,

XML-Konfigurationseinstellungen

Mesh Manager
Infrastructure and Management Platform for a Wireless Testbed

Gruppe 4:	Benedikt Bakker <Benedikt.Bakker@stud.tu-darmstadt.de> Sascha Hans-Jürgen Dutschka <sascha.dutschka@arcor.de> Julian Harbarth <julian.harbarth@stud.tu-darmstadt.de> Sebastian Kropp <Sebastian.Kropp@stud.tu-darmstadt.de>
Teamleiter:	Benjamin Klein <klein.benjamin@gmail.com>
Auftraggeber:	Marc Werner <marc.werner@seemoo.tu-darmstadt.de> Secure Mobile Networking Lab - SEEMOO Fachbereich 20 <Informatik> Tobias Lange <tobias.lange@seemoo.tu-darmstadt.de> Secure Mobile Networking Lab - SEEMOO Fachbereich 20 <Informatik>
Stand:	30.03.2015

TECHNISCHE UNIVERSITÄT DARMSTADT

Bachelor-Praktikum WS 2014/2015
Fachbereich Informatik

Inhaltsverzeichnis

1 XML-Eingabeformate

Das System benötigt eine Anzahl von Informationen, um seine Aufgaben zu erfüllen. Dazu gehörten zum Beispiel eine Liste von Knoten oder die Topologie des Netzwerks. Ohne diese Informationen können manche Funktionen nur eingeschränkt oder gar nicht ausgeführt werden.

Diese Informationen werden im XML-Format festgehalten und im WebContent Ordner des Servers hinterlegt. Beim Starten werden die Daten eingelesen, allerdings ist ein neu einlesen während der Laufzeit über die REST Schnittstelle des Controllers möglich, falls Informationen geändert, hinzugefügt oder gelöscht werden sollen.

Im folgenden wird beschrieben welche Informationen dem System über die XML-Dateien zur Verfügung gestellt werden müssen, dazu wird jede XML-Datei beschrieben und an Hand eines Beispiels die die verschiedenen Tags erläutert.

2 XML-Konfigurationen des Controller

2.1 nodes.xml

Die nodes.xml beschreibt alle verfügbaren Knoten im Netzwerk. Ein Knoten besteht aus beliebig vielen Komponenten, die aus beliebig vielen Interfaces bestehen. Jedes Interface benötigt einen im Knoten eindeutigen Rollennamen. Dieser Rollenname kann in der configs.xml verwendet werden, um die einzelnen Interfaces zu verbinden. Eine Knotendefinition beginnt mit dem <id> Tag und endet mit einem erneuten Erscheinen des <id> Tags. Als Stromquellen und Ports müssen selbstverständlich legale Namen eingetragen werden.

```
<Nodes>
<id>Node A</id>

        <type>APU+WARP</type>

<component type = "WARP">
        <trunk>NetGear2;1</trunk>
        <powerSource>AeHome1;1</powerSource>

        <name>eth0</name>
        <port>NetGear2;2</port>
        <role>WARP A</role>

        <name>eth1</name>
        <port>NetGear2;3</port>
        <role>WARP B</role>
</component>

<component type = "APU">
        <trunk>NetGear2;1</trunk>
        <powerSource>AeHome1;2</powerSource>

        <name>eth2</name>
        <port>NetGear2;4</port>
        <role>APU 1</role>

        <name>eth3</name>
        <port>NetGear2;5</port>
        <role>APU 2</role>

        <name>eth4</name>
        <port>NetGear2;6</port>
        <role>APU 3</role>
```

```
        </component>

                <building>Building 1</building>
                <room>Room right</room>

        <id>Node C</id>

                <type>APU PoE</type>

        <component type = "APU">
                <trunk>NetGear1;1</trunk>
                <powerSource>AeHome1;3</powerSource>
                <name>eth0</name>
                <port>NetGear1;2</port>
                <role>APU 1</role>
        </component>

                <building>Building 2</building>
                <room>Room left</room>

        <id>Node B</id>
                ...

</Nodes>
```

