
  - Node registers its Node resource with selected Registration API
  - Node registers each of its sub-resources (Devices, Senders, Receivers, Sources, Flows) and begins to post regular heartbeats

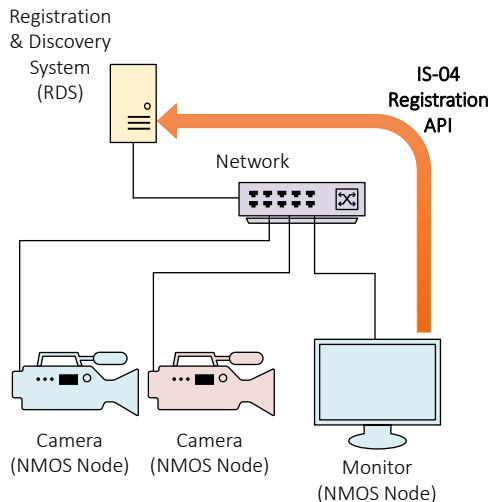


SONY



### AMWA IS-04 for Discovery and Registration

- On connecting a Node to the network:
  - Node discovers Registration APIs advertised over DNS-SD
  - Node selects a Registration API (by highest priority)
  - Node registers its Node resource with selected Registration API
  - Node registers each of its sub-resources (Devices, Senders, Receivers, Sources, Flows) and begins to post regular heartbeats

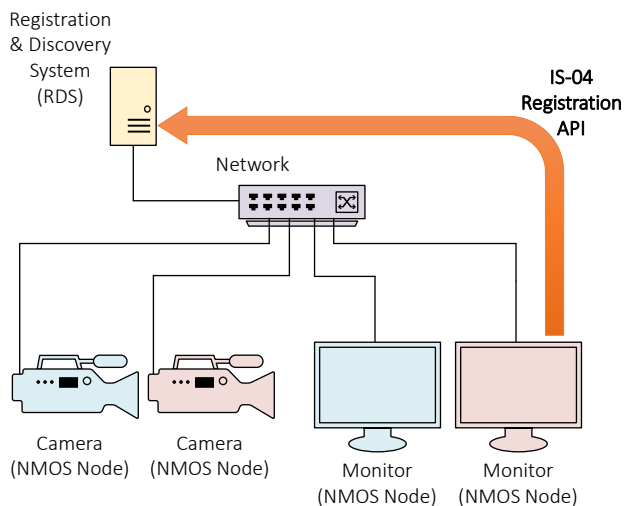


SONY




### AMWA IS-04 for Discovery and Registration

- On connecting a Node to the network:
  - Node discovers Registration APIs advertised over DNS-SD
  - Node selects a Registration API (by highest priority)
  - Node registers its Node resource with selected Registration API
  - Node registers each of its sub-resources (Devices, Senders, Receivers, Sources, Flows) and begins to post regular heartbeats



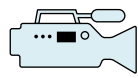
SONY



**AMWA IS-04 for Discovery and Registration**

- What is going over the wire?
  - Registration of an NMOS Node


NMOS Node



**Registration API Request (Node→RDS):**  
 POST /x-nmos/registration/v1.2/resource HTTP/1.1  
 Content-Type: application/json

```


{
  "type": "node",
  "data": {
    "version": "1441973902:879053935",
    "hostname": "host1",
    "label": "host1",
    "description": "host1",
    "tags": {},
    "href": "http://172.29.80.65:12345/",
    "api": {
      "versions": ["v1.1", "v1.2"],
      "endpoints": [
        {
          "host": "172.29.80.65",
          "port": 12345,
          "protocol": "http"
        },
        {
          "host": "172.29.80.65",
          "port": 443,
          "protocol": "https"
        }
      ]
    }
  }
}
        
```



