H2020 iP Over IcN- the betTer IP (POINT)

Examples

Deploy an ICN-over-SDN Network without a SDN Controller

Author: Mays AL-Naday

1. Overview

This example illustrates the steps required to construct a basic ICN-over-SDN network setup, using the core ICN part of the POINT platform (aka Blackadder), over which any ICN application can run, including the NAP app to construct an IP-over-ICN-over-SDN setup. The SDN network here does not facilitate an explicit controller and instead relies on the default behaviour of the switch to show the POINT capability of having path-based flow switching. In more complex setups it is highly recommended to run a controller, to minimize any ‘undefined behaviour’ from the switches. To learn how to run a setup that has a controller (OpenDayLight), please check the example “ICN_over_SDN_with_ODL_Controller”. Notice that POINT ICN-SDN implementation comes with two types of SDN connectivity setups:

1. pipeline (`tables`), which allows for pipelined flow tables, relying on the ‘submit’ action to the next table as a way of realising multicast switching with a small size flow table.
2. bridges, where multiple bridges are created and ports are divided into subsets, each of which is assigned to one bridge. Similarly the objective here is to cover the multicast possibilities with small size flow table.

This example shows how to construct the ICN-over-SDN network following the pipeline implementation. The topology considered in this example is shown below in Figure 4.1:

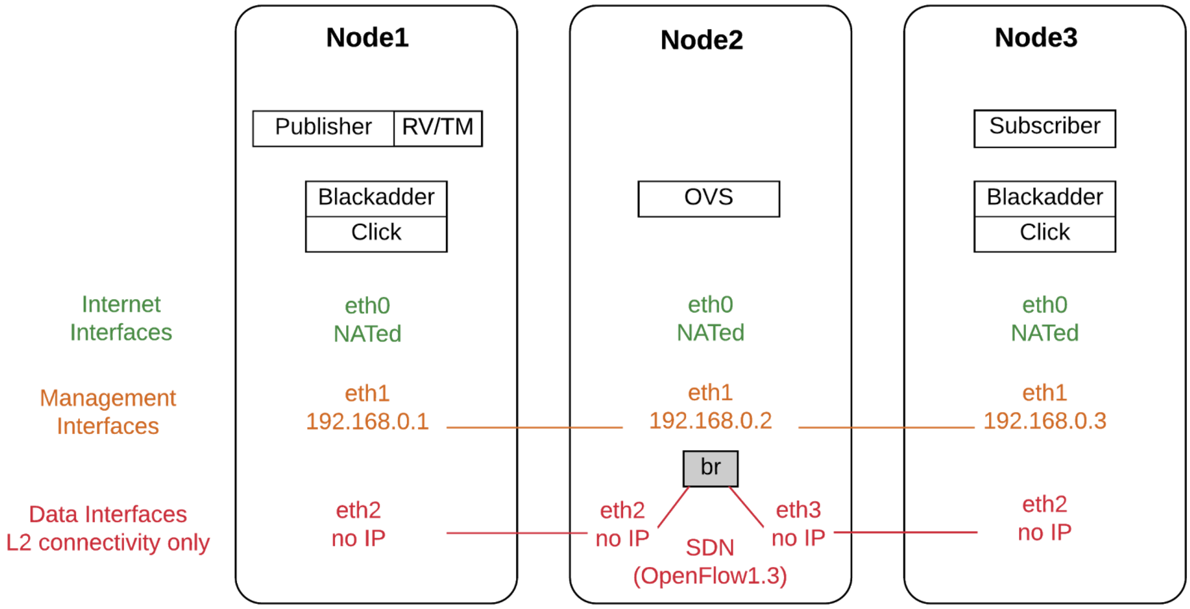

Figure 1: ICN-over-SDN Topology

In this topology, management and data connectivity are separated and provided through different interfaces. However, it is possible to provide both networks jointly through the same interface.

2. Prepare the Setup

1. Get three nodes - possibly Virtual Machines (VMs) - prepared with Linux based operating system. We have tested this example over Ubuntu 14.04 server edition; however, if losing the desktop is too much for you at this stage, Ubuntu Desktop will do just fine as well.
2. Make sure that each machine has at least three interfaces, with the exception of Node2 having 4 interfaces:
    a. Internet interface (`eth0`): this should connect the node to the Internet. if your nodes are VMs, this interface can be configured as NAT interface in VBox or a bridged interface to the NIC of your host machine, which connects to the Internet.
    b. Management Interface (`eth1`): this interface is used for managing the network through standard IP tools such as, ssh, scp , ping,..etc.
    c. ICN (data) Interface (`eth2 and eth3 for Node2`): the interface(s) used to establish ICN connections between the nodes. These interface do not

require IP addresses, and if one is provided no error will occur, but it will just be ignored; except for Node2, as the interfaces **must not** be given IP addresses.

Notice: Because our solution relies on the usage of the 'all-zeros' in the destination MAC field of the ethernet frame, as a way of distinguishing ICN packets from the rest, all data interfaces need to be set to 'promiscuous' mode so that the nodes can pick-up packets even when the destination MAC is different from that of the receiving interface. The instruction to set the interface in promisc mode is provided in the Configuration section below.

3. Install click and the POINT platform on all three nodes. For instructions on download, installations and configurations, please refer to the HowTo document

2.1 OpenVSwitch

OpenVSwitch is required in Node2 to provide SDN connectivity between Node1 and Node3 based on our ICN-SDN forwarding mechanism, described in D3.1. You can either install OVS (version 2.3.0) by itself, or install mininet (2.2.1) with '–nfv' options (if the node is intended to host a mininet cluster in the future) and it will come as part of it. For instructions on installing mininet, please refer to the HowTo.

