EMANE User Training Solutions

0.6.2

CenGen Inc
9250 Bendix Road North
Columbia, MD 21045

cengen.com
labs.cengen.com/emane

EMANE User Training Workbook

Solutions

0.6.2

CenGen Labs

EMANE User Training Workbook: Solutions: 0.6.2

CenGen Labs

Copyright © 2010 CenGen Inc., Columbia, Maryland

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of CenGen Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

Preface vi
How to use this solution manual vi
Workshop 00 - EMANE Applications and Documentation 1
Problem Statement 1
Workshop 01 - Centralized Deployment 2
Deployment Description 2
Problem Statement 2
Platform Attributes 3
Platform Parameters 3
NEM Attributes 4
NEM Parameters 4
XML Listing 4
Concept Review 5
Workshop 02 - Distributed Deployment 7
Deployment Description 7
Problem Statement 7
Platform Attributes 8
Platform Parameters 8
NEM Attributes 9
NEM Parameters 9
XML Listing 9
Concept Review 11
Workshop 03 - Hybrid Deployment 12
Deployment Description 12
Problem Statement 12
Platform Attributes 13
Platform Parameters 13
NEM Attributes 14
NEM Parameters 14
XML Listing 14
Concept Review 17
Workshop 04 - Virtual Transport 18
Deployment Description 18
Problem Statement 18
NEM Attributes 18
NEM Parameters 19
Virtual Transport Parameters 19
XML Listing 20
Concept Review 22
Workshop 05 - Raw Transport 23
Deployment Description 23
Problem Statement 23
NEM Attributes 23
NEM Parameters 24
Raw Transport Parameters 24
XML Listing 24
Concept Review 26
Workshop 06 - Interactive Transport 27
Deployment Description 27
Problem Statement 27
NEM Attributes 28

NEM Parameters 28
Interactive Transport Parameters 28
Interactive Commands 28
XML Listing 29
Concept Review 30
Workshop 07 - RF Pipe Model 32
Deployment Description 32
Problem Statement 32
RF Pipe MAC Parameters 33
RF Pipe PHY Parameters 33
Interactive Commands 36
XML Listing 36
Concept Review 37
Workshop 08 - IEEE 802.11 abg Model Model 39
Deployment Description 39
Problem Statement 39
IEEE 802.11 abg MAC Parameters 40
IEEE 802.11 abg PHY Parameters 45
Interactive Commands 47
XML Listing 48
Observations Explained 49
Concept Review 50
Workshop 09 - Comm Effect Model Model 51
Deployment Description 51
Problem Statement 51
Comm Effect Shim Parameters 52
Interactive Commands 53
XML Listing 54
Concept Review 55
Workshop 10 - Mitre Mobility Event Generator 56
Deployment Description 56
Problem Statement 56
Mitre Mobility Generator Parameters 57
XML Listing 58
Concept Review 58
Workshop 11 - Emulation Script Generator 60
Deployment Description 60
Problem Statement 60
Emulation Script Generator Parameters 61
RF Pipe PHY Parameters 61
XML Listing 64
Concept Review 65
Workshop 12 - Emulation Event Log Generator 66
Deployment Description 66
Problem Statement 66
Emulation Event Log Generator Parameters 67
XML Listing 67
Concept Review 67
Workshop 13 - GPSd Location Agent 69
Deployment Description 69
Problem Statement 69
GPSd Location Agent Parameters 69
XML Listing 69
Concept Review 70

Workshop 14 - EMANE Loss Controller 71
Deployment Description 71
Problem Statement 71
XML Listing 71
Concept Review 72
Workshop 15 - EMANE Comm Effect Controller 73
Deployment Description 73
Problem Statement 73
XML Listing 73
Concept Review 74
Workshop 16 - Surrogate Radio Models using RF Pipe 75
Deployment Description 75
Problem Statement 75
XML Listing 76
Workshop 17 - libemaneeventservice 77
Description 77
Publisher Sample Application 77
Subscriber Sample Application 77
libemaneeventservice Configuration 77
Configuration Format 78
Workshop 18 - libemaneeventservice Perl language bindings 79
Description 79
Sample Scripts 79
Workshop 19 - libemaneeventservice Python language bindings 80
Description 80
Sample Scripts 80
Workshop 20 - Putting it All together: Distributed RF Pipe MANET 81
Deployment Description 81
Workshop 21 - Putting it All together: Centralized RF Pipe MANET 82
Deployment Description 82
Workshop 22 - Putting it All together: Distributed Comm Effect MANET 83
Deployment Description 83
Workshop 23 - Putting it All together: Centralized Comm Effect MANET 84
Deployment Description 84
Workshop 24 - Putting it All together: Centralized IEEE 802.11 abg MANET 85
Deployment Description 85

Preface

How to use this solution manual

This manual contains solutions to the workshop XML activities and concept review questions found in the EMANE User Training Workbook. The solution XML was generated using Training Id 4.

Workshop 00 - EMANE Applications and Documentation

The goal of this workshop is to become familiar with the EMANE applications and documentation.

Problem Statement

Examine the man pages of each of the EMANE applications and utilities. Locate and examine the available documentation and sample XML.

Application man pages:

- **emane** - NEM Platform Server

```
> man emane
```

- **emanetransportd** - Transport Daemon

```
> man emanetransportd
```

- **emaneeventservice** - Event Service

```
> man emaneeventservice
```

- **emaneeventd** - Event Daemon

```
> man emaneeventd
```

Utility man pages:

- **emaneloglevel** - Dynamically modify log level of running **emane** process

```
> man emaneloglevel
```

- **emanegentransportxml** - Generate transport XML from platform XML

```
> man emanegentransportxml
```

- **emanegendeploymentxml** - Generate deployment XML from platform XML

```
> man emanegendeploymentxml
```

- **emanegenxml** - Generate all XML required for an EMANE deployment

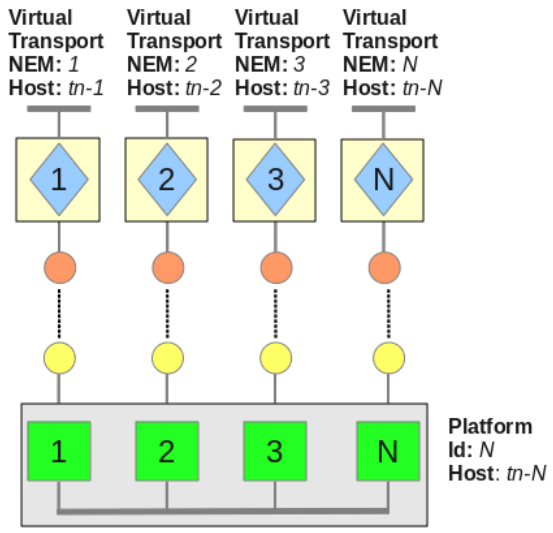
```
> man emanegenxml
```

Workshop 01 - Centralized Deployment

The goal of this workshop is to become familiar with configuring a centralized EMANE deployment.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Virtual Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 4 NEMs with incomplete configuration. Modify the NEM Platform Server configuration to achieve the following:

- Use Platform Id N, where N is your Training Id.
- Disable OTA Manager channel communication.
- The event service multicast group is *224.1.2.8* and port *45703*.
- The event service device is *lo*.
- All NEMs should use the RF Pipe Model.
- Use NEM Ids 1, 2, 3, and N, where N is your Training Id.
- Use NEM Platform Server *tn-N* or *10.99.0.N*, where N is your Training Id.
- Use test node addresses *tn-1*, *tn-2*, *tn-3*, and *tn-N* or *10.99.0.1*, *10.99.0.2*, *10.99.0.3*, and *10.99.0.N*, where N is your Training Id.
- For NEM N use the virtual interface address *radio-N* or *10.100.0.N*, where N is your Training Id.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Platform Attributes

Name:	**name**
Description:	Descriptive name of the platform.
Type:	string

Name:	**id**
Description:	Platform identifier. **Must be unique for all platforms in the deployment.**
Type:	UINT16

Platform Parameters

Name:	**otamanagergroup**
Description:	The OTA Manager mulitcast endpoint that is used to communicate between all platforms in a test deployment. GROUP:PORT format. **Required only for multi-platform EMANE deployments.**
Note:	See *otamanagerdevice*.
Type:	string
XML Format:	<param name="otamanagergroup" value="224.1.2.8:45702"/>

Name:	**otamanagerdevice**
Description:	The network device to associate with OTA Manager mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *otamanagergroup*.
Type:	string
XML Format:	<param name="otamanagerdevice" value="eth1"/>

Name:	**otamanagerchannelenable**
Description:	The OTA Manager Channel Enable parameter enables or disables the OTA Manager channel. When disabled there is no platform to platform communication. Only locally managed NEMs will be able to communicate with each other.
Type:	bool
XML Format:	<param name="otamanagerchannelenable" value="on"/>

Name:	**eventservicegroup**
Description:	The Event Service multicast endpoint that is used to transmit events to and from the platform. GROUP:PORT format.
Note:	See *eventservicedevice*.
Type:	string
XML Format:	<param name="eventservicegroup" value="224.1.2.8:45703"/>

Name:	**eventservicedevice**
Description:	The network device to associate with Event Service mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *eventservicegroup*.
Type:	string
XML Format:	<param name="eventservicedevice" value="eth1"/>

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set Platform name and id
   * Set otamanagerchannelenable value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 4" id="4">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <!--
     * Set NEM 1 definition
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8101"/>
    <param name="transportendpoint" value="tn-1:8201"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-1" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>

  <!--
     * Set NEM 2 name, id and definition
     * Set platform endpoint address
```

```
     * Set transport endpoint address
    -->
  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8102"/>
    <param name="transportendpoint" value="tn-2:8202"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-2" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>

  <!--
     * Set NEM 3 name, id and definition
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8103"/>
    <param name="transportendpoint" value="tn-3:8203"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-3" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>

  <!--
     * Set NEM N name, id and definition
     * Set platform endpoint address
     * Set transport endpoint address
     * Set transport parameter address
    -->
  <nem name="NODE-004" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8104"/>
    <param name="transportendpoint" value="tn-4:8204"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-4" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Review the **rfpipenem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="RF-PIPE NEM">
  <mac definition="rfpipemac.xml"/>
  <phy definition="rfpipephy.xml"/>
  <transport definition="transvirtual.xml"/>
</nem>
```

Concept Review

1. Why was the OTA Manager channel disabled in this workshop?

   NEM Platform Servers communicate via the OTA Manager channel. Since there is only a single NEM Platform Server there is no need to enable the OTA Manager Channel.

2. How do NEMs hosted by the same NEM Platform Server communicate?

   NEMs hosted by the same NEM Platform Server communicate internal to the platform using shared memory.

3. What can be deduced from setting the Event Service device to *lo*?

   Setting the Event Service device to the loopback (lo) implies that the NEM Platform Server is running on the same machine since that is the only way it could receive events.

4. What are the implications of not specifying an Event Service device?

   If the Event Service device is not specified, the kernel routing table is used to associate a device with the Event Service group (multicast group). You either need a specific route for the group or a default route. Missing both of these will result in an EMANE start exception.

5. What are the Platform *name* and NEM *name* XML attributes used for?

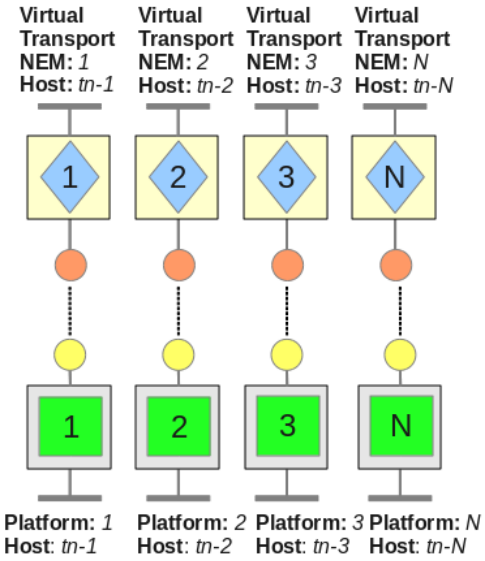
   These are descriptive string names that are not used by any EMANE component. Users are free to use names that convey meaningful information. For example, mapping an NEM to a scenario node name.

Workshop 02 - Distributed Deployment

The goal of this workshop is to become familiar with configuring a distributed EMANE deployment.

Deployment Description

This workshop contains a distributed RF Pipe Deployment using Virtual Transports. There are 4 NEM Platform Servers each hosting a single NEM.



Problem Statement

The 4 NEM Platform Server files `platform1.xml`, `platform2.xml`, `platform3.xml`, and `platformN.xml` contain incomplete configuration. Modify the NEM Platform Server configurations to achieve the following:

- Use Platform Ids 1, 2, 3, and N, where N is your Training Id.
- The OTA Manager mulitcast group is *224.1.2.8* and port *45702*.
- The OTA Manager device is *eth1*.
- The event service multicast group is *224.1.2.8* and port *45703*.
- The event service device is *eth1*.
- All NEMs should use the RF Pipe Model.
- Use NEM Ids 1, 2, 3, and N, where N is your Training Id.
- Use test node addresses *tn-1*, *tn-2*, *tn-3*, and *tn-N* or *10.99.0.1*, *10.99.0.2*, *10.99.0.3*, and *10.99.0.N*, where N is your Training Id.
- For NEM N use the virtual interface address *radio-N* or *10.100.0.N*, where N is your Training Id.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Platform Attributes

Name:	**name**
Description:	Descriptive name of the platform.
Type:	string

Name:	**id**
Description:	Platform identifier. **Must be unique for all platforms in the deployment.**
Type:	UINT16

Platform Parameters

Name:	**otamanagergroup**
Description:	The OTA Manager mulitcast endpoint that is used to communicate between all platforms in a test deployment. GROUP:PORT format. **Required only for multi-platform EMANE deployments.**
Note:	See *otamanagerdevice*.
Type:	string
XML Format:	<param name="otamanagergroup" value="224.1.2.8:45702"/>

Name:	**otamanagerdevice**
Description:	The network device to associate with OTA Manager mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *otamanagergroup*.
Type:	string
XML Format:	<param name="otamanagerdevice" value="eth1"/>

Name:	**otamanagerchannelenable**
Description:	The OTA Manager Channel Enable parameter enables or disables the OTA Manager channel. When disabled there is no platform to platform communication. Only locally managed NEMs will be able to communicate with each other.
Type:	bool
XML Format:	<param name="otamanagerchannelenable" value="on"/>

Name:	**eventservicegroup**
Description:	The Event Service multicast endpoint that is used to transmit events to and from the platform. GROUP:PORT format.
Note:	See *eventservicedevice*.
Type:	string
XML Format:	<param name="eventservicegroup" value="224.1.2.8:45703"/>

Name:	**eventservicedevice**
Description:	The network device to associate with Event Service mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *eventservicegroup*.
Type:	string
XML Format:	<param name="eventservicedevice" value="eth1"/>

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

XML Listing

Modify the **platform1.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set otamanagergroup value
   * Set otamanagerdevice value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 1" id="1">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <!--
     * Set NEM 1 definition
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-1:8101"/>
    <param name="transportendpoint" value="tn-1:8201"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-1" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Modify the **platform2.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set otamanagergroup value
   * Set otamanagerdevice value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 2" id="2">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <!--
     * Set NEM 2 definition
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-2:8102"/>
    <param name="transportendpoint" value="tn-2:8202"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-2" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Modify the **platform3.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set otamanagergroup value
   * Set otamanagerdevice value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 3" id="3">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <!--
     * Set NEM 3 definition
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-3:8103"/>
    <param name="transportendpoint" value="tn-3:8203"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-3" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>

</platform>
```