RDS

**SONY**

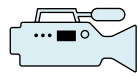
13



**AMWA IS-04 for Discovery and Registration**

- What is going over the wire?
  - Registration of an NMOS Node

NMOS Node



**Registration API Request (Node→RDS):**  
 POST /x-nmos/registration/v1.2/resource HTTP/1.1  
 Content-Type: application/json


```

{
  "type": "node",
  "data": {
    "version": "1441973902:879053935",
    "hostname": "host1",
    "label": "host1",
    "description": "host1",
    "tags": {},
    "href": "http://172.29.80.65:12345/",
    "api": {
      "versions": ["v1.1", "v1.2"],
      "endpoints": [
        {
          "host": "172.29.80.65",
          "port": 12345,
          "protocol": "http"
        },
        {
          "host": "172.29.80.65",
          "port": 443,
          "protocol": "https"
        }
      ]
    }
  }
}
        
```

**Registration API Response (RDS→Node):**  
 HTTP/1.1 201 Created  
 /x-nmos/registration/v1.2/resource/nodes/3b8be755-08ff-452b-b217-c9151eb21193/

```

{
  "port": 443,
  "protocol": "https"
}
        
```



RDS

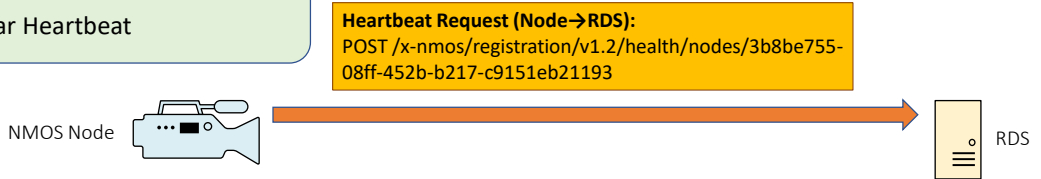
**SONY**

14



### AMWA IS-04 for Discovery and Registration

- What is going over the wire?
  - Regular Heartbeat



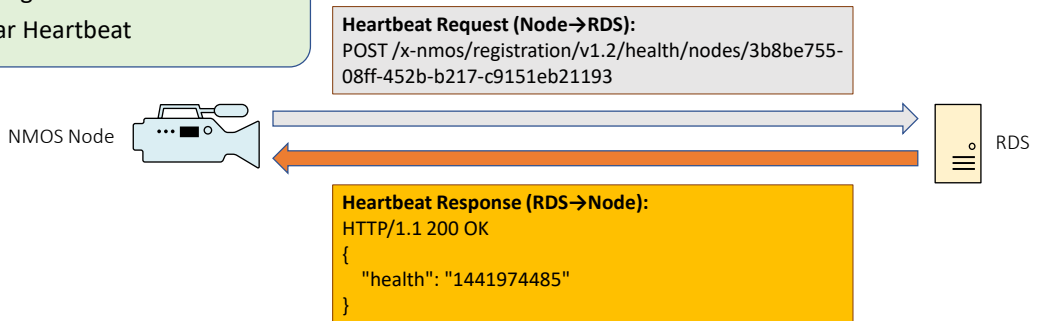
SONY

15



### AMWA IS-04 for Discovery and Registration

- What is going over the wire?
  - Regular Heartbeat



SONY

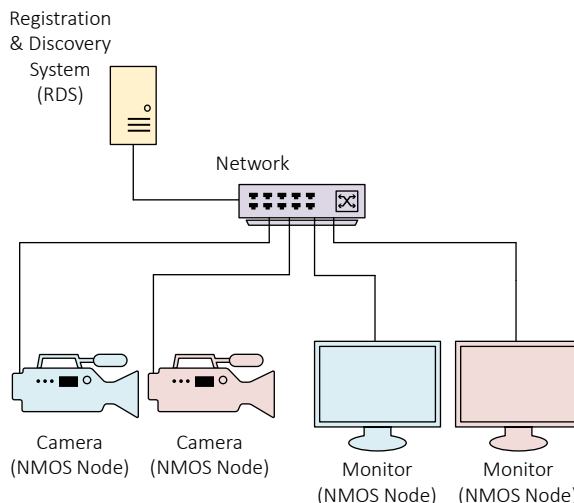
16





### AMWA IS-04 for Discovery and Registration

- Introduce an NMOS Client, for example a Broadcast Controller, to the network



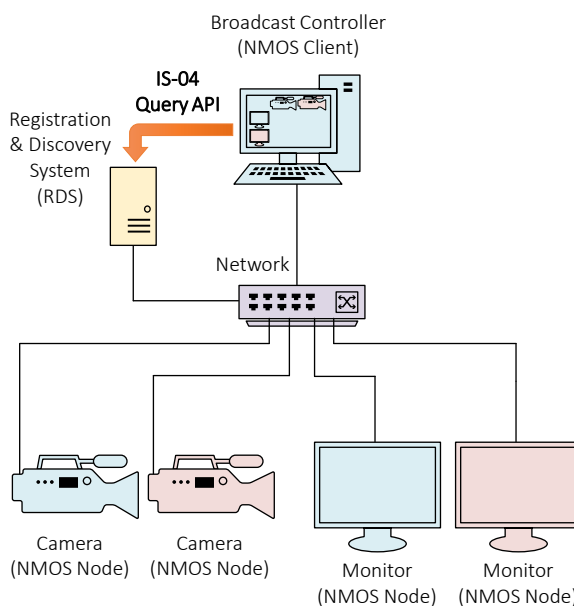