<Nodes></Nodes> Umschließt die Definition der einzelnen Knoten.
<id></id> Enthält die ID des im folgenden beschriebenen Knoten. Ein Knoten besteht aus beliebig vielen Komponenten.
<component = TYP></component> Umschließt die Komponente und beschreibt den Typ der folgenden Komponente. Eine Komponente besteht aus beliebig vielen Interfaces
<trunk></trunk> Enthält den Trunkport/Uplink mit dem diese Komponente direkt verbunden ist. Muss den namingconventions für Ports folgen. Sollte auf der gleichen Netzwerkkomponente liegen wie der port
<powerSource></powerSource> Beschreibt die Stromquelle der Komponente.
<name></name> Indiziert, dass eine Interfacedefinition folgt und eine eventuell vorhergegangene Definition endet. Beschreibt den Namen des Interfaces.
<port></port> Beschreibt den Port des Switches an den das Interface angeschlossen ist.
<role></role> Beschreibt die Rolle des Interfaces innerhalb des Knotens. Muss eindeutig sein innerhalb des Knotens.
<building></building> Metainformation. Beschreibt das Gebäude des Knotens.
<room></room> Metainformation. Beschreibt den Raum des Knotens.

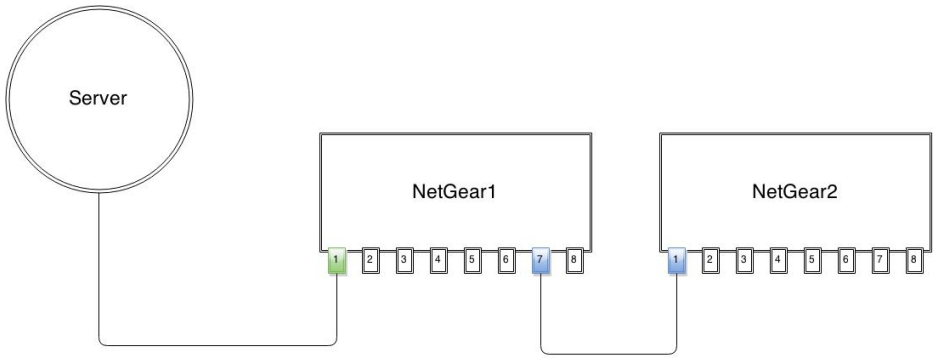
2.2 topology.xml

Die topology.xml beschreibt die Topologie des Netzwerkes. Diese wird benötigt um die einzelnen Knoten und deren Komponenten zu verbinden. Die Topologie wird als Graph dargestellt mit einer Menge von Knoten und einer Menge von Kanten. Jeder Knoten ist ein **Trunk**. Einer der Knoten muss als Startknoten ausgewählt werden, zu diesem Trunk wird das globale Experimentier VLAN gelegt. Dies ist sinnvollerweise der Trunk zum Server. Eine Kante besteht aus der Verbindung von zwei Knoten, es handelt sich um ungerichtete und ungewichtete Kanten. Die Knoten innerhalb der Kante werden mit dem Zeichen '∼' getrennt. Kanten müssen nur zwischen zwei unterschiedlichen Netzkomponenten beschrieben werden. Kanten innerhalb einer Netzkomponente müssen nicht explizit angegeben werden.

```
<topology>

<vertex>NetGear1;1</vertex>
<vertex>NetGear1;7</vertex>

<vertex>NetGear2;1</vertex>

<edge>NetGear1;7~NetGear2;1</edge>

<startVertex>NetGear1;1</startVertex>

</topology>
```

<topology></topology> Umschließt die Definition der Topologie
<vertex></vertex> Beschreibt einen Knoten des Topologiegraphen. Ein Knoten muss den **Namenskonventionen für Ports** folgen.
<edge></edge> Beschreibt eine Kante innerhalb der Topologie, besteht aus zwei Knoten, getrennt von '∼'
<startVertex></startVertex> Bezeichnet einen Knoten aus der Menge aller Knoten, dieser Knoten wird der Startknoten, zu dem das globale Experimentiernetz aufgebaut wird.

