H2020 iP Over IcN- the betTer IP (POINT)

HowTo-SDN

Installation and Configuration of the OpenVSwitch and the OpenDayLight Controller in POINT

List of Authors:

George Petropoulos, Mays AL-Naday

1.Introduction

This document describes the steps required to install and configure an SDN environment. It comprises two main parts:

- SDN Switching with OpenVSwitch: covering the configuration of OVS-based switches in the POINT network, both for soft switches in linux boxes, and the PICA hardware.
- SDN Control: covering the download, installation and configuration of the OpenDayLight SDN Controller to realise the ICN over SDN functions in the POINT network.

There are two different options to deploy the ODL controller, (a) using the packaged Opendaylight Boron release artifacts, and (b) using the POINT Opendaylight application, which also packages a complete ICN-aware Opendaylight container. Appropriate commands to install prerequisite software and configure the development environment are provided, as well as compilation and execution instructions. Finally, the required Blackadder configuration file changes to deploy an ICN topology over SDN networks are specified.

2. OpenVSwitch (OVS) for POINT

2.1. Prerequisites

To install Openvswitch v2.4.0, follow the defined steps:

```
~ sudo -s
# cd /root
# wget http://openvswitch.org/releases/openvswitch-2.4.0.tar.gz
# tar zxvf openvswitch-2.4.0.tar.gz

# cd openvswitch-2.4.0/
#    ./configure    --prefix=/usr    --with-linux=/lib/modules/`uname
-r`/build
# make
# make install
# make modules_install
# rmmod openvswitch
# depmod -a
```

To disable openvswitch controller from starting on boot:

```
# /etc/init.d/openvswitch-controller stop
# update-rc.d openvswitch-controller disable
```

To start OVS server process:

```
# /etc/init.d/openvswitch-switch start
```

Verify that Openvswitch is properly installed:

```
# ovs-vsctl show
```

You can follow the exact same steps for any Openvswitch version. Just replace version 2.4.0 in the respective commands above with the desired version, e.g. 2.3.0.

2.2. Configuration Instructions

- The first thing one would need to set up in an OVS switch is a bridge that supports OpenFlow 1.3 protocol. To do so, use the run the following in with admin/root privileges (i.e. sudo):

  ```
  $ ovs-vsctl add-br <BRIDGE_NAME>
  $ ovs-vsctl set bridge -- <BRIDGE_NAME> protocols=OpenFlow13
  ```

  If the switch is a PICA8 hardware, the last line need to be extended to include datapath type:

  ```
  $ ovs-vsctl set bridge -- <BRIDGE_NAME> datapath_type=pica8
  protocols=OpenFlow13
  ```

- Next would be to add switch interfaces to the bridge and configure them to support flows with explicit port numbers, to do so run (again with an admin/root account):

```
$ ovs-vsctl add-port <BRIDGE_NAME> <IF_NAME> -- set interface
<IF_NAME> options:key=flow ofport_request=<PORT_NUMBER>
```

If the switch is a PICA8 hardware, then the last line need to be extended to configure the port type as ‘pica8’:

```
$ ovs-vsctl add-port <BRIDGE_NAME> <IF_NAME> -- set interface
<IF_NAME> type=pica8 options:key=flow
ofport_request=<PORT_NUMBER>
```

- With Pica8 hardware, one would need to configure the port mod to be ‘up’, in order for the interface to be activated, as follows:

```
$ ovs-ofctl -O OpenFlow13 mod-port <BRIDGE_NAME> <IF_NAME> up
```

- If the switch should also be connected to a SDN controller (e.g. ODL), then the controller must be set for the bridge - provided that the controller is reachable from the switch - in addition to the fail mode. The latter defines the behaviour of the switch in case the controller becomes unreachable:

```
$ovs-vsctl set-controller <BRIDGE_NAME>
tcp:CONTROLLER_IP_ADDRESS:CONTROLLER_TCP_PORT
```

The controller tcp port is normally set to `6633`.

```
$ovs-vsctl set-fail-mode <BRIDGE_NAME> <secure|standalone>
```

Upon completion of these configurations, the switch is ready to be deployed in a POINT network.