SONY

17



### AMWA IS-04 for Discovery and Registration

- Introduce an NMOS Client, for example a Broadcast Controller, to the network
- NMOS Client is able to get list of registered resources from RDS using IS-04 Query API
- It can also subscribe to WebSockets notifications of changes in the RDS
- What about making connections between Nodes?



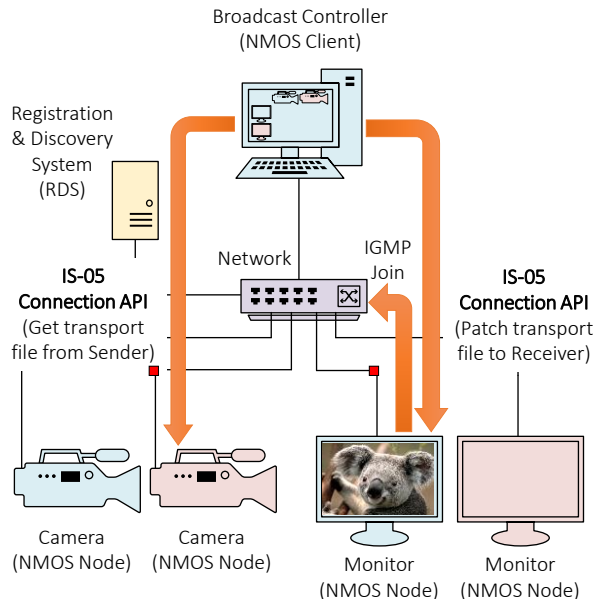
SONY

18



### AMWA IS-05 for Connection Management

- Broadcast Controller (NMOS Client) can use a Node's IS-05 Connection API to make a connection
  - Get transport file from Sender
  - Patch transport file to Receiver
  - Join multicast group
- IS-05 features include:
  - Two-stage connection
  - Delayed activation
  - Bulk mode connection



SONY

19



### Open Source Software for AMWA IS-04 and IS-05

- Various NMOS implementations are available in different languages
- All available under Apache 2.0 licence

BBC | Research & Development

Streampunk Media

SONY

Creator	Language	Licence	URL	Description
BBC R&D	Python	Apache 2.0	<a href="https://github.com/bbc/nmos-joint-ri">https://github.com/bbc/nmos-joint-ri</a>	IS-04 Registry and IS-04/-05 Node
Streampunk Media	Javascript (NodeJS)	Apache 2.0	<a href="https://github.com/Streampunk/ledger">https://github.com/Streampunk/ledger</a>	IS-04 v1.0 Registry and Node
Sony	Javascript (AngularJS)	Apache 2.0	<a href="https://github.com/sony/nmos-js">https://github.com/sony/nmos-js</a>	IS-04 and IS-05 Client
Sony	C++	Apache 2.0	<a href="https://github.com/sony/nmos-cpp">https://github.com/sony/nmos-cpp</a>	IS-04 Registry and IS-04/-05 Node