Modify the **platformN.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set otamanagergroup value
```

```
   * Set otamanagerdevice value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 4" id="4">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <!--
     * Set NEM N name, id, and  definition
     * Set platform endpoint address
     * Set transport endpoint address
     * Set transport parameter address
    -->
  <nem name="NODE-004" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8104"/>
    <param name="transportendpoint" value="tn-4:8204"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-4" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Review the **rfpipenem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="RF-PIPE NEM">
  <mac definition="rfpipemac.xml"/>
  <phy definition="rfpipephy.xml"/>
  <transport definition="transvirtual.xml"/>
</nem>
```

Concept Review

1. Why was the OTA Manager channel enabled in this workshop?

   This workshop contains a distributed EMANE deployment. Distributed deployments contain multiple NEM Platform Servers which communicate via the OTA Manager channel.

2. Why was the Event Service device set to *eth1*?

   Since there are multiple NEM Platform Servers, each residing on a different host, EMANE events must be routed towards the appropriate interface the communicate, in this case *eth1*.

3. What are the implications of not specifying an OTA Manager device?

   If the OTA Manager device is not specified, the kernel routing table is used to associate a device with the OTA Manager group (multicast group). You either need a specific route for the group or a default route. Missing both of these will result in an EMANE start exception.

4. What are the EMANE components associated with the deployment type?

   The three types of EMANE deployment terms centralized, distributed, and hybrid are used to describe *where* the emulation is occurring. This translates to how many NEM platform Servers are being used and how many NEMs are being hosted.

5. When is it possible to use **localhost** for the *platformendpoint* and *transportendpoint* parameters?

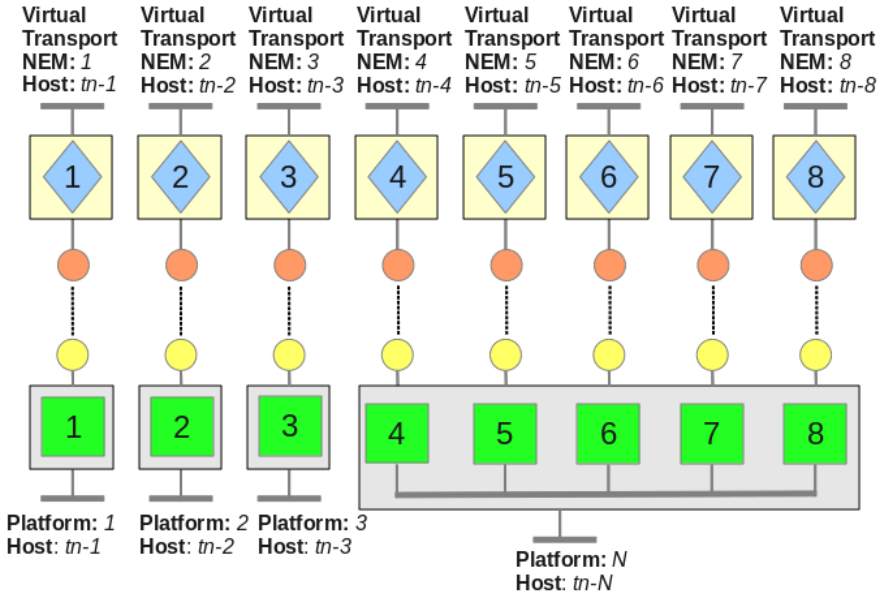
   When an NEM being hosted by an NEM Platform Server and the transport instance associated with that NEM are located on the same host machine.

Workshop 03 - Hybrid Deployment

The goal of this workshop is to become familiar with configuring a hybrid EMANE deployment.

Deployment Description

This workshop contains a hybrid RF Pipe Deployment using Virtual Transports. There are 3 NEM Platform Servers each hosting a single NEM and 1 NEM Platform Server hosting 5 NEMs.



Problem Statement

There are 4 NEM Platform Server files `platform1.xml`, `platform2.xml`, `platform3.xml`, and `platformN.xml`. `platformN.xml` contains incomplete configuration. Modify the NEM Platform Server configuration to achieve the following:

- Use Platform Id N, where N is your Training Id.
- The OTA Manager mulitcast group is *224.1.2.8* and port *45702*.
- The OTA Manager device is *eth1*.
- The event service multicast group is *224.1.2.8* and port *45703*.
- The event service device is *eth1*.
- All NEMs should use the RF Pipe Model.
- Use NEM Ids 4 - 8.
- Platform 1 contains NEM 1 and resides on *tn-1*.
- Platform 2 contains NEM 2 and resides on *tn-2*.
- Platform 3 contains NEM 3 and resides on *tn-3*.
- Platform N contains NEM 4 - 8 and resides on *tn-N*, where N is your Training Id.

  - NEM 4's respective transport daemon resides on *tn-4*.
  - NEM 5's respective transport daemon resides on *tn-5*.
  - NEM 6's respective transport daemon resides on *tn-6*.
  - NEM 7's respective transport daemon resides on *tn-7*.
  - NEM 8's respective transport daemon resides on *tn-8*.
- Use test node addresses *tn-1 - tn-8* and tn-N, or *10.99.0.1 - 10.99.0.8* and *10.99.0.N*, where N is your Training Id.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Platform Attributes

Name:	**name**
Description:	Descriptive name of the platform.
Type:	string

Name:	**id**
Description:	Platform identifier. **Must be unique for all platforms in the deployment.**
Type:	UINT16

Platform Parameters

Name:	**otamanagergroup**
Description:	The OTA Manager mulitcast endpoint that is used to communicate between all platforms in a test deployment. GROUP:PORT format. **Required only for multi-platform EMANE deployments.**
Note:	See *otamanagerdevice*.
Type:	string
XML Format:	<param name="otamanagergroup" value="224.1.2.8:45702"/>

Name:	**otamanagerdevice**
Description:	The network device to associate with OTA Manager mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *otamanagergroup*.
Type:	string
XML Format:	<param name="otamanagerdevice" value="eth1"/>

Name:	**otamanagerchannelenable**
Description:	The OTA Manager Channel Enable parameter enables or disables the OTA Manager channel. When disabled there is no platform to platform communication. Only locally managed NEMs will be able to communicate with each other.
Type:	bool
XML Format:	<param name="otamanagerchannelenable" value="on"/>

Name:	**eventservicegroup**
Description:	The Event Service multicast endpoint that is used to transmit events to and from the platform. GROUP:PORT format.
Note:	See *eventservicedevice*.
Type:	string
XML Format:	<param name="eventservicegroup" value="224.1.2.8:45703"/>

Name:	**eventservicedevice**
Description:	The network device to associate with Event Service mulitcast endpoint. If missing the kernel routing table is used to route multicast joins and packet transmissions.
Note:	See *eventservicegroup*.
Type:	string
XML Format:	<param name="eventservicedevice" value="eth1"/>

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

XML Listing

Review the **platform1.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
```

```
  <param name="eventservicedevice" value="eth1"/>

  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-1:8101"/>
    <param name="transportendpoint" value="tn-1:8201"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-1" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Review the **platform2.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 2" id="2">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-2:8102"/>
    <param name="transportendpoint" value="tn-2:8202"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-2" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>
</platform>
```

Review the **platform3.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 3" id="3">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>

  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-3:8103"/>
    <param name="transportendpoint" value="tn-3:8203"/>
    <transport definition="transvirtual.xml">
      <param name="address" value="radio-3" />
      <param name="mask" value="255.255.255.0" />
    </transport>
  </nem>

</platform>
```

Modify the **platformN.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set Platform name and id
   * Set otamanagerchannelenable value
   * Set eventservicegroup value
   * Set eventservicedevice value
  -->
<platform name="Platform 4" id="4">
  <param name="otamanagergroup" value="224.1.2.8:45702"/>
  <param name="otamanagerdevice" value="eth1"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>
```

```
<!--
   * Set NEM 4 definition
   * Set platform endpoint address
   * Set transport endpoint address
  -->
<nem name="NODE-004" id="4" definition="rfpipenem.xml">
  <param name="platformendpoint" value="tn-4:8104"/>
  <param name="transportendpoint" value="tn-4:8204"/>
  <transport definition="transvirtual.xml">
    <param name="address" value="radio-4" />
    <param name="mask" value="255.255.255.0" />
  </transport>
</nem>

<!--
   * Set NEM 5 definition
   * Set platform endpoint address
   * Set transport endpoint address
  -->
<nem name="NODE-005" id="5" definition="rfpipenem.xml">
  <param name="platformendpoint" value="tn-4:8105"/>
  <param name="transportendpoint" value="tn-5:8205"/>
  <transport definition="transvirtual.xml">
    <param name="address" value="radio-5" />
    <param name="mask" value="255.255.255.0" />
  </transport>
</nem>

<!--
   * Set NEM 6 definition
   * Set platform endpoint address
   * Set transport endpoint address
  -->
<nem name="NODE-006" id="6" definition="rfpipenem.xml">
  <param name="platformendpoint" value="tn-4:8106"/>
  <param name="transportendpoint" value="tn-6:8206"/>
  <transport definition="transvirtual.xml">
    <param name="address" value="radio-6" />
    <param name="mask" value="255.255.255.0" />
  </transport>
</nem>

<!--
   * Set NEM 7 definition
   * Set platform endpoint address
   * Set transport endpoint address
  -->
<nem name="NODE-007" id="7" definition="rfpipenem.xml">
  <param name="platformendpoint" value="tn-4:8107"/>
  <param name="transportendpoint" value="tn-7:8207"/>
  <transport definition="transvirtual.xml">
    <param name="address" value="radio-7" />
    <param name="mask" value="255.255.255.0" />
  </transport>
</nem>


<!--
   * Set NEM 8 definition
   * Set platform endpoint address
   * Set transport endpoint address
  -->
<nem name="NODE-008" id="8" definition="rfpipenem.xml">
  <param name="platformendpoint" value="tn-4:8108"/>
  <param name="transportendpoint" value="tn-8:8208"/>
  <transport definition="transvirtual.xml">
    <param name="address" value="radio-8" />
    <param name="mask" value="255.255.255.0" />
  </transport>
</nem>
```

```
</platform>
```