Abbildung 2.1: Topologie



Topologie wie in topology.xml beschrieben

2.3 wports.xml

Die wports.xml beschreibt alle notwendigen Informationen um den Ports ein VLAN zuzweisen. Achtung: Die Ports in der wports.xml bezeichnen keine Switchports, sondern Ports, die in Büros an der Wand angebracht sind, um dort zum Beispiel seinen Computer anzuschließen. Diesen Ports wird das globale Experimentiert VLAN zugewiesen.

```
<wport>

<id>testport</id>
<port>netgear1;2</port>
<trunk>netgear1;1</trunk>
<building>building1</building>
<room>room1</room>

<id> ... </id>
        ...
</wport>
```

<id></id> Die eindeutige ID des Ports.
<port></port> Der Switchport an den ein Port angeschlossen ist.
<trunk></trunk> Der Trunk, der dem Switchport zugehörig ist.
<building></building> Metainformation: Das Gebäude in dem der Port gelegt ist.
<room></room> Metainformation: Der Raum in dem der Port gelegt ist.

2.4 configs.xml

Die configs.xml beschreibt verschiedene Knotenkonfigurationen, die ein Knoten annehmen kann. Dazu werden die <wire></wire> Tags benutzt. Jeder Knoten stellt eine gewisse Menge von Rollen zur Verfügung. <wire></wire> bezeichnet einen virtuellen 'Draht', der durch ein VLAN realisiert wird. Mit Hilfe dieses Drahtes werden ausgewählte Rollen miteinander verbunden, indem die Rollen mit dem Zeichen '~' innerhalb des Tags verbunden werden. Nicht jeder Knoten ist für jede Konfiguration geeignet, da nicht jeder Knoten jede Rolle zur Verfügung stellt. Die Rollennamen müssen mit den Rollennamen in der nodes.xml übereinstimmen. Enthält ein <wire></wire> Tag das Zeichen '*' so handelt es sich um ein **Globales VLAN**, das Experimentiernetz, welches auch mit dem VM-Server verbunden ist.

```
<configs>

<config name = "WARP+APU">
<wire>WARP B~APU 2</wire>
<wire>WARP A~APU 3~*</wire>
</config>

<config name = "APU">
<wire>APU 3~*</wire>
</config>

<config name = "WARP">
<wire>WARP A~WARP B~*</wire>
</config>

<config name = "WARP Special">
<wire>WARP B~APU 1~*</wire>
</config>

</configs>
```

<configs></configs> Bezeichnet die Liste aller Konfigurationen.
<config name = _____></config> Bezeichnet eine Menge von <wire></wire> Tags, die zusammen eine Konfiguration bilden. Das Attribut name gibt den Namen der Konfiguration an.
<wire></wire> Eine Konfiguration kann mehrere dieser Tags enthalten, sie verbinden die darin enthaltenen Rollen später miteinander.

2.5 reloadOnStartup.xml

Die reloadOnStartup.xml beschreibt die Einstellungen ob der Controller eine bereits vorhandene Liste von Experimenten einlesen soll und wenn ja welche. Die Liste der Experimente muss im WebContent Ordner des Servers zur Verfügung stehen.

```
<reload>

    <reloadOnStartup>1</reloadOnStartup>
    <reloadFile></reloadFile>

</reload>
```

<reload></reload> Umschließt die startup Informationen.
<reloadOnStartup></reloadOnStartup> Zwei Werte möglich. 1: Eine bereits vorhandene Liste wird eingelesen. 0: Es wird keine Liste eingelesen.
<reloadFile></reloadFile> Dieses Feld kann mit einem Namen einer bereits vorhandenen Liste ausgefüllt werden. Der Controller wird beim Starten versuchen diese Liste einzulesen. Sollte das Feld leer sein lädt er automatisch die zuletzt erzeugte Liste ein (sofern <reload> auf 1 gesetzt ist).