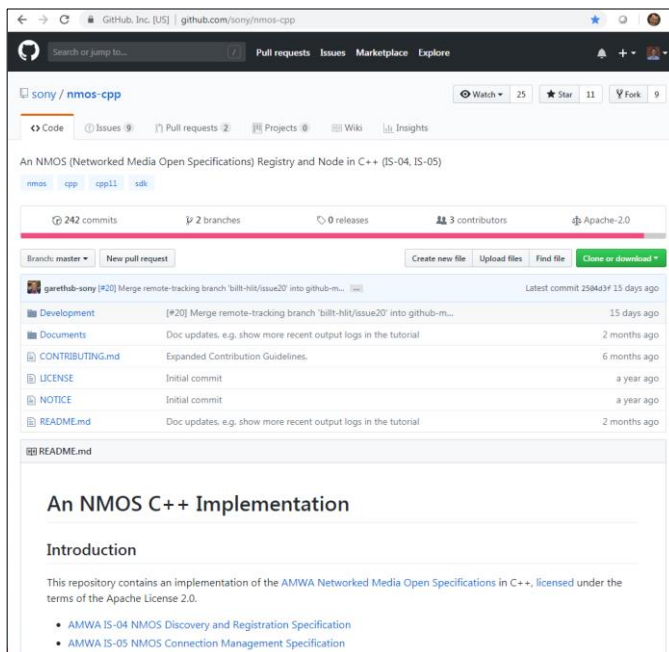
SONY

20



**Sony's Open Source Software:  
nmos-cpp**

- Implementation of the NMOS IS-04 and IS-05 specifications in C++
- Includes an **NMOS Registry** implementation and an **NMOS Node** implementation
- Actively being developed and supported
- Several contributors already involved – no need to sign a Contributor Licence Agreement (CLA)
- <https://github.com/sony/nmos-cpp>



**SONY**

21



**AMWA NMOS IS-04 and IS-05  
Scalability and Performance**

**Part 1**

**Overview of AMWA IS-04 and IS-05**

- APIs
- Open Source Software

**Part 2**

**AMWA NMOS Scalability Study**

- Methodology
- Results

**Part 3**

**Discussion**

- Best Practice Recommendations
- Future Work

**SONY**

22

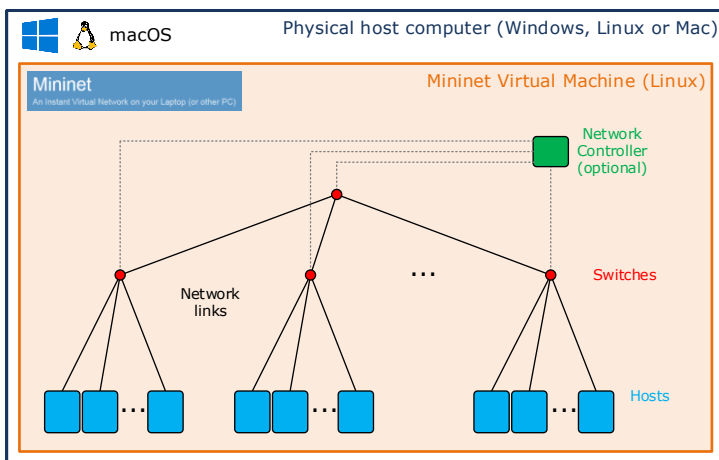


### The AMWA NMOS Scalability Study

- A key requirement of the AMWA IS-04 and IS-05 APIs is that they can be used reliably at scale
  - i.e. for very large networks comprising thousands of NMOS Nodes such as might be found in a typical broadcast installation.
- AMWA NMOS Scalability Study aims to help address this
- Study taking place within the AMWA community and led by Sony
- The study makes use of a virtualised network to test and make timing measurements of various operations at scale