Review the **rfpipenem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="RF-PIPE NEM">
  <mac definition="rfpipemac.xml"/>
  <phy definition="rfpipephy.xml"/>
  <transport definition="transvirtual.xml"/>
</nem>
```

Concept Review

1. What is the `deployment.xml` file?

   A `deployment.xml` is created by the **emanegendepolymentxml** utility. This utility takes as input all the platform XML files included in the deployment. The resulting deployment XML is a mapping of NEM to NEM Platform Server locations. This file is used by the Event Service.

2. What is the communication path for messages sent from tn-1 to tn-8 via the OTA?

   Data received from tn-1's associated transport destined for tn-8 first travels through the NEM 1 stack. Next it is transmitted via Platform 1's OTA Manager and received by Platform N's OTA Manager. Platform N's OTA Manager delivers the data to the NEM 8 stack which would then transmit the data to its associated transport.

3. What is the communication path for messages sent from tn-5 to tn-8 via the OTA?

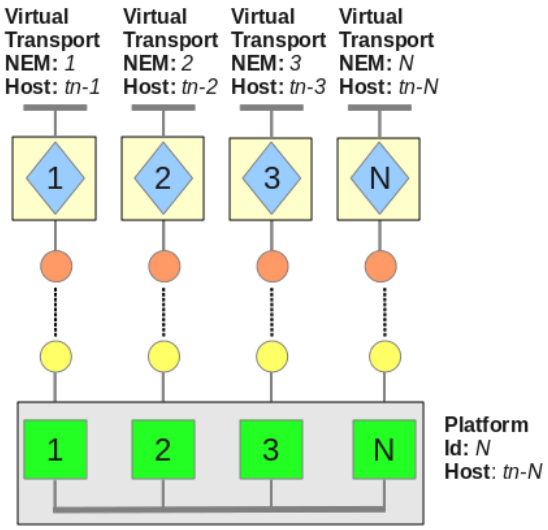
   Data received from tn-5's associated transport destined for tn-8 first travels through the NEM 5 stack. Next it is delivered internally to the NEM 8 stack via Platform N's OTA Manager. The NEM 8 stack transmits the data to its associated transport.

Workshop 04 - Virtual Transport

The goal of this workshop is to become familiar with configuring a Virtual Transport.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Virtual Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 4 NEMs with incomplete configuration. Modify the XML configuration to achieve the following:

- Use the NEM Platform Server *tn-N* or *10.99.0.N*, where N is your Training Id.
- Use Platform Id N, where N is your Training Id.
- NEM 1 virtual device address use *radio-1* or *10.100.0.1* with a 24 bit mask.
- NEM 2 virtual device address use *radio-2* or *10.100.0.2* with a 24 bit mask.
- NEM 3 virtual device address use *radio-3* or *10.100.0.3* with a 24 bit mask.
- NEM N virtual device address use *radio-N* or *10.100.0.N* with a 24 bit mask, where N in your Training Id.
- NEM N virtual device name is *manet0*.
- All NEMs should be limited to *1Mpbs*.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

Virtual Transport Parameters

Name:	**bitrate**
Description:	Transport bitrate in Kbps. This is the total allowable throughput for the transport. A value of 0 disables the bitrate feature. The maximum value is only limited to the storage size of the type uint64 18446744073709551615 / 1024, or 18014398509481983.
Type:	Double
Default:	8096
XML Format:	<param name="bitrate" value="500"/>

Name:	**arpcacheenable**
Description:	Enable ARP request/reply monitoring to map ethernet address to NEM.
Type:	Boolean
Default:	on
XML Format:	<param name="arpcacheenable" value="on"/>

Name:	**devicepath**
Description:	Path to the tap device.
Type:	string
Default (linux):	/dev/net/tun
Default (OSX):	/dev/tap0
Default (Win32):	SYSTEM\\CurrentControlSet\\Control\\Network\\{4D36E972-E325-11CE-BFC1-08002BE10318}
XML Format:	<param name="devicepath" value="/dev/net/tun"/>

Name:	**device**
Description:	Virtual device name. Note: on OS X this *must* be tap0.
Type:	string
Default:	emane0
XML Format:	<param name="device" value="emane1"/>

Name:	**address**
Description:	Virtual device address. Supports IPv4 and IPv6.
Type:	string
Default:	none
XML Format:	<param name="address" value="172.30.1.1"/>

Name:	**mask**
Description:	Virtual device network mask. Supports IPv4 and IPv6.
Type:	string
Default:	none
XML Format:	<param name="mask" value="255.255.0.0"/>

Name:	**arpmode**
Description:	Enable arp on the virtual device.
Type:	boolean
Default:	on
XML Format:	<param name="arpmode" value="on"/>

Name:	**broadcastmode**
Description:	Broadcast all packets to all NEM's.
Type:	boolean
Default:	off
XML Format:	<param name="broadcastmode" value="on"/>

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set Platform name and id
  -->
<platform name="Platform 4" id="4">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <!--
     * Set NEM 1 platform endpoint address
    -->
  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8101"/>
    <param name="transportendpoint" value="tn-1:8201"/>
    <transport definition="transvirtual.xml">
```

```
      <!--
         * Set transport parameter address
         * Set transport parameter mask
        -->
      <param name="address" value="radio-1"/>
      <param name="mask" value="255.255.255.0"/>
    </transport>
  </nem>

  <!--
     * Set NEM 2 platform endpoint address
    -->
  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8102"/>
    <param name="transportendpoint" value="tn-2:8202"/>
    <transport definition="transvirtual.xml">
      <!--
         * Set transport parameter address
         * Set transport parameter mask
        -->
      <param name="address" value="radio-2"/>
      <param name="mask" value="255.255.255.0"/>
    </transport>
  </nem>


  <!--
     * Set NEM 3 platform endpoint address
    -->
  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8103"/>
    <param name="transportendpoint" value="tn-3:8203"/>
    <transport definition="transvirtual.xml">
      <!--
         * Set transport parameter address
         * Set transport parameter mask
        -->
      <param name="address" value="radio-3"/>
      <param name="mask" value="255.255.255.0"/>
    </transport>
  </nem>


  <!--
     * Set NEM N name and id
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-4" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8104"/>
    <param name="transportendpoint" value="tn-4:8204"/>
    <transport definition="transvirtual.xml">
      <!--
         * Set transport parameter address
         * Set transport parameter mask
         * Set transport parameter device
        -->
      <param name="address" value="radio-4"/>
      <param name="mask" value="255.255.255.0"/>
      <param name="device" value="manet0"/>
    </transport>
  </nem>
</platform>
```

Modify the **transvirtual.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE transport SYSTEM "file:///usr/share/emane/dtd/transport.dtd">
<transport name="Virtual Transport" library="transvirtual">
   <!--
      * Set transport parameter bitrate
```

```
     -->
   <param name="bitrate" value="1000000"/>
</transport>
```

Concept Review

1. What are alternative ways the transport *bitrate* configuration for this workshop could have been specified?

   The transport *bitrate* parameter could have been set once in the `transvirtual.xml` file, or set once in the `rfpipenem.xml` file, or set 4 times, once for each NEM, in the `platform.xml` file.

2. What happens if you do not specify the address and mask for the Virtual Transport? When might this be desirable?

   The virtual interface will be created without any configuration. This might be desirable when you are using a protocol to assign an address to the interface, i.e. dhcp.

3. Are IPv6 virtual addresses supported?

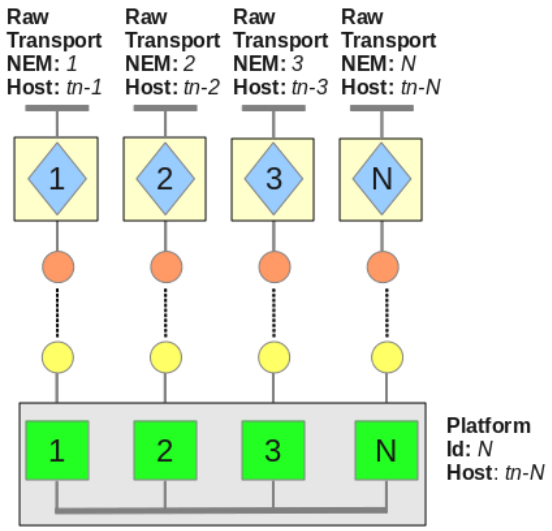
   Yes.

Workshop 05 - Raw Transport

The goal of this workshop is to become familiar with configuring a Raw Transport.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Raw Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 4 NEMs with incomplete configuration. Modify the NEM Platform Server configuration to achieve the following:

- Use NEM Platform Server *tn-N* or *10.99.0.N*, where N is your Training Id.
- Use Platform Id N, where N is your Training Id.
- All NEMs should use the Raw Transport.
- For NEM N use test node address *tn-N* or *10.99.0.N*, where N is your Training Id.
- NEM 1 should use *eth2* as its raw device.
- NEM 2 should use *eth2* as its raw device.
- NEM 3 should use *eth2* as its raw device.
- NEM N should use *eth2* as its raw device.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

Raw Transport Parameters

Name:	**bitrate**
Description:	Transport bitrate in Kbps. This is the total allowable throughput for the transport. A value of 0 disables the bitrate feature. The maximum value is only limited to the storage size of the type uint64 18446744073709551615 / 1024, or 18014398509481983.
Type:	Double
Default:	8096
XML Format:	<param name="bitrate" value="500"/>

Name:	**arpcacheenable**
Description:	Enable ARP request/reply monitoring to map ethernet address to NEM.
Type:	Boolean
Default:	on
XML Format:	<param name="arpcacheenable" value="on"/>

Name:	**device**
Description:	Device to use as the raw packet entry point
Type:	string
Default:	none
XML Format:	<param name="device" value="eth1"/>

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">

<!--
   * Set Platform name and id
  -->
```

```
<platform name="Platform 4" id="4">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <!--
     * Set NEM 1 platform endpoint address
    -->
  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8101"/>
    <param name="transportendpoint" value="tn-1:8201"/>
    <transport definition="transraw.xml">
      <!--
         * Set transport definition
         * Set transport parameter device
        -->
      <param name="device" value="eth2"/>
    </transport>
  </nem>

  <!--
     * Set NEM 2 platform endpoint address
    -->
  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8102"/>
    <param name="transportendpoint" value="tn-2:8202"/>
    <transport definition="transraw.xml">
      <!--
         * Set transport definition
         * Set transport parameter device
        -->
      <param name="device" value="eth2"/>
    </transport>
  </nem>

  <!--
     * Set NEM 3 platform endpoint address
    -->
  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8103"/>
    <param name="transportendpoint" value="tn-3:8203"/>
    <transport definition="transraw.xml">
      <!--
         * Set transport definition
         * Set transport parameter device
        -->
      <param name="device" value="eth2"/>
    </transport>
  </nem>

  <!--
     * Set NEM N name and id
     * Set platform endpoint address
     * Set transport endpoint address
    -->
  <nem name="NODE-4" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="tn-4:8104"/>
    <param name="transportendpoint" value="tn-4:8204"/>
    <transport definition="transraw.xml">
      <!--
         * Set transport definition
         * Set transport parameter device
        -->
      <param name="device" value="eth2"/>
    </transport>
  </nem>
</platform>
```

Review the **transraw.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE transport SYSTEM "file:///usr/share/emane/dtd/transport.dtd">
<transport name="Raw transport" library="transraw"/>
```

Concept Review

1. What routes need to be added to client machines that connect to an interface dedicated for use by a Raw Transport?

   No routes are needed. Basically, if your client machines were able to communicate on the same switch or hub then they should be able to communicate through the Raw Transport.

2. Should applications that route traffic over an interface in use by a Raw Transport be used?

   No. Once an interface is in use by a Raw Transport it should not be used for anything else. All traffic received on that interface is transmitted to the transport's associated NEM, regardless of the traffic's destination. The Raw Transport in essence turns the interface directly connected to it into an emulated wireless interface.

3. What does the *arpcacheenable* parameter do?

   When enabled the *arpcacheenable* parameter instructs the Raw Transport to peek into all ARP response packets and create a mapping of NEM ids to IP addresses. This allows the transport to correctly convert IP destinations into NEM ids. The same is true for the Virtual Transport.

4. How does the Raw Transport address packets when *arpcacheenable* is off?

   When the *arpcacheenable* parameter is off all packets received by the Raw Transport are transmitted to its associate NEM using the NEM broadcast destination. This will result in all the packets being processed as if they we destined for each NEM. This is what normally happens for broadcast messages but not for unicast messages. Unicast messages usually die in the MAC layer if they are not destined for the receiving node. In this case they would make their way all the way to each receiving NEM transport and get dropped by the attached host's kernel.

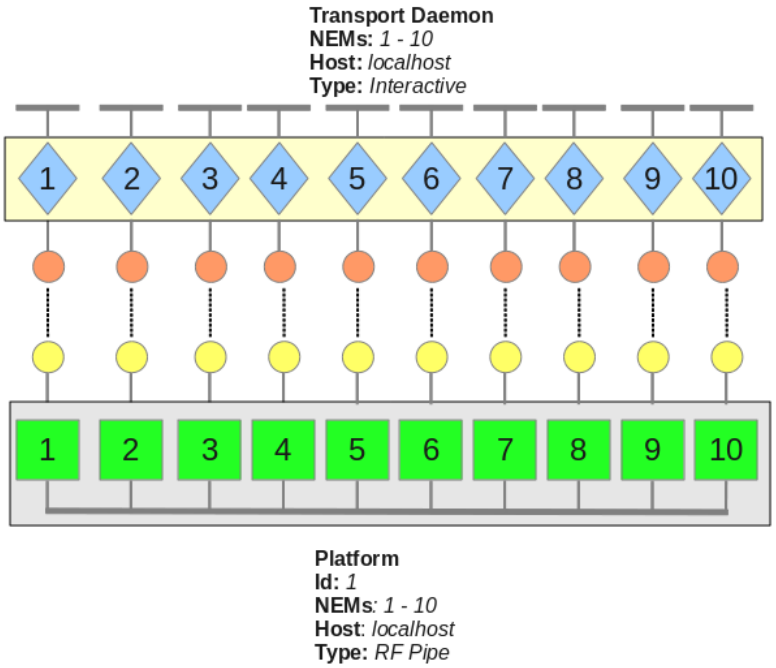
Workshop 06 - Interactive Transport

The goal of this workshop is to become familiar with configuring and interacting with the Interactive Transport.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 10 NEMs. The Interactive Transport implements a telnet server that provides a testing shell which supports transport to transport unicast and broadcast *pinging* with simple statistics. This transport is designed for use with simple model test cases and training activities.

You can log in to each of the 10 NEM transports created by this deployment using any telnet client. Each transport server is listening on a specific port based on the *portbase* transport configuration item and the transport NEM id. Each transport server (instance of transinteractive) only allows one active telnet connection at a time.



Problem Statement

The `platform.xml` file in this workshop contains 10 NEMs with incomplete transport configuration. Modify the NEM Platform Server configuration to achieve the following:

- Set all NEM transport group entries so all are managed by a single instance of the Transport Daemon.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Log in to a few of the interactive sessions and send unicast and broadcast pings between transports.

In this workshop *portbase* is 46000. To connect to an Interactive Transport associated with an NEM you need to connect to the telnet server located at the *portbase* + NEM Id.

For example, to connect to the Interactive Transport for NEM 1:

```
> telnet localhost 46001
```

NEM Attributes

Name:	**name**
Description:	Descriptive name of the NEM.
Type:	string

Name:	**id**
Description:	NEM identifier. **Must be unique for all NEMs in the deployment.**
Type:	UINT16

NEM Parameters

Name:	**platformendpoint**
Description:	The endpoint that the NEM should bind to in order to receive messages from it's respective transport.
Type:	string
XML Format:	<param name="platformendpoint" value="platformserver:8181"/>

Name:	**transportendpoint**
Description:	The transport endpoint that the NEM should use when transmitting messages to it's respective transport.
Type:	string
XML Format:	<param name="transportendpoint" value="tn-1:8181"/>

Interactive Transport Parameters

Name:	**portbase**
Description:	The telnet server will listen on port portbase + NEM id on all interfaces
Type:	UINT16
Default:	46000
XML Format:	<param name="portbase" value="46000"/>

Interactive Commands

The interactive transport command shell supports the following commands:

clear	Clear Screen
exit	Exit telnet session. Same as *quit*.
ping	Transport to Transport ping.
quit	Quit telnet session. Same as *exit*.
stats	Show or clear statistic counters.
version	Print version.
whoami	Print NEM id.

ping Usage

The ping commands sends an interactive transport PING Request to another interactive transport to elicit an interactive transport PING Response.

ping [-i interval] [-c count] [-s size] NEMID|'all'

interval	Wait interval seconds between sending each packet. The default is to wait for one second between each packet.
count	Stop after sending count ping requests packets.
size	Specifies the number of data bytes to be sent. The default is 4, which translates into 64 data bytes when combined with the 60 bytes of interactive ping header.

stats Usage

The stats command displays or clears stats related to inbound and outbound interactive transport PING requests and responses.

stats [clear]

XML Listing

Review the **platform.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <!--
       * Set transport group
      -->
```

```
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-005" id="5" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8105"/>
    <param name="transportendpoint" value="localhost:8205"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-006" id="6" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8106"/>
    <param name="transportendpoint" value="localhost:8206"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-007" id="7" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8107"/>
    <param name="transportendpoint" value="localhost:8207"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-008" id="8" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8108"/>
    <param name="transportendpoint" value="localhost:8208"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-009" id="9" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8109"/>
    <param name="transportendpoint" value="localhost:8209"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-010" id="10" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8110"/>
    <param name="transportendpoint" value="localhost:8210"/>
    <!--
       * Set transport group
      -->
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

</platform>
```

Review the **transinteractive.xml** file.

```
<!DOCTYPE transport SYSTEM "file:///usr/share/emane/dtd/transport.dtd">
<transport name="Interactive transport" library="transinteractive">
  <param name="portbase" value="46000"/>
</transport>
```

Concept Review

1. What data message format (or type of data message) does any NEM expect to receive from an attached Transport?

An NEM does not (should not) expect any format for data messages received from its attached transport. Data is opaque. There is a control message path that allows transports and NEMs to communicate.