3 XML-Konfigurationen des NetService

Der NetService ist zuständig für das Erstellen, Aktualisieren und Löschen von virtuellen LANs. Informationen, die zur Erfüllung dieser Aufgabe nötig sind werden in den XML-Konfigurationsdateien im WebContent Ordner des NetServices hinterlegt. Dazu gehören eine Auflistung der Netzwerkhardware, die gesteuert werden muss, Konfigurationsdaten für die interne VLan Verwaltung und eine Auflistung aller Komponenten, die mit einem dauerhaften, statischen virtuellen LAN verbunden werden müssen.

3.1 netComponents.xml

Die netComponents.xml beinhaltet alle Informationen, die nötig sind um die verschiedenen Netzwerkkomponenten zu identifizieren und anzusprechen. Eine **Netzwerkkomponente** wird durch ein 4-Tupel beschrieben, wobei das erste Element immer den Tag <id></id> tragen muss, wohingegen die restlichen 3 Elemente frei vertauschbar innerhalb des Tupels sind.

```
<NetComponents>

<id>NetGear1</id>
<trunk>1;5;6;7;8</trunk>
<host>netgeargs1hostname</host>
<type>NetGearGS108Tv2</type>
<id> ... </id>
   ...
</NetComponents>
```

<NetComponents></NetComponents> Umschließt die Liste der Netzwerkkomponenten
<id></id> Muss eine global eindeutige, frei wählbare, ID enthalten. Durch diese wird das Gerät identifiziert.
<trunk></trunk> Eine Folge von ganzen Zahlen, die die Trunkports der Netzwerkkomponente bezeichnet. Jeder Port ist mit einem ';' von den anderen getrennt.
<host></host> Bezeichnet den Hostnamen oder die IP Adresse unter der die Komponente errreichbar ist.
<type></type> Enthält den Typ der Komponente. Kann nur Werte aus einer bestimmten Menge annehmen. Diese ist darüber definiert, welche Geräte durch das System unterstützt werden. Bisher in der Menge enthalten: 'NetGearGS108Tv2'

3.2 staticComponents.xml

Die staticComponents.xml enthält alle Informationen zu Knotenhardware, die mit dauerhaften, statischen virtuellen LANs verbunden ist, wie zum Beispiel das **PowerNet**, über welches alle Pakete verrsendet werden, die Steuerungsinformationen für die Powerkomponenten enthalten. Dazu müssen alle **Switchports**, an denen ein solches Gerät angeschlossen ist Mitglied in ein und dem selben Netzwerk sein. Die statischen Komponenten werden durch Tupel der Größe 3 beschrieben, dabei muss das erste Element eines jeden Tupels immer den Tag <id></id> tragen. Die beiden verbleibenden Elemente sind beliebig vertauschbar.

```
<staticComponents>

<id>PowerOutlet1</id>
<port>NetGear2;8</port>
<type>power</type>

<id>management1</id>
<port>NetGear2;7</port>
<type>management</type>

</staticComponents>
```

<staticComponents></staticComponents> Rahmt die Liste aller statischen Komponenten.
<id></id> Beschreibt eine eindeutige ID der **Statische Komponente**, die Eindeutigkeit muss nur innerhalb des NetServices gewährleistet werden. Die Namen aus dem PowerService für die jeweilige Komponente kann wiederverwendet werden.
<port></port> Der Switchport an dem die Komponente angeschlossen ist. Benennung folgt den **Namenskonventionen für Ports**.
<type></type> Die Art oder der Typ der beschriebenen Komponente. Dieser Wert bestimmt welchem virtuellen LAN die Komponente beim initialisieren zugewiesen wird. Folgende Werte sind möglich: 'power', 'management'.