SONY

23



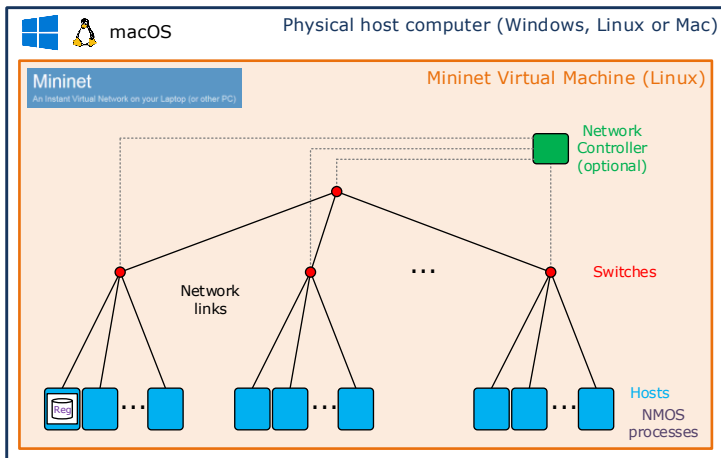
SONY

24

### AMWA NMOS Scalability Study – Methodology

- Use Mininet virtualised network, extended for NMOS, to simulate large number of network endpoints

```
$ sudo nmos-mn -topo=tree,2,40
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 ... h1600
*** Adding switches:
s1 ... s41
*** Adding links:
(h1, s1) (h2, s1) ...
*** Starting controller
c0
*** Starting 41 switches
s1 ... s41
*** Starting CLI:
mininet> _
```



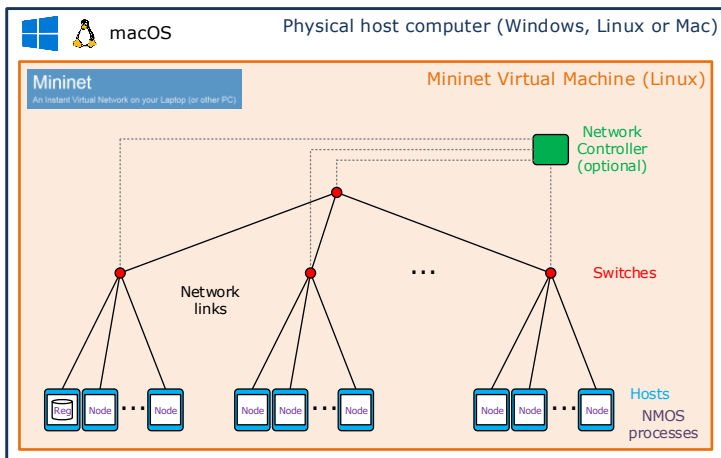
SONY

25

### AMWA NMOS Scalability Study – Methodology

- Run nmos-cpp-registry on one Mininet host

```
mininet> start_registry h1
*** Starting NMOS registry
...
mininet> _
```