2. Can an Interactive Transport interoperate with a Raw or Virtual Transport?

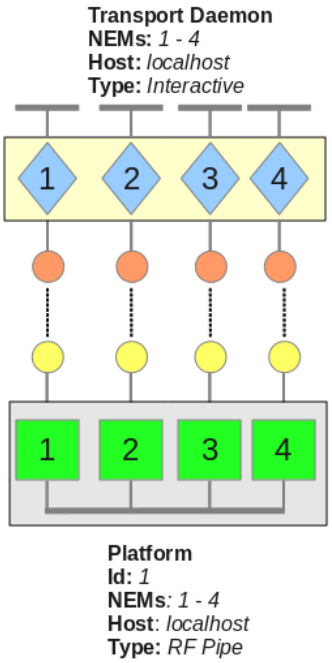
   No. Both the Raw and Virtual Transports encapsulate ethernet frames. The Interactive Transport has its own messaging mechanism that is based on its own protocol.

Workshop 07 - RF Pipe Model

The goal of this workshop is to become familiar with configuring the RF Pipe Model.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 4 NEMs with incomplete configuration. Modify the XML configuration to achieve the following:

- Add a 100 msec *jitter* to NEM 1.
- Add a 500 msec *jitter* to NEM 2.
- Add a 30 msec *jitter* to NEM 3.
- Change the *frequency* for all NEMs to 3000 MHz.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify the settings using the Interactive Transport to unicast and broadcast *ping* between NEMs.

RF Pipe MAC Parameters

Name:	**enablepromiscuousmode**		
Description:	Determines if all packets received over the air will be sent up the stack		
	on	Send all packets up	
	off	Send only broadcast and locally addressed unicast up.	
Type:	Boolean		
Default:	off		
XML Format:	<param name="enablepromiscuousmode" value="off"/>		

Name:	**transmissioncontrolmap**		
Description:	Sends the data rate, frequency, and power level to the PHY for all transmissions to a specified node. When a packet is transmitted to the destination NEM, the PHY will include the specified data rate, frequency, and transmission power in the OTA PHY header. A 0 in any value other than the destination will cause the XML configuration value for that respective RF Pipe PHY parameter to be used.		
	Destination NEM	NEM Id of destination	UINT16
	Data Rate	Data Rate in Kbps	UINT16
	Frequency	Frequency in MHz	UINT16
	Tx Power	Power Level in dBm	UINT8
Type:	string		
Default:	none		
Count:	Unlimited		
XML Format:	<param name="transmissioncontrolmap" value="1:128:0:99"/>		
	<param name="transmissioncontrolmap" value="2:256:0:98"/>		
	<param name="transmissioncontrolmap" value="3:512:0:97"/>		

RF Pipe PHY Parameters

Name:	**frequency**
Description:	Defines the frequency used in MHz for the emulated network. The frequency is included in the PHY header of all over the air messages and serves 2 functions: 1. Filters out packets received from nodes that are on different frequencies. 2. Is used to determine free space path loss when configured to use compute path loss based on node positions using the free space propagation algorithm. See configuration parameters *pathlosssource* and *pathlossalgorithm* below.
Type:	UINT16
Default:	2437
XML Format:	<param name="frequency" value="2437">

Name:	**bandwidth**
Description:	Defines the bandwidth in Khz. Bandwidth is included in the PHY header. The current version does not utilize this information.
Type:	UINT16
Default:	20
XML Format:	<param name="bandwidth" value="20">

Name:	**txpower**
Description:	Defines the transmit the power (dBm) and is used to compute the receive power in conjunction with path loss and antenna gains. The txpower is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="txpower" value="0">

Name:	**antennagain**
Description:	Defines the antenna gain (dBi) and is used to compute the receive power in conjunction with path loss and txpower. The antennagain is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="antennagain" value="0">

Name:	**bitrate**
Description:	Defines the bitrate/burstrate (kbps) of the waveform being emulated. It is used on the transmit side to compute transmission delay based on the packet size and bitrate.
Type:	UINT16
Default:	0
XML Format:	<param name="bitrate" value="0">

Name:	**propagationdelay**
Description:	Defines the propagation delay (usec) that is to be included in the transmission delay. The propagationdelay is added to the delay introduced by the bitrate defined above.
Type:	UINT32
Default:	0
XML Format:	<param name="propagationdelay" value="0">

Name:	**jitter**
Description:	Defines the jitter (usec) to be included to the transmission delay. The jitter will be computed for each packet transmission based on uniform random distribution between +- the configured jitter value.
Type:	UINT32
Default:	0
XML Format:	<param name="jitter" value="0">

Name:	**pathlosssource**
Description:	Defines the method of use for determining path loss. The parameter accepts 2 options: 0 — Pathloss directly provided via an external event 1 — Pathloss computed real time based on external position events and selected *pathlossalgorithm*
Type:	UINT16
Default:	0
XML Format:	<param name="pathlosssource" value="0">

Name:	**pathlossalgorithm**
Description:	Defines the algorithm to be used to compute the path loss. This is valid only when *pathlosssource* is set to *1*. The algorithm selections are as follows: 0 — Free space path loss model 1 — 2 ray flat earth model
Type:	UINT16
Default:	0
XML Format:	<param name="pathlossalgorithm" value="0">

Name:	**rxpowernolossthreshold**
Description:	Defines the receive power threshold (dBm) for determining if packets will be received or dropped. This threshold is used in conjunction with the *rxpowerfulllossthreshold* to determine the probability of packet reception (PoR) based on received power as follows: Rx Power >= rxpowernolossthreshold ==> PoR = 1 rxpowerfulllossthreshold < Rx Power < rxpowernolossthreshold ==> linear packet loss Rx Power <= rxpowerfulllossthreshold ==> PoR = 0
Type:	UINT16
Default:	90
XML Format:	<param name="rxpowernolossthreshold" value="90">

Name:	**rxpowerfulllossthreshold**
Description:	See description of *rxpowernolossthreshold* above.
Type:	UINT16
Default:	110
XML Format:	<param name="rxpowerfulllossthreshold" value="110">

Name:	**defaultconnectivity**
Description:	Defines the default connectivity of all nodes at start up prior to receiving any mobility or path loss events. When set to *on* the reception of a single path loss or position event will disable the default connectivity setting. off — All nodes are assumed isolated on — All nodes are assumed to be fully informed
Type:	Boolean
Default:	on
XML Format:	<param name="defaultconnectivity" value="on">

Interactive Commands

The interactive transport command shell supports the following commands:

clear	Clear Screen
exit	Exit telnet session. Same as *quit*.
ping	Transport to Transport ping.
quit	Quit telnet session. Same as *exit*.
stats	Show or clear statistic counters.
version	Print version.
whoami	Print NEM id.

ping Usage

The ping commands sends an interactive transport PING Request to another interactive transport to elicit an interactive transport PING Response.

ping [-i interval] [-c count] [-s size] NEMID|'all'

interval	Wait interval seconds between sending each packet. The default is to wait for one second between each packet.
count	Stop after sending count ping requests packets.
size	Specifies the number of data bytes to be sent. The default is 4, which translates into 64 data bytes when combined with the 60 bytes of interactive ping header.

stats Usage

The stats command displays or clears stats related to inbound and outbound interactive transport PING requests and responses.

stats [clear]

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
```

```
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <phy definition="rfpipephy.xml">
      <!--
         Set PHY parameter jitter
        -->
      <param name="jitter" value="100000"/>
    </phy>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <phy definition="rfpipephy.xml">
      <!--
         Set PHY parameter jitter
        -->
      <param name="jitter" value="500000"/>
    </phy>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <phy definition="rfpipephy.xml">
      <!--
         Set PHY parameter jitter
        -->
      <param name="jitter" value="30000"/>
    </phy>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>
</platform>
```

Modify the **rfpipephy.xml** file to achieve the workshop goals.

```
<!DOCTYPE phy SYSTEM "file:///usr/share/emane/dtd/phy.dtd">
<phy name="RF-PIPE PHY" library="rfpipephylayer">
  <param name="txpower" value="0"/>
  <param name="antennagain" value="0"/>
  <param name="bitrate" value="0"/>
  <param name="jitter" value="0"/>
  <!--
     Set parameter frequency
    -->
  <param name="frequency" value="3000"/>
  <param name="bandwidth" value="20"/>
  <param name="propagationdelay" value="0"/>
  <param name="pathlosssource" value="1"/>
  <param name="pathlossalgorithm" value="0"/>
  <param name="defaultconnectivity" value="on"/>
  <param name="rxpowernolossthreshold" value="90"/>
  <param name="rxpowerfulllossthreshold" value="110"/>
</phy>
```

Concept Review

1. What does the *defaultconnectivity* parameter control?

   When enabled, the *defaultconnectivity* parameter instructs the RF Pipe PHY layer to accept all messages by bypassing the receive power threshold logic. It will remain in this state until it receives

either a pathloss or location message depending on which pathloss source is selected. After which, the receive power threshold logic determines which messages are sent to the RF Pipe MAC layer. When disabled, no messages are accepted until the correct event type is received.

2. What does the *rxpowernolossthreshold* parameter configure?

   Defines the receive power threshold (dBm) for determining if packets will be received or dropped. This threshold is used in conjunction with the *rxpowerfulllossthreshold* to determine the probability of packet reception (PoR) based on received power as follows:

Rx Power >= -rxpowernolossthreshold	PoR = 1
-rxpowerfulllossthreshold < Rx Power < -rxpowernolossthreshold	linear packet loss
Rx Power <= -rxpowerfulllossthreshold	PoR = 0

3. What does the *rxpowerfulllossthreshold* parameter configure?

   See previous question.

4. Which events does the RF Pipe Model process?

   The RF Pipe Model processes Pathloss or Location events depending on the pathloss source configuration.

5. What controls which event type is used as part of the receive power calculation?

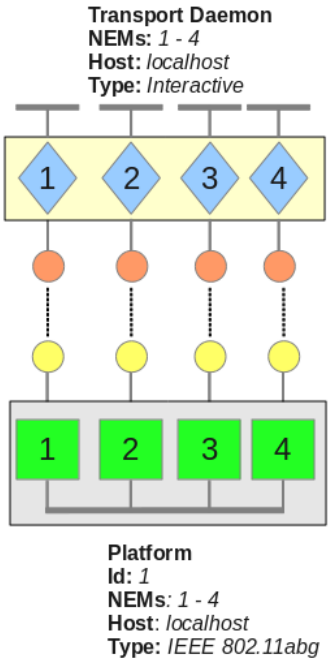
   The *pathlosssource* configuration parameter.

Workshop 08 - IEEE 802.11 abg Model Model

The goal of this workshop is to become familiar with configuring the IEEE 802.11 abg Model.

Deployment Description

This workshop contains a centralized IEEE 802.11 abg Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 4 NEMs with incomplete configuration. Modify the XML configuration to achieve the following:

- Set NEM 1's unicast data rate to 1Mbps
- Set NEM 2's unicast data rate to 1Mbps
- Set NEM 3's unicast data rate to 11Mbps
- Set NEM 4's unicast data rate to 54Mbps

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

1. Verify the settings by using the Interactive Transport:

   a. From NEM 1 interactive transport session send 100 128 byte packets at a rate of 10 packets per second to NEM 2

```
transport-001 %% ping -i .1 -c 100 -s 128 2
```

Use the *stats* command to view the completion statistics. Note the results.

b. From NEM 1 interactive transport session send 100 128 byte packets at a rate of 10 packets per second to NEM 3

```
transport-001 %% ping -i .1 -c 100 -s 128 3
```

Use the *stats* command to view the completion statistics. Note the results.

c. From NEM 1 interactive transport session send 100 128 byte packets at a rate of 10 packets per second to NEM 4