3.3 vlan.xml

Die vlan.xml enthält verschiedene Konfigurationseinstellungen in Bezug auf virtuelle LANs. Die Reihenfolge der einzelnen Elemente ist unerheblich, auch existiert keinerlei Rückmeldung über etwaige ßinnloseËinstellungen. Sollte eine minimale ID größer als eine maximale ID gewählt werden, so stehen dem NetService keine VLANs zur Verfügung. Ein Überschneiden der beiden Bereiche ist nicht erlaubt und führt zu fehlerhaftem Verhalten.

```
<vlan>
<globalRangeMin>100</globalRangeMin>
<globalRangeMax>200</globalRangeMax>
<localRangeMin>201</localRangeMin>
<localRangeMax>210</localRangeMax>
<power>98</power>
<manage>99</manage>
</vlan>
```

<vlan></vlan> Beschreibt die Liste der Konfigurationseinstellungen.
<globalRangeMin></globanRangeMin> Die kleinste ID, die ein **Globales VLAN** erhalten kann.
<globalRangeMix></globalRangeMax> Die größte ID, die einem globalen VLAN zugewiesen werden kann.
<localRangeMin></localRangeMin> Die kleinste ID, die ein **Lokales VLAN** erhalten kann.
<localRangeMax></localRangeMax> Die größte ID, die einem lokalen VLAN zugewiesen werden kann.
<power></power> Die VLAN ID, die dem **PowerNet** zugewiesen wird.
<manage></manage> Die VLAN ID, die dem **ManagementNet** zugewiesen wird.

4 XML-Konfigurationen des PowerService

Der PowerService ist zuständig für die Steuerung der Stromversorgung einzelner Knoten. Diese ist steuerbar über Netzwerksteckdosen oder **Power over Ethernet**. Die API unterstützt die beiden eben ge nannten Möglichkeiten und bietet so dem Nutzer dynamisch Knoten ein- sowie auszuschalten. Um dies zu ermöglichen werden Informationen benötigt, die in den XML-Konfigurationsdateien des PowerService zu finden sind. Diese Daten werden im WebContent Ordner des PowerService abgelegt und können bei Bedarf neu eingelesen werden.

4.1 powerSupply.xml

Die powerSupply.xml beschreibt alle Stromversorgungen, die über das Netz gesteuert werden können und muss alle Informationen, die dazu nötig sind bereitstellen. Die Stromversorgungen müssen dabei in Tupeln der Größe 3 beschrieben werden. Das erste Element eines jeden Tupels muss den Tag <id></id> tragen. Die restlichen Attribute können frei untereinander vertauscht werden.

```
<Supplys>

<id>AeHome1</id>
<host>aehome1hostname</host>
<type>AeHome</type>

<id> ... </id>
        ...
</Supplys>
```

<Supplys></Supplys> Gibt an, dass eine Liste von Stromversorgungen beschrieben wird
<id></id> Muss eine global eindeutige, frei wählbare, ID zur Identifizierung der Stromversorungen beinhalten
<host></host> Gibt den Hostnamen oder die IP Adresse an, unter der die Stromversorgung zu erreichen ist.
<type></type Enthält den Typ der Stromversorgung. Kann nur Werte annehmen, die von der API unterstützt werden. Derzeitige Unterstützung: 'AeHome'.

5 XML-Konfigurationen des ServerService

Der ServerService ist für die Kommunikation zu einem VM-Server zuständig. Da es nicht sinnvoll ist, dass bei jedem Anlegen einer Instanz immer alle Attribute vom Benutzer im Webinterface eingegeben werden müssen, werden die meisten Attribute in der dazugehörigen XML Datei gespeichert und ausgelesen. So kann der Benutzer sich eines der angelegten Templates aussuchen, die gespeicherten Daten werden geladen und die Instanz angelegt. Die VM.xml Datei wird im WebContent Ordner des ServerServices gespeichert und kann von dort auch immer aktualisiert werden.