3.Packaged Opendaylight Boron release

3.1.Prerequisites

1. Install OpenJDK 8.
   For any Ubuntu version (14-16.04), execute the following commands:
   `sudo add-apt-repository ppa:webupd8team/java -y`

```
sudo apt-get update
sudo apt-get install oracle-java8-installer
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

2. Download the pre-built tar for Boron release from the official Opendaylight downloads page: https://www.opendaylight.org/downloads.
   Latest tar file at the time of writing was: https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/distribution-karaf/0.5.2-Boron-SR2/distribution-karaf-0.5.2-Boron-SR2.tar.gz

3.2.Execution Instructions

Unpack the tar file into an appropriate folder:

```
~ tar xvzf distribution-karaf-0.5.2-Boron-SR2.tar.gz
~ cd distribution-karaf-0.5.2-Boron-SR2
~ ./bin/karaf
```

At the Opendaylight console, install the appropriate features to install flow rules:

```
> feature:install odl-openflowplugin-flow-services
odl-openflowplugin-southbound odl-openflowplugin-flow-services-rest
odl-l2switch-all
```

To install more features x, y, etc, use the same command:

```
> feature:install x y
```

To uninstall any feature, use the following command:

```
> feature:uninstall x
```

To check the execution log, execute:

```
> log:tail
```

To shutdown the Opendaylight controller, execute:

```
> shutdown -f
```

4.Packaged Opendaylight Boron release

4.1.Prerequisites

1. Install OpenJDK 8.

   For any Ubuntu version (14-16.04), execute the following commands:

   ```
   ~ sudo add-apt-repository ppa:webupd8team/java -y
   ~ sudo apt-get update
   ~ sudo apt-get install oracle-java8-installer
   ~ sudo update-alternatives --config java
   ~ sudo update-alternatives --config javac
   ```

2. Install Maven 3.3.3.

   ```
   ~ wget
   http://mirrors.sonic.net/apache/maven/maven-3/3.3.3/binaries/ap
   ache-maven-3.3.3-bin.tar.gz
   ~ tar -zxf apache-maven-3.3.3-bin.tar.gz
   ~ sudo cp -R apache-maven-3.3.3 /usr/local
   ~ sudo ln -s /usr/local/apache-maven-3.3.3/bin/mvn /usr/bin/mvn
   ```

   `~ mvn -version` (to verify Maven installation)

3. As the POINT-enabled SDN application is an Opendaylight Boron application, it is essential that the development environment for implementing such applications must be configured properly. Hence, follow the provided instructions by the Opendaylight community, available at:

   https://wiki.opendaylight.org/view/GettingStarted:Development_Environment_Setup

   Specifically, update the Maven settings to include the Opendaylight repositories.

   ```
   ~ mkdir ~/.m2
   ~ wget -q -O -
   ```

```
https://raw.githubusercontent.com/opendaylight/odlparent/master
/settings.xml > ~/.m2/settings.xml
```

4.2.Build Instructions

From the root POINT folder, execute the following commands:

```
~ cd sdn/
~ mvn clean install -DskipTests
```

To generate the javadoc documentation execute the following command. It will be available at the target/site/apidocs/index.html.

```
~ mvn javadoc:javadoc
```

To generate the doxygen documentation execute the following command. It will generate the html and latex documentation at target/doxygen/html/index.html and target/doxygen/latex/index.html.

```
~ doxygen doxygen.config
```

4.3.Execution Instructions

To run the POINT-enabled application execute the following commands:

```
~ ./distribution/opendaylight-karaf/target/assembly/bin/karaf
```

Install the POINT feature with the following commands in the Opendaylight console:

```
> feature:install tm-sdn
> feature:install point-ui
```

To check the execution log, execute the following command. You will see a few messages declaring that ICN modules have been successfully initialized and configured.

```
> log:tail
```

To also check the Opendaylight user interface, use your browser to navigate to http://localhost:8181/index.html. In the left menu you will also see the ICN-SDN tab.

To shutdown the Opendaylight controller, execute:

```
> shutdown -f
```