```
transport-001 %% ping -i .1 -c 100 -s 128 4
```

Use the *stats* command to view the completion statistics. Note the results.

2. Stop the workskop components and set the antenna gain for NEM 1 to 13dBi.

Rerun the interactive commands and compare the results with what you observed previously.

IEEE 802.11 abg MAC Parameters

Name:	**mode**	
Description:	Defines the 802.11 mode. The mode is included in the PHY header of all over the air messages and is used to filter out packets received from nodes that are on different modes. Should be noted that mode setting does not care about frequency and is used only to identify the actual physical signal in space.	
	0	802.11b (DSSS only)
	1	802.11b (DSSS only)
	2	802.11 a or g (OFDM)
	3	802.11 b/g (DSSS and OFDM)
	4-255	NOT USED
Type:	UINT8	
Default:	0	
XML Format:	<param name="mode" value="0"/>	

Name:	**enablepromiscuousmode**	
Description:	Determines if all packets received over the air will be sent up the stack.	
	off	Disable or send only broadcast and locally addressed unicast up
	on	Enable or send all packets up
Type:	Boolean	
Default:	off	
XML Format:	<param name="enablepromiscousmode" value="off"/>	

Name:	**beaconinterval**
Description:	The model does not currently support beacon transmissions. The only mode currently supported is ad-hoc demo mode. 0 Disable beacon transmissions
Type:	UINT32
Default:	0
XML Format:	<param name="beaconinterval" value="0"/>

Name:	**distance**
Description:	Defines the maximum distance (meters) for point to point links within the network to be supported. This is used to adjust the slot timing to account for round trip propagation delays.
Type:	UINT32
Default:	1000
XML Format:	<param name="distance" value="1000"/>

Name:	**atpenable**
Description:	The model does not currently support adaptive data rate and power selections. off Disable on Enable **(Not Supported)**
Type:	Boolean
Default:	off
XML Format:	<param name="atpenable" value="off"/>

Name:	**unicastrate**
Description:	Defines the data rate to be used for all unicast transmissions. The rate selection must be valid for the mode selected as defined in the MAC capabilities overview. The rates are defined via an index between 1 through 12.
	0 — 11Mbps (Default)
	1 — 1 Mbps
	2 — 2 Mbps
	3 — 5.5 Mbps
	4 — 11 Mbps
	6 — 6 Mbps
	6 — 9 Mbps
	7 — 12 Mbps
	8 — 18 Mbps
	9 — 24 Mbps
	10 — 36 Mbps
	11 — 48 Mbps
	12 — 54 Mbps
	13-255 — Not Defined.
Type:	UINT8
Default:	0
XML Format:	<param name="unicastrate" value="0"/>

Name:	**multicastrate**
Description:	Defines the data rate to be used for all multicast/broadcast transmissions. The rate selection must be valid for the mode selected. The rates are defined via an index between 1 through 12.
	0 — 1 Mbps (Default)
	1-12 — Same as defined for *unicastrate*
Type:	UINT8
Default:	0
XML Format:	<param name="multicastrate" value="0"/>

Name:	**rtsthreshold**
Description:	Defines the minimum packet size in bytes required to trigger RTS/CTS for unicast transmissions.
	0 — Disable RTS/CTS
Type:	UINT16
Default:	0
XML Format:	<param name="rtsthreshold" value="0"/>

Name:	**enablertsctsack**	
Description:	Defines the mechanism for emulating point to point control messages such as RTS/CTS/ACK.	
	off	Disable
	on	Enable **(Not Supported)**
Type:	Boolean	
Default:	off	
XML Format:	<param name="enablertsctsack" value="off"/>	

Name:	**fragmentationthreshold**	
Description:	The model does not currently support packet fragmentation.	
	off	Disable fragmentation
	on	Enable fragmentation **(Not Supported)**
Type:	Boolean	
Default:	off	
XML Format:	<param name="fragmentation" value="off"/>	

Name:	**wmmenable**	
Description:	Provides the ability to enable the WiFi Multimedia (WMM) type feature. Current capability supports the ability to service packets from a higher priority queue first and does not yet support the internal contention based logic as defined by 802.11e.	
	off	Disable WMM
	on	Enable WMM
Type:	Boolean	
Default:	off	
XML Format:	<param name="wmmenable" value="off"/>	

Name:	**queuesize**
Description:	Defines the size of the queue for the given access category. The format for the value within XML is defined as follows: "A:B" , where A identifies the access category (0 to 4) and B defines the queue size (1 to 255) for access category A.
Type:	String
Default:	0:255 1:255 2:255 3:255
XML Format:	<param name="queuesize" value="0:255 1:255 2:255 3:55"/>

Name:	**cwmin**
Description:	Defines the minimum contention window in slots for the appropriate access category. The format for the value within XML is defined as follows: "A:C", where A identifies the access category (0 to 4) and C defines the minimum contention window for access category A.
Type:	String
Default:	0:31 1:31 2:15 3:7
XML Format:	<param name="cwmin" value="0:31 1:31 2:15 3:7"/>

Name:	**cwmax**
Description:	Defines the maximum contention window in slots for the appropriate access category. The format for the value within XML is defined as follows: "A:D", where A identifies the access category (0 to 4) and D defines the minimum contention window for access category A.
Type:	String
Default:	0:1023 1:1023 2:63 3:15
XML Format:	<param name="cwmax" value="0:1023 1:1023 2:63 3:15"/>

Name:	**aifs**
Description:	Defines the Arbitration Inter Frame Space (AIFS) time factor in slots for the appropriate access category. The inter frame space time (usec) is computed as follows: time = aifs*slotduration + sifs, where slot duration is a function of distance and sifs is a function of the 802.11 mode. The format for the value within XML is defined as follows: "A:E", where A identifies the access category (0 to 4) and E defines the AIFS factor value in slots for access category A.
Type:	String
Default:	0:2 1:2 2:2 3:1
XML Format:	<param name="aifs" value="0:2 1:2 2:2 3:1"/>

Name:	**txop**
Description:	Defines the maximum time (usec) a packet can reside within the queue for a given access category. Once the packet enters the MAC queue and is not serviced for this time, it will be discarded and not transmitted. Setting the value to 0 disables the feature and will service all packets regardless of duration in the queue. The format for the value within XML is defined as follows: "A:E", where A identifies the access category (0 to 4) and E defines the txop duration for access category A.
Type:	String
Default:	0:0 1:0 2:0 3:0
XML Format:	<param name="txop" value="0:0 1:0 2:0 3:0"/>

Name:	**retrylimit**
Description:	Defines the number of retries permitted for the unicast messages for the appropriate access category. The format for the value within XML is defined as follows: "A:F", where A identifies the access category (0 to 4) and F defines the number of retries for access category A.
Type:	String
Default:	0:3 1:3 2:3 3:3
XML Format:	<param name="retrylimit" value="0:3 1:3 2:3 3:3"/>

Name:	**maxmsdu**
Description:	Defines the maximum packet size (Bytes) permitted for the appropriate access category. The format for the value within XML is defined as follows: "A:G", where A identifies the access category (0 to 4) and G defines the packet size for access category A. Setting the value (G) to 0 disables the feature and allows all packet sizes
Type:	String
Default:	0:0 1:0 2:0 3:0
XML Format:	<param name="maxmsdu" value="0:0 1:0 2:0 3:0"/>

IEEE 802.11 abg PHY Parameters

Name:	**mode**
Description:	Defines the 802.11 mode. The mode is included in the PHY header of all over the air messages and is used to filter out packets received from nodes that are on different modes. Should be noted that mode setting does not care about frequency and is used only to identify the actual physical signal in space.
	0 — 802.11b (DSSS only)
	1 — 802.11b (DSSS only)
	2 — 802.11 a or g (OFDM)
	3 — 802.11 b/g (DSSS and OFDM)
	4-255 — Not Defined
Type:	UINT8
Default:	0
XML Format:	<param name="mode" value="0">

Name:	**frequency**
Description:	Defines the frequency in MHz for the 802.11 network. The frequency is included in the PHY header of all over the air messages and is used to filter out packets received from nodes that are on different frequencies. This will eventually be used to determine noise from out of band packets.
Type:	UINT16
Default:	2437
XML Format:	<param name="frequency" value="2437">

Name:	**bandwidth**
Description:	Defines the bandwidth in Khz. Bandwidth is included in the PHY header. The current version does not utilize this information. An upgrade to the 802.11abg model will use this information to determine the amount of band noise introduced by co-located networks operating within the same frequency space.
Type:	UINT16
Default:	20
XML Format:	<param name="bandwidth" value="20">

Name:	**systemnoisefloor**
Description:	Defines the system noise floor (dB) for a given radio which determines the radio's sensitivity. An entry of 95 represents sensitivity of -95dB for the radio. Upon the reception of a packet, the receive power of the received packet (based on path loss, txpower and tx/rx antenna gains) is compared to the noise floor. If the receive power is < noise floor, the packet is discarded.
Type:	UINT8
Default:	95
XML Format:	<param name="systemnoisefloor" value="95">

Name:	**carriersensethreshold**
Description:	Defines the System Interference Noise Ratio (SINR) value in dB where the PHY will notify the MAC of channel busy indication when the SINR from a given packet exceeds the threshold. The MAC uses this to hold off on local transmissions until the channel is free. Setting this to 0 (default) results in disabling this feature. **Current capability does not support carrier sense for out of band signaling.**
Type:	UINT8
Default:	0
XML Format:	<param name="carriersensethreshold" value="0">

Name:	**txpower**
Description:	Defines the transmit power (dBm) and is used to compute the receive power in conjunction with path loss and antenna gains. The txpower is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="txpower" value="0">

Name:	**antennatype**	
Description:	Defines if the antenna being utilized is omni or directional.	
	0	Omni
	1	Derectional **(Not currently supported)**
	2-255	Not Defined
Type:	UINT8	
Default:	0	
XML Format:	<param name="antennatype" value="0">	

Name:	**antennagain**
Description:	Defines the antenna gain (dBi) and is used to compute the receive power in conjunction with path loss and txpower. The antennagain is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="antennagain" value="0">

Name:	**antennabeamwidth**
Description:	Defines the antenna beam width (deg) when antennatype = Directional. Currently not used.
Type:	UINT8
Default:	0
XML Format:	<param name="antennabeamwidth" value="0">

Name:	**bercurves**
Description:	Identifies the XML file containing the BER curves for the 802.11 DSSS and OFDM rates. The bercurves XML file also contains the minimum packet size, maximum packet size and packet size interval parameters which are used at initialization to generate Probability of Reception (PoR) tables/curves. The PoR tables are a function of data rate, SINR and packet size and are generated using the following formula: PoR = (1 - BER)**L where L = number of bits in the packet and BER is the bit error rate as a fn(SINR, Data Rate) as provided via the bercurves XML file. **You must use a fully qualified URI (absolute file names).**
Type:	string
Default:	ieee80211bercurves.xml
XML Format:	<param name="bercurves" value="ieee80211bercurves.xml">"

Name:	**forcepass**
Description:	Provides the ability to bypass path loss and probability of reception checks. When forcepass is enabled, all received packets during idle (not busy Tx or Rx) that pass the mode and frequency checks are passed up to the MAC. off — Disable forcepass on — Enable forcepass
Type:	Boolean
Default:	off
XML Format:	<param name="forcepass" value="off">"

Name:	**defaultconnectivity**
Description:	Defines the default connectivity of all nodes at start up prior to receiving any path loss events. off — All nodes are assumed isolated on — All nodes are assumed to be fully informed
Type:	Boolean
Default:	on
XML Format:	<param name="defaultconnectivity" value="on">"

Interactive Commands

The interactive transport command shell supports the following commands:

clear	Clear Screen
exit	Exit telnet session. Same as *quit*.
ping	Transport to Transport ping.
quit	Quit telnet session. Same as *exit*.
stats	Show or clear statistic counters.
version	Print version.
whoami	Print NEM id.

ping Usage

The ping commands sends an interactive transport PING Request to another interactive transport to elicit an interactive transport PING Response.

ping [-i interval] [-c count] [-s size] NEMID|'all'

interval	Wait interval seconds between sending each packet. The default is to wait for one second between each packet.
count	Stop after sending count ping requests packets.
size	Specifies the number of data bytes to be sent. The default is 4, which translates into 64 data bytes when combined with the 60 bytes of interactive ping header.

stats Usage

The stats command displays or clears stats related to inbound and outbound interactive transport PING requests and responses.

stats [clear]

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="ieee80211abgnem.xml">
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <mac definition="ieee80211abgmac.xml">
      <!--
         * Set unicast data rate to 1Mbps
        -->
      <param name="unicastrate" value="1"/>
    </mac>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="ieee80211abgnem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <mac definition="ieee80211abgmac.xml">
      <!--
         * Set unicast data rate to 1Mbps
        -->
      <param name="unicastrate" value="1"/>
```

```
    </mac>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="ieee80211abgnem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <mac definition="ieee80211abgmac.xml">
      <!--
         * Set unicast data rate to 11Mbps
        -->
      <param name="unicastrate" value="4"/>
    </mac>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="ieee80211abgnem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <mac definition="ieee80211abgmac.xml">
      <!--
         * Set unicast data rate to 54Mbps
        -->
      <param name="unicastrate" value="12"/>
    </mac>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>
</platform>
```

Review the **ieee80211abgmac.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mac SYSTEM "file:///usr/share/emane/dtd/mac.dtd">
<mac name="IEEE 802.11 MAC" library="ieee80211abgmaclayer">
  <param name="enablepromiscuousmode" value="off"/>
  <param name="distance" value="1000"/>
  <param name="rtsthreshold" value="0"/>
  <param name="mode" value="3"/>
  <param name="unicastrate" value="0"/>
  <param name="multicastrate" value="0"/>
  <param name="wmmenable" value="off"/>
  <param name="queuesize" value="0:255"/>
  <param name="retrylimit" value="0:0"/>
  <param name="cwmin" value="0:31"/>
</mac>
```

Review the **ieee80211abgphy.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE phy SYSTEM "file:///usr/share/emane/dtd/phy.dtd">
<phy name="IEEE 802.11 PHY" library="ieee80211abgphylayer">
  <param name="mode" value="0"/>
  <param name="carriersensethreshold" value="0"/>
  <param name="frequency" value="2347"/>
  <param name="bandwidth" value="20"/>
  <param name="systemnoisefloor" value="95"/>
  <param name="txpower" value="0"/>
  <param name="antennagain" value="0"/>
  <param name="bercurves" value="file:///usr/share/emane/models/ieee80211abg/xml/
ieee80211bercurves.xml"/>
  <param name="forcepass" value="off"/>
  <param name="defaultconnectivity" value="off"/>
</phy>
```

Observations Explained

This workshop provides visibility into the physical layer component of the IEEE 802.11 abg model as it pertains to packet completion rate as a function of Signal to Noise (SINR), modulation type (data rate) and packet size. The workshop is configured such that all 4 nodes have the same SINR (8 dB) when

receiving a packet from any other node. SINR of 8 dB is achieved by using the default parameters for system noise floor (-95 dB), transmit power (0 dBm) and antenna gain (0 dBi) while setting the path loss to 87 dB (via events) between all nodes. In addition, each node is configured to transmit at the following rates:

- NEM 1: 1 Mbps
- NEM 2: 1 Mbps
- NEM 3: 11 Mbps
- NEM 4: 54 Mbps

As NEM 1 pings NEMs 2 through 4, the return ping received at NEM 1 will be processed based on Bit Error Rate curves associated with the respective modulation type (BPSK, QPSK and 64-QAM). The table below shows the expected completion rates from NEM 1 to all other nodes for packet size of 128Bytes and 1024Bytes.