5.1 vm.xml

Die vm.xml Datei enthält alle Einstellungen einer Instanz, die nicht unbedingt vom Benutzer geändert werden sollen. Neben den Attributen der Instanz wird dort eine eindeutige id eingetragen werden, die den Namen des Templates festlegt und den Namen des Verwaltungstool der Instanzen. Sollten Attribute in Form einer Liste dem Tool übergeben werden, muss der Tag des Attributs und der Tag der Liste angegeben werden, z.B. für ”mode” in der ”nics” Liste lautet der Tag: nic_mode. Der erste Eintrag eines Templates in der XML-Datei muss den Tag id tragen, die restlichen Attribute sind nicht eine bestimmte Reihenfolge gebunden.

```
<Instances>
<id>Instanz1</id>
<server>ganeti</server>
<__version__>1</__version__>
<name_check>false</name_check>
<pnode>pxhost01.seemoo.tu–darmstadt.de</pnode>
<disk_template>plain</disk_template>
<conflicts_check>false</conflicts_check>
<ip_check>false</ip_check>
<nic_mode>bridged</nic_mode>
<os_type>debootstrap+wheezy</os_type>
<mode>create</mode>
<start>false</start>
...
</Instances>
```

Die Tags, ausgenommen von ”id”, ”server” und den Attributen in Listen, müssen genauso heißen, wie sie von dem Verwaltungstool der Instanzen verlangt werden.
Da nur der Ganeti Dienst zur Verfügung stand, ist der XML-Parser des ServerService auch nur auf die Listenattributen von Ganeti eingestellt. Sollte ein anderes Verwaltungsprogramm von Instanzen andere Listentags benötigen, muss dieses evtl. am Parser angepasst werden.

Glossary

Globales VLAN Das globale VLAN eines Experiments verbindet, je nach Konfiguration auf unterschiedliche Weise, die einzelnen Komponenten eines Knoten untereinander sowie mit dem VM-Server. 7, 11

Lokales VLAN Verbindet einzelne Komponenten eines Knoten untereinander, ist also lokal für jeden einzelnen Knoten. Variiert je nach gewählter Konfiguration. 11

ManagementNet Beschreibt das VLAN, welches alle Komponenten des Netzwerks, die Managementfunktionen bereitstellen, verbindet. 11

Namenskonventionen für Ports Ein Port besteht aus zwei Teilen, die beide mit einem Semikolon voneinander getrennt sind. Der erste Teil besteht aus einer ID, die eine Netzwerk Komponente eindeutig identifiziert. Der zweite Teil besteht aus einer ganzen Zahl, die den exakten Port der vorher gennanten Komponente beschreibt. Beispiel: NetGear;1. 5, 10

Netzwerkkomponente Eine NetComponent beschreibt ein Gerät, welches aktiv am Netzaufbau mitwirkt und Teil der Topologie ist. Also zum Beispiel Switches oder Router. 9

Power over Ethernet Beschreibt die Technik über ein Ethernet Interface nicht nur Daten, sondern auch Strom zu beziehen und so die Stromzufuhr aus dem Netzwerk heraus steuerbar zu machen. 12

PowerNet Beschreibt das VLAN, welches alle Komponenten des Netzwerks, die anderen Komponenten als Stromquelle dienen und steuerbar sind, verbindet. 10, 11

Statische Komponente Beschreibt eine Komponente des Netzwerks, die immer dem gleichen virtuellen LAN zugewiesen ist. Dieser werden beim Starten des NetServices initialisiert. 10

Switchports Ein Switchport bezeichnet ein physikalisches Interface an einem Switch, welches mit einer Nummer innerhalb des Switches eindeutig identifiziert werden kann. 10

Trunk Ein Trunk ist ein Port auf dem mehrere VLANs liegen, dieser leitet also nicht nur Pakete mit einer bestimmten ID weiter, sondern alle ankommenden und dient zur Verbindung, wenn mehrer VLANs über mehrere Geräte gespannt sind. 5