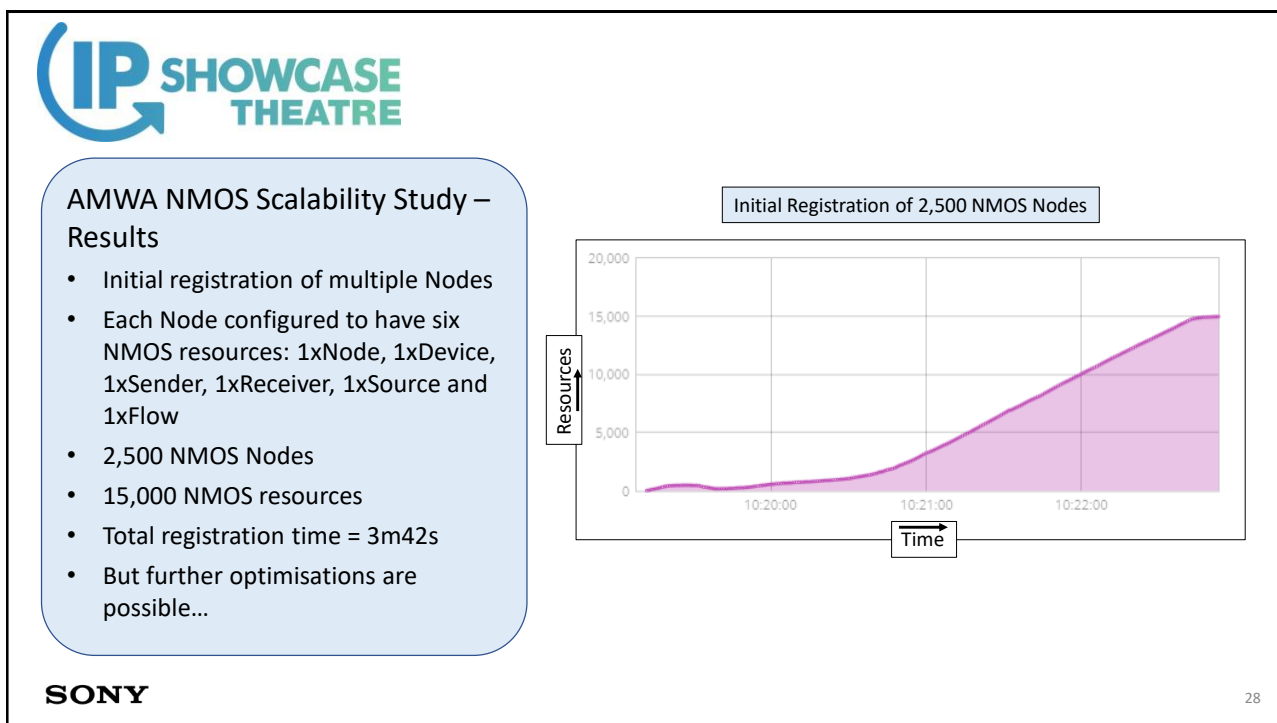
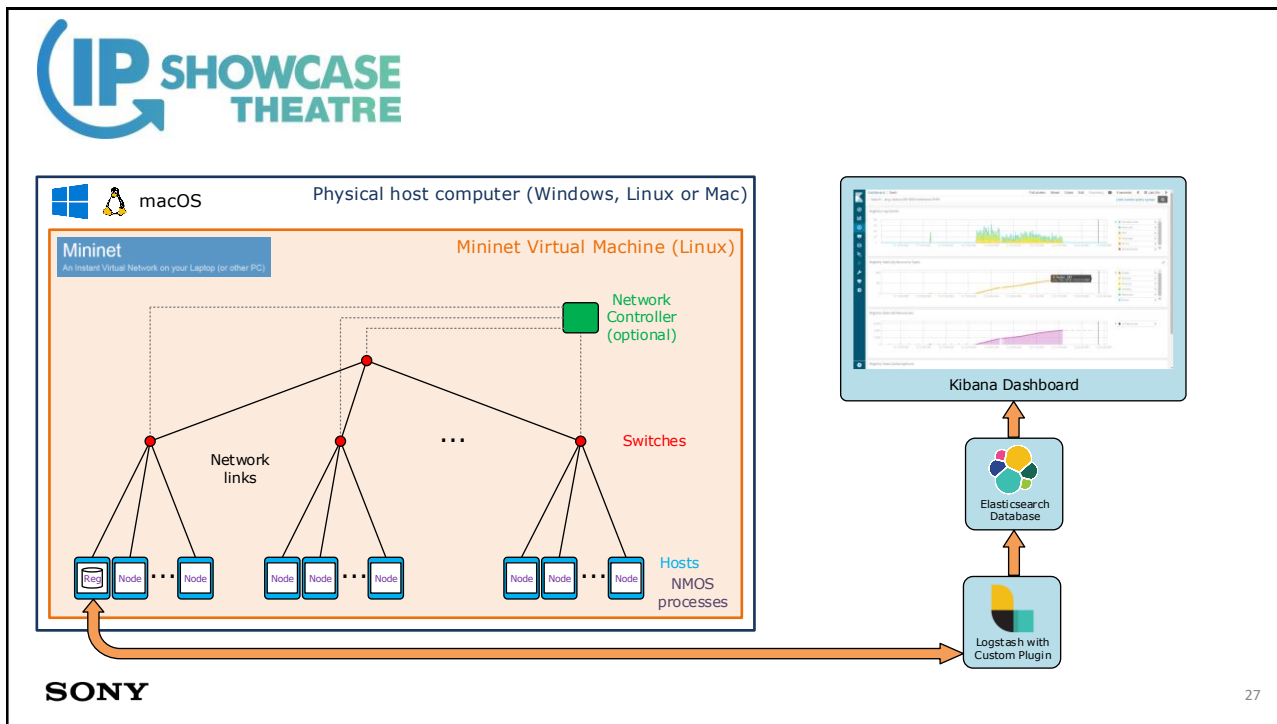
SONY