SINR = 8 dB			
	Completion Rate (%)		
Packet Size (Bytes)	NEM 2 (1 Mbps)	NEM 3 (11 Mbps)	NEM 4 (54 Mbps)
128	100	49	0
1024	100	0	0

The second part of the workshop requires the adjustment of the antenna gain to 13 dBi on NEM 1 which in essence raises the SINR from 8 dB to 20 dB for all packets received by NEM 1. Table below shows the expected completion rates with the higher SINR for the different rates.

SINR = 20 dB			
	Completion Rate (%)		
Packet Size (Bytes)	NEM 2 (1 Mbps)	NEM 3 (11 Mbps)	NEM 4 (54 Mbps)
128	100	100	70
1024	100	100	7

Concept Review

1. Why is timing so important for the IEEE 802.11 abg model?

   Timing is utilized to accurately estimate message receive period to emulate in-band collisions.

2. How did the path loss between all the NEMs get set to 87 dB?

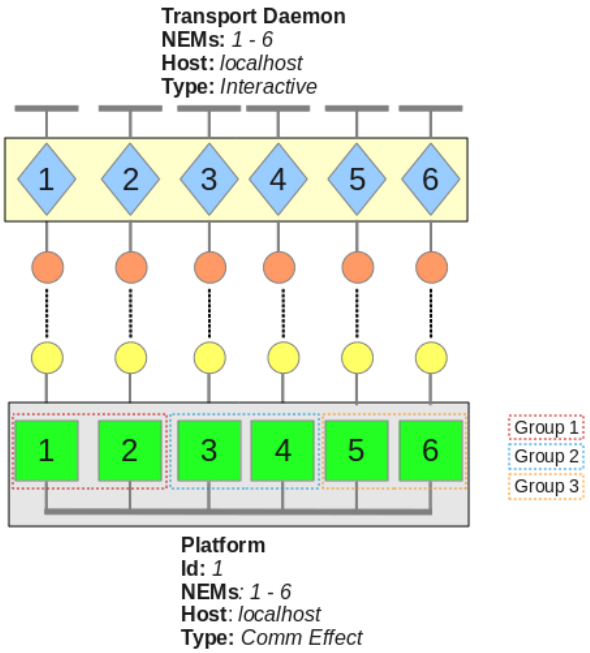
   Looking at the contents of the **start_workshop.sh** script will reveal a call to the **sendpathloss.pl** perl script. This script uses the libemaneeventservice perl language bindings to send path loss events.

Workshop 09 - Comm Effect Model Model

The goal of this workshop is to become familiar with configuring the Comm Effect Model.

Deployment Description

This workshop contains a centralized Comm Effect Deployment using Interactive Transports. The single NEM Platform Server is hosting 6 NEMs.



Problem Statement

The `platform.xml` file in this workshop contains 6 NEMs with incomplete configuration. Modify the NEM Platform Server XML configuration to achieve the following:

- Place NEM 1 and NEM 2 into Comm Effect group 1
- Place NEM 3 and NEM 4 into Comm Effect group 2
- Place NEM 5 and NEM 6 into Comm Effect group 3

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify the settings using the Interactive Transport to unicast and broadcast *ping* between NEMs.

Comm Effect Shim Parameters

Name:	**defaultconnectivity:**	
Description:	Defines the default communication effects of all NEMs at start up prior to receiving any commeffect events. All filter rules, if any, are still processed regardless of whether *defaultconnectivity* is in use.	
	on	All NEMS are allowed to communication without first receiving commeffect event data.
	off	No NEMS are allowed to communication without first receiving comm effect event data that does not match any filter rules.
Type:	Boolean	
Default:	on	
XML Format:	<param name="defaultconnectivity" value="on"/>	

Name:	**filterfile:**	
Description:	The name of the CommEffect filter file to load. **You must use a fully qualified URI (absolute file names).**	
Type:	String	
Default:	None	
XML Format:	<param name="filterfile" value="file:///etc/emane/filterfile.xml"/>	

Name:	**groupid:**	
Description:	Defines the NEM Group ID which will be used to group NEMs by the assigned ID value. When an NEM is assigned to a group it can only receive traffic from other members of the same group regardless of communication effects.	
	Equal to 0	Not associated with a NEM Group
	Greater than 0	Associated with that NEM Group
Type:	UINT32	
Default:	none	
XML Format:	<param name="groupid" value="0"/>	

Name:	**enablepromiscuousmode:**	
Description:	Determines if all packets received will be sent upstream to the transport	
	on	Send only broadcast and locally addressed unicast up.
	off	Send all packets up
Type:	Boolean	
Default:	off	
XML Format:	<param name="enablepromiscuousmode" value="off"/>	

Name:	**enabletighttimingmode:**
Description:	Determines whether transmission time of the packet will be factored in when calculating delivery scheduling time. **When enabled tight time sync will be required for all platforms in the EMANE deployment.** on — Factor in transmission time of the packet when calculating delivery scheduling time. off — Ignore transmission time of the packet when calculating delivery scheduling time
Type:	Boolean
Default:	off
XML Format:	<param name="enabletighttimingmode" value="off"/>

Name:	**receivebufferperiod:**
Description:	Speicify the max sum of buffering time in seconds for packets received from an NEM. The buffering interval for each packet is determined by the bitrate for the src NEM and packet size. Packets are then placed in a timed queue based on this interval and any packets that would cause the receive buffer period to be exceeded are discarded. A value of 0.0 disables the limit and allows all received packets to stack up in the queue.
Type:	Float
Default:	1.0
XML Format:	<param name="receivebufferperiod" value="1.0"/>

Interactive Commands

The interactive transport command shell supports the following commands:

clear	Clear Screen
exit	Exit telnet session. Same as *quit*.
ping	Transport to Transport ping.
quit	Quit telnet session. Same as *exit*.
stats	Show or clear statistic counters.
version	Print version.
whoami	Print NEM id.

ping Usage

The ping commands sends an interactive transport PING Request to another interactive transport to elicit an interactive transport PING Response.

ping [-i interval] [-c count] [-s size] NEMID|'all'

interval	Wait interval seconds between sending each packet. The default is to wait for one second between each packet.
count	Stop after sending count ping requests packets.
size	Specifies the number of data bytes to be sent. The default is 4, which translates into 64 data bytes when combined with the 60 bytes of interactive ping header.

stats Usage

The stats command displays or clears stats related to inbound and outbound interactive transport PING requests and responses.

stats [clear]

XML Listing

Modify the **platform.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="1" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="1" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="2" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="2" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>
```

```
  <nem name="NODE-005" id="5" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8105"/>
    <param name="transportendpoint" value="localhost:8205"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="3" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-006" id="6" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8106"/>
    <param name="transportendpoint" value="localhost:8206"/>
    <shim definition="commeffectshim.xml">
      <!--
         Set the shim definition file
         Set the shim parameter groupid
        -->
      <param name="groupid" value="3" />
    </shim>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

</platform>
```

Review the **commeffectnem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="COMMEFFECT NEM" type="unstructured">
  <shim definition="commeffectshim.xml"/>
  <transport definition="transinteractive.xml"/>
</nem>
```

Concept Review

1. Is the Comm Effect Model a radio model?

   No. The Comm Effect Model allows you to add various effects to individual links between NEMs. There is no notion of a single set of characteristics for a Comm Effect NEM.

2. What are filters?

   Filters are packet matching rules that allow you to effect certain types of traffic differently. For more information see the Comm Effect Manual.

3. What is an *unstructured* NEM?

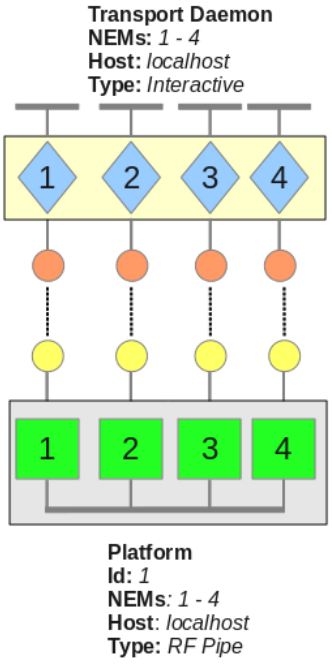
   An *unstructured* NEM may contain any number of layers in any order. Unstructured NEMs provide EMANE component developers the ability to circumvent the internal builder logic used when creating an NEM. The only rule is that an unstructured NEM may contain only a single MAC and a single PHY.

Workshop 10 - Mitre Mobility Event Generator

The goal of this workshop is to become familiar with the Mitre Mobility Event Generator.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs. An Event Service hosting the Mitre Mobility Event Generator is used to publish pathloss and location events.



Problem Statement

The `mitremobilitygenerator.xml` file contained in this workshop is missing certain configuration. Modify the generator XML configuration to achieve the following:

- Use `/opt/mobilitymodel/path_loss_matrix_00.txt` as the single mobility input file
- Set the generator to loop continuously

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify that the EMANE Event Service is publishing events using the **locationviewer.py** and **pathlossviewer.pl** scripts.

Mitre Mobility Generator Parameters

Name:	**inputfileformat**
Description:	Full filename including path of mobility file. One or more files may be specified buy using a `printf()` style convention using the *inputfilecount* configuration item as an index.
Type:	Filename
XML Format:	<param name="inputfileformat" value="path_loss_matrix_%02d.txt"/>

Name:	**inputfilecount**
Description:	Total number of input files. If the *inputfilename* contains a `printf()` style expression the count will be used as an index when creating the file names.
Type:	UINT32
XML Format:	<param name="inputfilecount" value="1"/>

Name:	**totalnodes**
Description:	Total number of nodes whose data is contained in the mobility files.
Type:	UINT32
XML Format:	<param name="totalnodes" value="70"/>

Name:	**maxnemidpresent**
Description:	Maximum NEM Id present in test. This value is used to reduce the number of events generated when a subset of nodes from the larger mobility data are used.
Type:	UINT16
XML Format:	<param name="maxnemidpresent" value="12"/>

Name:	**repeatcount**
Description:	The number of times the mobility data should be parsed and events generated. A *repeatcount* of 1 simply means to process the file once, generating events, and stop when the file is complete. A value greater than one allows you to process the data *repeatcount* times. A value of zero will process the data repeatedly, restarting indefinitely.
Type:	UINT32
XML Format:	<param name="repeatcount" value="1"/>

Name:	**utmzone**
Description:	UTM zone that corresponds to the UTM position information contained in the mobility data. This is required to convert the data when generating location events.
Limitation:	Position data cannot cross zones.
Type:	String
XML Format:	<param name="utmzone" value="18T"/>

Name:	**entryreplay**
Description:	Specify one or more space separated *time:count* pairs. Where time corresponds to the second of data contained in the mobility model and count represents the number of times (seconds) you want to replay that data. This effectively allows you hold at certain mobility entries for a specified amount of time. For example a value of "0:120 3600:1800" would use the data at mobility entry T0 for 2 minutes, sending out the T0 update 120 times and use the data at T3600 for 30 minutes.
Type:	String
XML Format:	<param name="entryreplay" value="0:120 3600:1800"/>

Name:	**publishlocationevents**
Description:	Create/Publish location events from mobility model input files.
Type:	Boolean
XML Format:	<param name="publishlocationevents" value="on"/>

Name:	**publishpathlossevents**
Description:	Create/Publish pathloss events from mobility model input files.
Type:	Boolean
XML Format:	<param name="publishpathlossevents" value="on"/>

XML Listing

Modify the **mitremobilitygenerator.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE eventgenerator SYSTEM "file:///usr/share/emane/dtd/eventgenerator.dtd">
 <eventgenerator name="Mitre Mobility Generator" library="mitremobilitygenerator">
   <!--
      Set the mobility file input
     -->
   <param name="inputfileformat" value="/opt/mobilitymodel/path_loss_matrix_00.txt"/>
   <param name="inputfilecount" value="1"/>
   <param name="totalnodes" value="70"/>
   <param name="maxnemidpresent" value="40"/>

   <!--
       Set the repeat count to loop continuously
     -->
   <param name="repeatcount" value="0"/>
   <param name="utmzone" value="18T"/>

</eventgenerator>
```

Review the **eventservice.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventservice SYSTEM "file:///usr/share/emane/dtd/eventservice.dtd">
<eventservice name="Sample Event Service" deployment="deployment.xml">
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>
  <generator name="Mobility Model Generator" definition="mitremobilitygenerator.xml"/>
</eventservice>
```

Concept Review

1. How do you set the overall start time for the EMANE Event Service?

   Use the **emaneeventservice** --*starttime* option.

2. Why is the UTM Zone a configuration parameter?

   The Mitre Mobility Model format does not include UTM zone. This is an issue if your scenario crosses zones. EMANE Location events are made up of GPS coordinates. Zone information is necessary to convert UTM to latitude and longitude.

3. What is the purpose of the *entryreplay* configuration parameter?

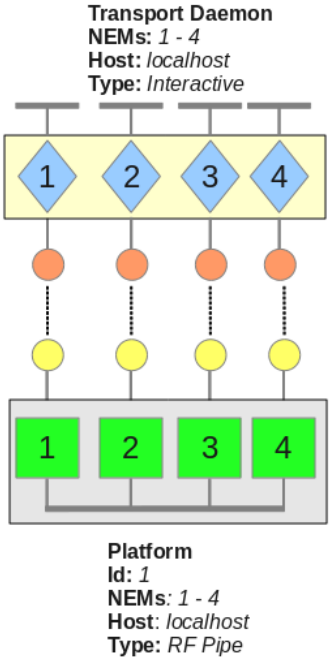
   The *entryreplay* parameter provides a means to replay one or more specific time indexes for a specified amount of time.

Workshop 11 - Emulation Script Generator

The goal of this workshop is to become familiar with the Emulation Script Event Generator.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs. An Event Service hosting the Emulation Script Generator is used to publish location events.



Problem Statement

The `emulationscriptgenerator.xml` file contained in this workshop is missing certain configuration. Modify the XML configuration to achieve the following:

Use the following input files:

- `/opt/locationscript/location_00.xml`
- `/opt/locationscript/location_01.xml`
- `/opt/locationscript/location_02.xml`
- `/opt/locationscript/location_03.xml`
- Set the generator to loop continuously
- Configure the RF Pipe PHY layer to use location events to calculate path loss in realtime.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify that the EMANE Event Service is publishing events using the **locationviewer.py** script.