2.2 Configuration

2.2.1 Credentials

To allow the deployment tool to use the configuration file for constructing the ICN setup and access all three nodes , they need to have the same credentials (i.e. username and password) for remote login. To do so, create on each node an account with a fixed username and password (e.g. username = point, password = point). **This must be the same for all three nodes**.

2.2.2 Interface Configuration

configure the management and data interfaces on each node by editing the interfaces file as follows (use any editing tool of your preference):

On Nodes 1 & 3:

```
~$sudo nano /etc/network/interfaces
```

add the following lines (replace X with the host address of the corresponding node):

```
auto eth0
interface eth1 inet dhcp

auto eth1
interface eth1 inet static
      address 192.168.0.X
      netmask 255.255.255.0
      broadcast 192.168.0.255

auto eth2
interface eth2 inet manual
up /sbin/ifconfig eth2 promisc on
down /sbin/ifconfig eth2 promisc down
```

save and exit the file

On Nodes 2:

```
~$sudo nano /etc/network/interfaces
```

add the following lines :

```
auto eth0
interface eth1 inet dhcp

auto eth1
interface eth1 inet static
      address 192.168.0.2
      netmask 255.255.255.0
      broadcast 192.168.0.255

auto eth2
interface eth2 inet manual
up /sbin/ifconfig eth2 promisc on
down /sbin/ifconfig eth2 promisc down

auto eth3
interface eth3 inet manual
up /sbin/ifconfig eth3 promisc on
down /sbin/ifconfig eth3 promisc down
```

2.2.3 Passwordless SSH

In order to allow the deployment tool fast access to each node, without having to enter the node password for each remote login process, which can be a very time consuming process, create (if not existent already) a ssh key of the deploying node and copy it to each node in the setup (refer to Figure 1 for node details) :

```
~$ ssh-keygen -t rsa -b 2048
~$ ssh-copy-id point@192.168.0.x
```

2.2.4 OVS Bridge

In Node2, the SDN bridge (br) need to be created and configured to support OpenFlow1.3. Then both eth2 and eth3 should be added to the bridge and configured with openflow port numbers. To create the bridge, do (in Node2):

```
~$ sudo ovs-vsctl add-br br
```

Configure the bridge to support OpenFlow1.3:

```
~$ sudo ovs-vsctl set Bridge br protocols=OpenFlow13
```

Add interfaces eth2 and eth3 to the bridge and configure them with port numbers 2 and 3 respectively:

```
~$ sudo ovs-vsctl add-port br eth2 -- set interface eth2 options:key=flow ofport_request=2
```

```
~$ sudo ovs-vsctl add-port br eth3 -- set interface eth3 options:key=flow ofport_request=3
```

Now, Node2 is ready to operate as an SDN forwarder in the ICN network and you can move to deploying the ICN-over-SDN network.

3. Deploy the Network

The ICN topology of Figure 1 can be found in blackadder/deployment/examples . The configuration file is named icn_over_sdn_without_controller.cfg . To familiarize with the configuration file, refer to Section 3 of the HowTo.

Notice: Blackadder does not run on the SDN forwarder, as the latter merely performs path-based forwarding. Therefore, the domain RV/TM MUST not be the SDN node.

To deploy the ICN-over-SDN network, using the POINT deployment, do:

```
~$ cd ~/blackadder/deployment/
~$ ./deploy -c ./examples/icn_over_sdn_without_controller.cfg -l
```

Now you can verify that the network is operational by checking click is working in Nodes 1 and 3:

```
~$ pgrep click
```

Check the TM is working on Node1 (the last process number, much bigger than the rest):

```
~$ pgrep tm
```

Check that your ICN forwarding flows are placed correctly in Node2:

```
~$ sudo ovs-ofctl -O OpenFlow13 dump-flows br
```

This should look like the following:

```
 cookie=0x0, duration=190.369s, table=0, n_packets=0, n_bytes=0,
priority=200,ipv6,dl_dst=00:00:00:00:00:00,ipv6_src=::128.0.0.0/::
128.0.0.0 actions=output:2,resubmit(,1)

 cookie=0x0, duration=190.37s, table=0, n_packets=0, n_bytes=0,
priority=100,ipv6,dl_dst=00:00:00:00:00:00 actions=resubmit(,1)

 cookie=0x0, duration=1764.563s, table=0, n_packets=8,
n_bytes=648, priority=0 actions=NORMAL

 cookie=0x0, duration=190.368s, table=1, n_packets=0, n_bytes=0,
priority=200,ipv6,dl_dst=00:00:00:00:00:00,ipv6_dst=::4000:0:0:0/:
:4000:0:0:0 actions=output:3

 cookie=0x0, duration=190.367s, table=1, n_packets=0, n_bytes=0,
priority=100,ipv6,dl_dst=00:00:00:00:00:00 actions=drop
```

3.1 Run a Pub/Sub Application

You can test the operation of the network by running any of the provided Pub/Sub applications (including the NAP). Here, we will use the Pub/Sub ping app for simplicity.

- In Node1:

  ```
  ~$ cd blackadder/examples/traffic_engineering
  ```

  - Run the ping publisher (request 5 pings):

    ```
    ~$ sudo ./ping_publisher 0 3 1 5
    ```

- In Node3:

  ```
  ~$ cd blackadder/examples/traffic_engineering
  ```

  - Run the ping subscriber:

```
        ~$ sudo ./ping_subscriber 0 3
```

If you see the pings flow between the two nodes then your test is complete and the network is fully operational. Well Done..!