26

### AMWA NMOS Scalability Study – Methodology

- Run multiple instances of nmos-cpp-node on multiple other Mininet hosts

```
mininet> start_registry h1
*** Starting NMOS registry
...
mininet> start_nodes h2 h1600
*** Starting NMOS nodes
h2 h3 h4 ... h1600
mininet> _
```

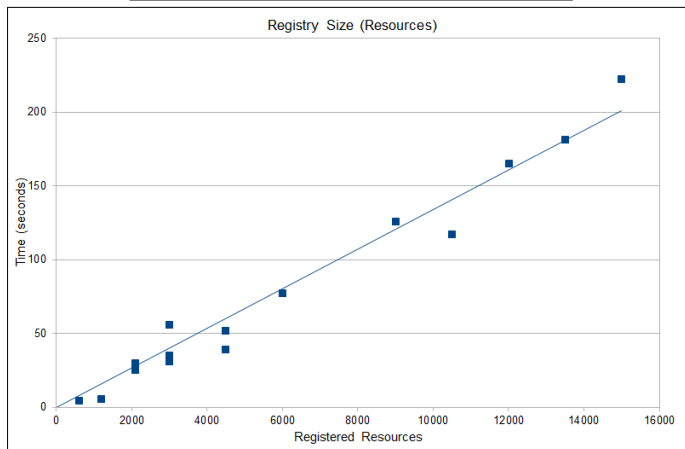




### AMWA NMOS Scalability Study – Results

- Initial registration of multiple Nodes
- Variation with number of Nodes
- 100 - 2,500 NMOS Nodes
- 600 - 15,000 NMOS resources
- Linear relationship in Mininet test environment

Initial Registration of 100 - 2,500 NMOS Nodes



SONY

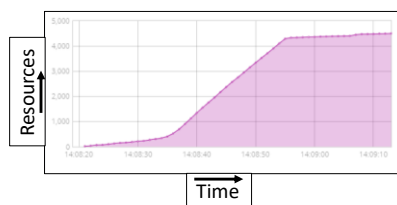
29



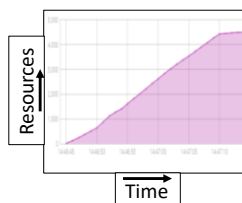
### AMWA NMOS Scalability Study – Results

- Initial registration of multiple Nodes
- Result for 750 NMOS Nodes
  - Slow start and long tail is noticeable
- Make optimisations to:
  - DNS-SD retry interval
  - HTTP timeout before retrying
- Improved result for 750 NMOS Nodes
  - Fast start and short tail
  - Registration time reduced from ~50s to ~30s
- Same improvements for 2,500 Nodes reduce registration time from 3m42s to 2m11s