Emulation Script Generator Parameters

Name:	**inputfile**
Description:	Full filename including path of mobility file. One or more files may be specified by using this parameter multiple times modifying the value attribute accordingly.
Type:	Filename
XML Format:	<param name="inputfile" value="location_00.xml"/>

Name:	**repeatcount**
Description:	The number of times the mobility data should be parsed and events generated. A *repeatcount* of 1 simply means to process the file once, generating events, and stop when the file is complete. A value greater than one allows you to process the data *repeatcount* times. A value of zero will process the data repeatedly, restarting indefinitely.
Type:	UINT32
XML Format:	<param name="repeatcount" value="1"/>

Name:	**schemalocation**
Description:	Specifies the location of the schema file used to validate files specified via *inputfile* parameters.
Type:	URI
XML Format:	<param name="schemalocation" value="http://configserver/schema/EmulationScriptSchema.xsd"/>

RF Pipe PHY Parameters

Name:	**frequency**
Description:	Defines the frequency used in MHz for the emulated network. The frequency is included in the PHY header of all over the air messages and serves 2 functions: 1. Filters out packets received from nodes that are on different frequencies. 2. Is used to determine free space path loss when configured to use compute path loss based on node positions using the free space propagation algorithm. See configuration parameters *pathlosssource* and *pathlossalgorithm* below.
Type:	UINT16
Default:	2437
XML Format:	<param name="frequency" value="2437">

Name:	**bandwidth**
Description:	Defines the bandwidth in Khz. Bandwidth is included in the PHY header. The current version does not utilize this information.
Type:	UINT16
Default:	20
XML Format:	<param name="bandwidth" value="20">

Name:	**txpower**
Description:	Defines the transmit the power (dBm) and is used to compute the receive power in conjunction with path loss and antenna gains. The txpower is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="txpower" value="0">

Name:	**antennagain**
Description:	Defines the antenna gain (dBi) and is used to compute the receive power in conjunction with path loss and txpower. The antennagain is included in the PHY header of all over the air messages.
Type:	UINT8
Default:	0
XML Format:	<param name="antennagain" value="0">

Name:	**bitrate**
Description:	Defines the bitrate/burstrate (kbps) of the waveform being emulated. It is used on the transmit side to compute transmission delay based on the packet size and bitrate.
Type:	UINT16
Default:	0
XML Format:	<param name="bitrate" value="0">

Name:	**propagationdelay**
Description:	Defines the propagation delay (usec) that is to be included in the transmission delay. The propagationdelay is added to the delay introduced by the bitrate defined above.
Type:	UINT32
Default:	0
XML Format:	<param name="propagationdelay" value="0">

Name:	**jitter**
Description:	Defines the jitter (usec) to be included to the transmission delay. The jitter will be computed for each packet transmission based on uniform random distribution between +- the configured jitter value.
Type:	UINT32
Default:	0
XML Format:	<param name="jitter" value="0">

Name:	**pathlosssource**
Description:	Defines the method of use for determining path loss. The parameter accepts 2 options: 0 — Pathloss directly provided via an external event 1 — Pathloss computed real time based on external position events and selected *pathlossalgorithm*
Type:	UINT16
Default:	0
XML Format:	<param name="pathlosssource" value="0">

Name:	**pathlossalgorithm**
Description:	Defines the algorithm to be used to compute the path loss. This is valid only when *pathlosssource* is set to *1*. The algorithm selections are as follows: 0 — Free space path loss model 1 — 2 ray flat earth model
Type:	UINT16
Default:	0
XML Format:	<param name="pathlossalgorithm" value="0">

Name:	**rxpowernolossthreshold**
Description:	Defines the receive power threshold (dBm) for determining if packets will be received or dropped. This threshold is used in conjunction with the *rxpowerfulllossthreshold* to determine the probability of packet reception (PoR) based on received power as follows: Rx Power >= rxpowernolossthreshold ==> PoR = 1 rxpowerfulllossthreshold < Rx Power < rxpowernolossthreshold ==> linear packet loss Rx Power <= rxpowerfulllossthreshold ==> PoR = 0
Type:	UINT16
Default:	90
XML Format:	<param name="rxpowernolossthreshold" value="90">

Name:	**rxpowerfulllossthreshold**
Description:	See description of *rxpowernolossthreshold* above.
Type:	UINT16
Default:	110
XML Format:	<param name="rxpowerfulllossthreshold" value="110">

Name:	**defaultconnectivity**
Description:	Defines the default connectivity of all nodes at start up prior to receiving any mobility or path loss events. When set to *on* the reception of a single path loss or position event will disable the default connectivity setting.
	off — All nodes are assumed isolated
	on — All nodes are assumed to be fully informed
Type:	Boolean
Default:	on
XML Format:	<param name="defaultconnectivity" value="on">

XML Listing

Modify the **emulationscriptgenerator.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventgenerator SYSTEM "file:///usr/share/emane/dtd/eventgenerator.dtd">
<eventgenerator name="Emulation Script Generator" library="emulationscriptgenerator">

   <!--
       Add the mobility file input
     -->
   <param name="inputfile" value="/opt/locationscript/location_00.xml"/>
   <param name="inputfile" value="/opt/locationscript/location_01.xml"/>
   <param name="inputfile" value="/opt/locationscript/location_02.xml"/>
   <param name="inputfile" value="/opt/locationscript/location_03.xml"/>

   <!--
       Set the repeat count to play continuously
     -->
   <param name="repeatcount" value="0"/>

   <param name="schemalocation"
     value="file:///usr/share/doc/emane-gen-emulationscript-0.6.2/EmulationScriptSchema.xsd"/
>
</eventgenerator>
```

Modify the **rfpipephy.xml** file to achieve the workshop goals.

```
<!DOCTYPE phy SYSTEM "file:///usr/share/emane/dtd/phy.dtd">
<phy name="RF-PIPE PHY" library="rfpipephylayer">
  <param name="txpower" value="0"/>
  <param name="antennagain" value="0"/>
  <param name="bitrate" value="0"/>
  <param name="jitter" value="0"/>
  <param name="frequency" value="2347"/>
  <param name="bandwidth" value="20"/>
  <param name="propagationdelay" value="0"/>
  <!--
     Set the pathloss source
    -->
  <param name="pathlosssource" value="1"/>
  <param name="pathlossalgorithm" value="0"/>
  <param name="defaultconnectivity" value="on"/>
  <param name="rxpowernolossthreshold" value="90"/>
  <param name="rxpowerfulllossthreshold" value="110"/>
</phy>
```

Review the **eventservice.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventservice SYSTEM "file:///usr/share/emane/dtd/eventservice.dtd">
<eventservice name="Sample Event Service" deployment="deployment.xml">
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>
  <generator name="Emulation Script Generator" definition="emulationscriptgenerator.xml"/>
```

```
</eventservice>
```

Concept Review

1. Why was it necessary to change the RF Pipe PHY Layer configuration?

   The RF Pipe must be configured to accept Location Events and calculate path loss based on distance.

2. Can the EMANE Event Service be configured to host multiple numbers of generators at the same time?

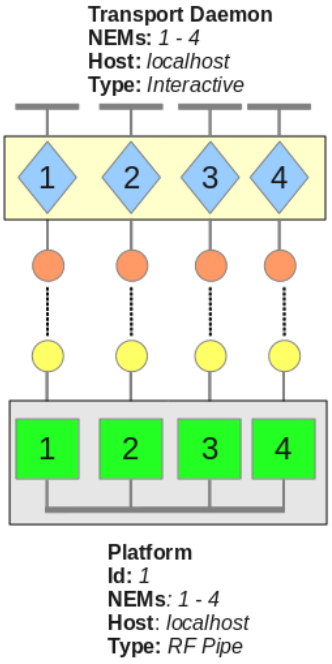
   Yes. In a manner similar to the way an NEM Platform Server can instantiate and manage various numbers of NEMs the Event Service can create and manage a various number of generators.

Workshop 12 - Emulation Event Log Generator

The goal of this workshop is to become familiar with the Emulation Event Log Event Generator.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs. An Event Service hosting the Emulation Event Log Event Generator is used to publish pathloss and location events.



Problem Statement

The `eelgenerator.xml` file contained in this workshop is missing certain configuration. Modify the generator XML configuration to achieve the following:

- Use `/opt/eelmodel/mobility.eel` as the single event input file.
- Configure a loader for *location* event type entries using the *eelloaderlocation* plugin. Use *full* updates.
- Configure a loader for *pathloss* event type entries using the *eelloaderpathloss* plugin. Use *full* updates.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify that the EMANE Event Service is publishing events using the **locationviewer.py** script.

Emulation Event Log Generator Parameters

Name:	**inputfile**
Description:	Full filename including path of EEL file. Additional EEL files may be specified using multiple *inputfile* parameters. Files are processed in the order they appear in the XML.
Multiplicity:	1:N
Type:	Filename
XML Format:	<param name="inputfile" value="/opt/mobility.eel"/>

Name:	**loader**
Description:	Map EEL event type keywords to EEL loader plugins. The optional **full** or **delta** specification determines whether events produced from the plugins represent only the new EEL entries processed since the last request for events or the complete current cached state. The default specification is **delta**.
Syntax:	<eventType>:<plugin>[:full|delta]
Multiplicity:	0:N
Type:	string
XML Format:	<param name="loader" value="pathloss:eelloaderpathloss:full"/>

XML Listing

Modify the **eelgenerator.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventgenerator SYSTEM "file:///usr/share/emane/dtd/eventgenerator.dtd">
<eventgenerator name="EEL Generator" library="eelgenerator">
  <!--
     * Add the mobility file input
    -->
  <param name="inputfile" value="/opt/eelmodel/mobility.eel"/>
  <!--
     * Add loader parameter for location entries
    -->
  <param name="loader" value="location:eelloaderlocation:full"/>
  <!--
     * Add loader parameter for pathloss entries
    -->
  <param name="loader" value="pathloss:eelloaderpathloss:full"/>
</eventgenerator>
```

Review the **eventservice.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventservice SYSTEM "file:///usr/share/emane/dtd/eventservice.dtd">
<eventservice name="Sample Event Service" deployment="./deployment.xml">
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>
  <generator name="EEL Generator" definition="eelgenerator.xml"/>
</eventservice>
```

Concept Review

1. What is the difference between the *full* and *delta* publish modes?

   In *full* mode, when a plugin is queried for event information, it returns all the cached information it is currently holding. In *delta* mode, the plugin only returns event information for those items that have changed since the last query.

2. Give an example of when you might want to use the *full* publish mode?

   In a deployment where test nodes might be going up and down, *full* mode allows a node to immediately sync with the current emulation state on restart.

3. How is an EEL loader associated with an event type keyword?

   The first part of the *loader* configuration parameter is the name of the event type keyword to associate with the EEL loader plugin. The plugin to load is the second part of the parameter.

   <eventType>:<plugin>[:full|delta]

Workshop 13 - GPSd Location Agent

The goal of this workshop is to become familiar with the GPSd Location Agent.

Deployment Description

This workshop contains an Event Service hosting the Mitre Mobility Event Generator publishing location events. The Event Daemon is hosting the GPSd Location Agent.

Problem Statement

The `gpsdlocationagent.xml` file contained in this workshop is missing certain configuration. Modify the agent XML configuration to achieve the following:

- Disable automatic connection to gpsd.
- Use `/tmp/gpsdlocation.pty` as the pty name file.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify location event translation using the GPSd client applications **gpsmon** and **cgps**.

GPSd Location Agent Parameters

Name:	**gpsdcontrolsocket**
Description:	The name of the GPSd control socket for adding the pseudo terminal to the device list.
Type:	Filename
Default:	/tmp/gpsd.control
XML Format:	<param name="gpsdcontrolsocket" value="/tmp/gpsd.control"/>

Name:	**pseudoterminalfile**
Description:	The name of the file to create which will contain the name of the pseudo terminal in use by the gpsdlocation agent. Only created when *gpsdconnectionenabled* set to *off*.
Type:	Filename
Default:	/tmp/gpsdlocation.pty
XML Format:	<param name="pseudoterminalfile" value="/tmp/gpsdlocation.pty"/>

Name:	**gpsdconnectionenabled**
Description:	Switch to set gpsdlocation agent to either actively connect to GPSd (*on*) or instead create a file containing the name of the pseudo terminal currently in use (*off*).
Type:	Boolean
Default:	off
XML Format:	<param name="gpsdconnectionenabled" value="off"/>

XML Listing

Modify the **gpsdlocationagent.xml** file to achieve the workshop goals.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE eventagent SYSTEM "file:///usr/share/emane/dtd/eventagent.dtd">
<eventagent name="gpsdlocationagent" library="gpsdlocationagent">
  <!--
     Disable automatic connection to gpsd
    -->
  <param name="gpsdconnectionenabled" value="off"/>

  <!--
     Set the pseudo terminal file filename
    -->
  <param name="pseudoterminalfile" value="/tmp/gpsdlocation.pty"/>

  <param name="gpsdcontrolsocket" value="/tmp/gpsd.control"/>
</eventagent>
```

Review the **eventdaemon.xml.in** file. *Note the @ID@ variable is a replaced with your Training Id when the workshop is built.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eventdaemon SYSTEM "file:///usr/share/emane/dtd/eventdaemon.dtd">
<eventdaemon name="EMANE Event Daemon" nemid="@ID@">
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="eth1"/>
  <agent definition="gpsdlocationagent.xml"/>
</eventdaemon>
```

Concept Review

1. What is the purpose of the *pseudoterminalfile* configuration parameter?

   The *pseudoterminalfile* parameter specifies the name of the file to create to hold the name of the pseudo terminal created by the agent.

2. Does this agent only work with gpsd?

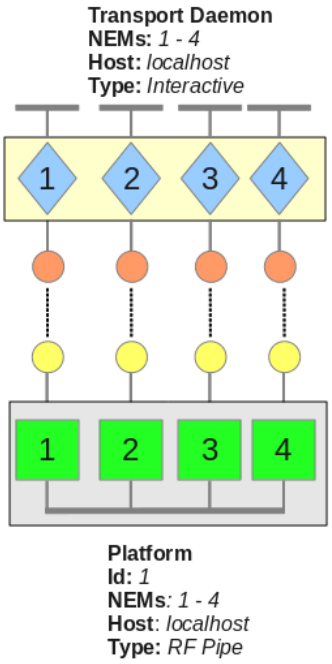
   No. Any application that can be configured to read from a TTY and can parse NMEA strings will work.

Workshop 14 - EMANE Loss Controller

The goal of this workshop is to become familiar with the EMANE Loss Controller application.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 4 NEMs.



Problem Statement

Using the **emanelosscontroller** to create and execute a 4 node loss scenario. Create a scenario to achieve the following:

- Use *lo* as the event device.
- Set the no loss receive threshold to *90*.
- Set the full loss receive threshold to *110*.
- Use the controller to publish pathloss events to modify link conditions.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify the executing controller scenario by using the Interactive Transport to unicast and broadcast *ping* between NEMs.

XML Listing

Review the **platform.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="rfpipenem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>
</platform>
```

Concept Review

1. What are the file components of a loss controller project?

   The Loss Controller saves all loss information in a format usable by the Mitre Mobility Event Generator. The controller also generates all the required XML to run the scenario with the Event Service.

   A Loss Controller project contains the following files:

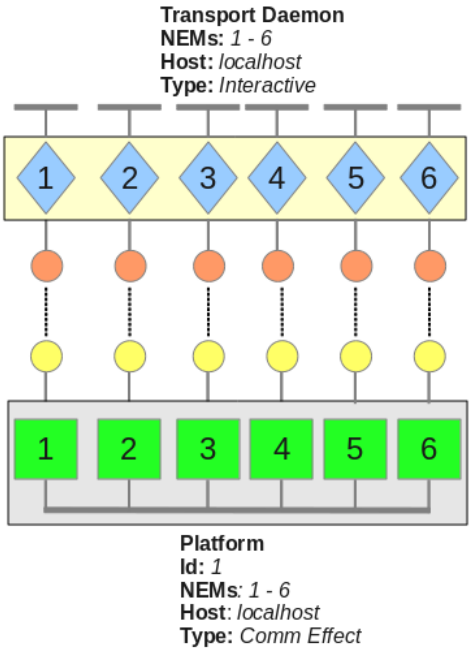
<PROJECTNAME>.xml	Project settings file
<PROJECTNAME>_es.xml	Event Service XML configuration
<PROJECTNAME>.txt	Pathloss model matrix
<PROJECTNAME>.xml	Mitre Mobility Model Generator XML configuration
<PROJECTNAME>_<NEMID>.xml	Filter definitions (if defined)

Workshop 15 - EMANE Comm Effect Controller

The goal of this workshop is to become familiar with the EMANE Comm Effect Controller application.

Deployment Description

This workshop contains a centralized Comm Effect Deployment using Interactive Transports. The single NEM Platform Server is hosting 6 NEMs.



Problem Statement

Using the **emanecommeffectcontroller** to create and execute a 6 node loss scenario. Create a scenario to achieve the following:

- Use *lo* as the event device.
- Use the controller to publish comm effect events to modify link conditions.

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify the executing controller scenario by using the Interactive Transport to unicast and broadcast *ping* between NEMs.

XML Listing

Review the **platform.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE platform SYSTEM "file:///usr/share/emane/dtd/platform.dtd">
<platform name="Platform 1" id="1">
  <param name="otamanagerchannelenable" value="off"/>
  <param name="eventservicegroup" value="224.1.2.8:45703"/>
  <param name="eventservicedevice" value="lo"/>

  <nem name="NODE-001" id="1" definition="commeffectnem.xml">
```

```
    <param name="platformendpoint" value="localhost:8101"/>
    <param name="transportendpoint" value="localhost:8201"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-002" id="2" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8102"/>
    <param name="transportendpoint" value="localhost:8202"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-003" id="3" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8103"/>
    <param name="transportendpoint" value="localhost:8203"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-004" id="4" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8104"/>
    <param name="transportendpoint" value="localhost:8204"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-005" id="5" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8105"/>
    <param name="transportendpoint" value="localhost:8205"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

  <nem name="NODE-006" id="6" definition="commeffectnem.xml">
    <param name="platformendpoint" value="localhost:8106"/>
    <param name="transportendpoint" value="localhost:8206"/>
    <transport definition="transinteractive.xml" group="1"/>
  </nem>

</platform>
```

Concept Review

1. What are the file components of a Comm Effect Controller project?

   The Comm Effect Controller saves all effect information in a format usable by the Comm Effect Event Generator. The controller also generates all the required XML to run the scenario with the Event Service.

   A Comm Effect Controller project contains the following files:

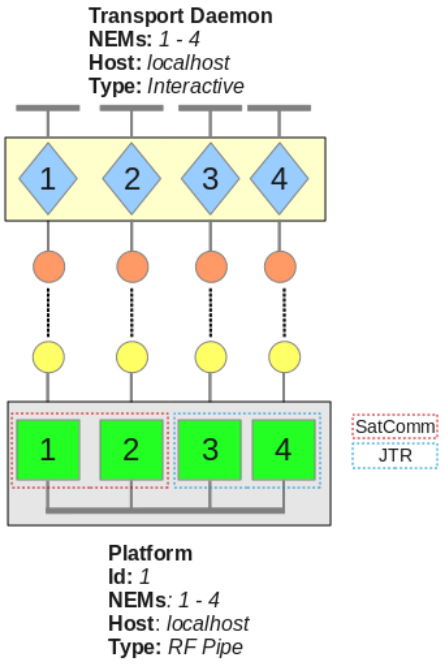
<PROJECTNAME>.xml	Project settings file
<PROJECTNAME>_es.xml	Event Service XML configuration
<PROJECTNAME>.txt	Comm Effect model matrix
<PROJECTNAME>.xml	Comm Effect Generator XML configuration
<PROJECTNAME>_<NEMID>.xml	Filter definitions (if defined)

Workshop 16 - Surrogate Radio Models using RF Pipe

The goal of this workshop is to become familiar with using the RF Pipe Model to create surrogate radio definitions.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Interactive Transports. The single NEM Platform Server is hosting 2 NEMs. Two NEMs are surrogate SatComm radios and two NEMs are surrogate JTR radios.



Problem Statement

The `satcommnem.xml` file and the `jtrnem.xml` file are surrogate radio definitions. Use interactive sessions to verify the surrogate model performance.

- NEM 1 and NEM 2 are SatComms
- NEM 3 and NEM 4 are JTR radios

Use the **start_workshop.sh** and **stop_workshop.sh** scripts to *start* and *stop* the workshop EMANE components.

Verify the following using the Interactive Transport sessions:

1. Node 1 ping to Node 2 with 1000 byte packets results in ~360 msec delay (80 msec for bitrate and 100 msec for propagation delay).

2. Node 1 can only send ~ 12 1000 byte packets per second and maintain the 360 msec latency. Once you exceed 12 packets per second the delay will start to increase.

3. Node 3 ping to Node 4 with 1000 byte packets results in ~56 msec delay (8 msec for the bitrate and 20 msec for propagation delay).

4. Node 3 can send about 125 packets per second before the latency starts to grow.

XML Listing

Review the **satcommnem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="SatComm NEM">
  <mac definition="rfpipemac.xml"/>
  <phy definition="rfpipephy.xml">
    <param name="frequency" value="2100"/>
    <param name="bitrate" value="100"/>
    <param name="propagationdelay" value="100000"/>
    <param name="jitter" value="0"/>
  </phy>
  <transport definition="transinteractive.xml"/>
</nem>
```

Review the **jtrnem.xml** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nem SYSTEM "file:///usr/share/emane/dtd/nem.dtd">
<nem name="JTR NEM">
  <mac definition="rfpipemac.xml"/>
  <phy definition="rfpipephy.xml">
    <param name="frequency" value="2500"/>
    <param name="bitrate" value="1000"/>
    <param name="propagationdelay" value="20000"/>
    <param name="jitter" value="0"/>
  </phy>
  <transport definition="transinteractive.xml"/>
</nem>
```

Workshop 17 - libemaneeventservice

The goal of this workshop is to become familiar with libemaneeventservice.

Description

libemaneeventservice is a C library that exposes a publisher/subscriber API for the EMANE event service.

The following man pages contain detailed API information:

1. emaneEventService (3)
2. emaneEventPathloss (3)
3. emaneEventLocation (3)
4. emaneEventCommEffect (3)

Publisher Sample Application

Review the **publisher** sample application which creates and publishes pathloss, location, and comm effect events.

usage: publisher [-p] [-c] [-l] [-n msgcount] [-x nodecount]

-p	Generate Pathloss events
-c	Generate CommEffect events
-l	Generate Location events
-n msgcount	Send *msgcount* events
-x nodecount	Set the number of nodes in the deployment

Subscriber Sample Application

Review the **subscriber** sample application which subscribes to pathloss, location, and comm effect events and prints a summary of received events.

usage: subscriber [-q]

-q	Quiet mode. Do not print

libemaneeventservice Configuration

Configuration for the event service is searched for in three locations prior to using default configuration

1. Environment Variable LIBEMANEEVENTSERVICECONFIG

   If LIBEMANEEVENTSERVICECONFIG exists it will be used. This variable may be set to a configuration file name.

2. $HOME/.libemaneeventservice.xml

   If .libemaneeventservice.xml exists it will be used.

3. /etc/libemaneeventservice.xml

If /etc/libemaneeventservice.xml exists it will be used.

4. Library default values

   The default values are: group *224.1.2.8*, port *45703*, multicast loop enabled *1*, TTL *32*. **No default multicast device is specified. The kernel routing table is used.**

Configuration Format

```
<?xml version='1.0' standalone='yes'?>
<emaneeventmsgsvc>
  <group>224.1.2.8</group>
  <port>45703</port>
  <device>lo</device>
  <mcloop>1</mcloop>
  <ttl>32</ttl>
</emaneeventmsgsvc>
```

Workshop 18 - libemaneeventservice Perl language bindings

The goal of this workshop is to become familiar with libemaneeventservice Perl language bindings.

Description

libemaneeventservice is a C library that exposes a publisher/subscriber API for the EMANE event service.

The following man pages contain detailed API information:

1. EMANE::EventService (3)
2. EMANE::Event::Pathloss (3)
3. EMANE::Event::Location (3)
4. EMANE::Event::CommEffect (3)

Sample Scripts

Review the following sample scripts:

ex01_publisher.pl	Simple publisher example
ex02_subscriber.pl	Simple subscriber example
ex03_subscriber_default.pl	Simple subscriber example using default callback
ex04_publisher_pathloss.pl	Simple pathloss publisher
ex05_subscriber_pathloss.pl	Simple pathloss subscriber that prints out event info
ex06_publisher_location.pl	Simple location publisher
ex07_subscriber_location.pl	Simple location subscriber that prints out event info
ex08_publisher_commeffect.pl	Simple comm effect publisher
ex09_subscriber_commeffect.pl	Simple comm effect subscriber that prints out event info

Workshop 19 - libemaneeventservice Python language bindings

The goal of this workshop is to become familiar with libemaneeventservice Python language bindings.

Description

libemaneeventservice is a C library that exposes a publisher/subscriber API for the EMANE event service.

The following pydoc contain detailed API information:

1. emaneeventservice
2. emaneeventservice.EventService
3. emaneeventpathloss
4. emaneeventpathloss.EventPathloss
5. emaneeventlocation
6. emaneeventlocation.EventLocation
7. emaneeventcommeffect
8. emaneeventcommeffect.EventCommEffect

Sample Scripts

Review the following sample scripts:

ex01_publisher.py	Simple publisher example
ex02_subscriber.py	Simple subscriber example
ex03_publisher_pathloss.py	Simple pathloss publisher
ex04_subscriber_pathloss.py	Simple pathloss subscriber that prints out event info
ex05_publisher_location.py	Simple location publisher
ex06_subscriber_location.py	Simple location subscriber that prints out event info
ex07_publisher_commeffect.py	Simple comm effect publisher
ex08_subscriber_commeffect.py	Simple comm effect subscriber that prints out event info

Workshop 20 - Putting it All together: Distributed RF Pipe MANET

The goal of this workshop is to have the entire class interact using a distributed RF Pipe deployment.

Deployment Description

This workshop contains a distributed RF Pipe Deployment using Virtual Transports. Each student will run their own NEM Platform Server hosting a single NEM.

Each student will run the following applications:

1. emane
2. emanetransportd
3. emaneeventd
4. olsrd
5. gpsd

The instructor will run the following application:

1. emaneeventservice

Use the **olsrviewer.sh** command to visually monitor olsrd.

Workshop 21 - Putting it All together: Centralized RF Pipe MANET

The goal of this workshop is to have the entire class interact using a centralized RF Pipe deployment.

Deployment Description

This workshop contains a centralized RF Pipe Deployment using Virtual Transports. Each student will run their own NEM transport.

Each student will run the following applications:

1. emanetransportd
2. emaneeventd
3. olsrd
4. gpsd

The instructor will run the following application:

1. emane
2. emaneeventservice

Use the **olsrviewer.sh** command to visually monitor olsrd.

Workshop 22 - Putting it All together: Distributed Comm Effect MANET

The goal of this workshop is to have the entire class interact using a distributed Comm Effect deployment.

Deployment Description

This workshop contains a distributed Comm Effect Deployment using Virtual Transports. Each student will run their own NEM Platform Server hosting a single NEM.

Each student will run the following applications:

1. emane
2. emanetransportd
3. emaneeventd
4. olsrd
5. gpsd

The instructor will run the following application:

1. emaneeventservice

Use the **olsrviewer.sh** command to visually monitor olsrd.

Workshop 23 - Putting it All together: Centralized Comm Effect MANET

The goal of this workshop is to have the entire class interact using a centralized Comm Effect deployment.

Deployment Description

This workshop contains a centralized Comm Effect Deployment using Virtual Transports. Each student will run their own NEM transport.

Each student will run the following applications:

1. emanetransportd
2. emaneeventd
3. olsrd
4. gpsd

The instructor will run the following application:

1. emane
2. emaneeventservice

Use the **olsrviewer.sh** command to visually monitor olsrd.

Workshop 24 - Putting it All together: Centralized IEEE 802.11 abg MANET

The goal of this workshop is to have the entire class interact using a centralized IEEE 802.11 abg deployment.

Deployment Description

This workshop contains a centralized IEEE 802.11 abg Deployment using Virtual Transports. Each student will run their own NEM transport.

Each student will run the following applications:

1. emanetransportd
2. emaneeventd
3. olsrd
4. gpsd

The instructor will run the following application:

1. emane
2. emaneeventservice

Use the **olsrviewer.sh** command to visually monitor olsrd.