Initial Registration of 750 NMOS Nodes without Optimisations



Initial Registration of 750 NMOS Nodes with Optimisations



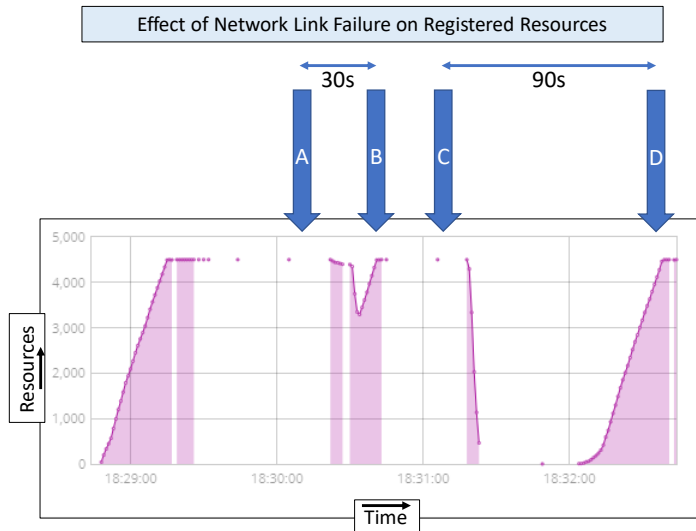
SONY

30



### AMWA NMOS Scalability Study – Results

- Recovery of Registry after a Network link failure
- Deployment with 750 NMOS Nodes
- 4,500 NMOS resources
- Network link failure for 5s at A
  - Full recovery by B
- Network link failure for 30s at C
  - Full recovery by D
- Effect of changing heartbeat interval and registration expiry interval



SONY

31



### AMWA NMOS Scalability Study – Results

#### Registry discovery

- Multicast DNS-SD
  - Sufficient for small layer 2 networks
- Unicast DNS-SD
  - Necessary for layer 3 networks
  - Improved registration performance for large deployments

#### Multiple Registries

- Improved registration performance
- Better tolerance to network link failures

#### Bulk API for registration?

- Time taken for Nodes to fully register depends on number of sub-resources
- Requires many single-resource requests
- Potential enhancement is for a bulk API

SONY

32





## AMWA NMOS IS-04 and IS-05 Scalability and Performance

### Part 1

#### Overview of AMWA IS-04 and IS-05

- APIs
- Open Source Software

### Part 2

#### AMWA NMOS Scalability Study

- Methodology
- Results

### Part 3

#### Discussion

- Best Practice Recommendations
- Future Work

SONY

33



### Best Practice Recommendations

Make timeouts and retry intervals configurable for:

- DNS-SD
- HTTP

Use default recommendations for:

- Heartbeat interval (5s)
- Registration expiry interval (12s)

Use clustered / federated Registries

- Better fault tolerance
- Improved registration performance

For Registry discovery, support unicast DNS-SD as well as multicast DNS-SD

- Greater scalability
- Improved registration performance

SONY

34



### Future Work

- Test Connection Management at scale (in preparation)
- Test effect of multiple network interfaces for redundancy
- Evaluate a bulk API for registration
- Confirm tests with other implementations
  - Environment is shared with all AMWA members so please get involved!
- Repeat tests on a real network with physical media endpoints
  - Scaled down to be practical in a test environment

**SONY**

35



### More

- Full paper to be presented at SMPTE Annual Technical Conference in Los Angeles, 22<sup>nd</sup>-25<sup>th</sup> October 2018
  - “Scalability and Performance of the AMWA IS-04 and IS-05 NMOS Specifications for Networked Media”
  - Please come along for more detailed results and explanation
- Tutorial presentation was given on IP Showcase Booth at NAB 2018
  - “Getting Started with NMOS IS-04 and IS-05”
  - Available to view at <https://www.theiabm.org/getting-started-nmos-04-05/>
- Sony nmos-cpp Open Source Software
  - <https://github.com/sony/nmos-cpp>

**SONY**

36



# Thank You

Rob Porter, Sony Europe Limited  
Rob.Porter@sony.com

**SONY**

IP SHOWCASE THEATRE AT IBC – SEPT. 14-18, 2018