H2020 i**P** **O**ver Ic**N**- the bet**T**er IP (POINT)

Examples

Deploy a Resilient POINT Network

Author: Mays AL-Naday

1. Overview

This example shows how to deploy a resilient POINT network that is capable of detecting node or link unavailability and alter the relevant delivery paths to avoid affected areas. This capability is facilitated by two solutions:

- Path Management (PM), which is based on distributing path information to all nodes; and, when a change occurs, it is published to all nodes. It is then up to the node to request a new path for continuing deliveries. Deploying this solution requires running the Link State Monitor application (LSM) and the TM with the extension '`-p`', as will be shown in later parts of this example.
- Information Resiliency, which is based on having a central *Resiliency Manager (RM)* that maintains the state of delivery paths and communicates with the Topology Manager to reoptimize the delivery paths after a network change. Deploying this solution requires running the LSM, the TM with the extension '`-r`' and running the RM

In this example we demonstrate both two solutions, while leaving the choice of preference to the user.

Notice, This example contains added network complexity that may not be required by all setups, for first time users of the POINT platform, it is recommend to try first to run the basic examples like the icn_only_point and ip_http_over_icn, before this example.

The example network setup (illustrated in Figure 1) consists of **six** machines (possibly VMs), organised as follows:

- **Core Network**: Four ICN nodes (ICN1-4), connected to each with redundant links, one node act as the RV/TM (in the second part of the example this will also act as RM), while two other nodes act as Network Attachment Points (NAPs):
  - **Client-side NAP (cNAP):** connects to an HTTP client and act as a publisher of requests
  - **Server-side NAP (sNAP):** connect to the HTTP server (web.point)
- **Edge Network,** consisting of two nodes:
  - One **HTTP client**
  - And, another that is a **HTTP server**

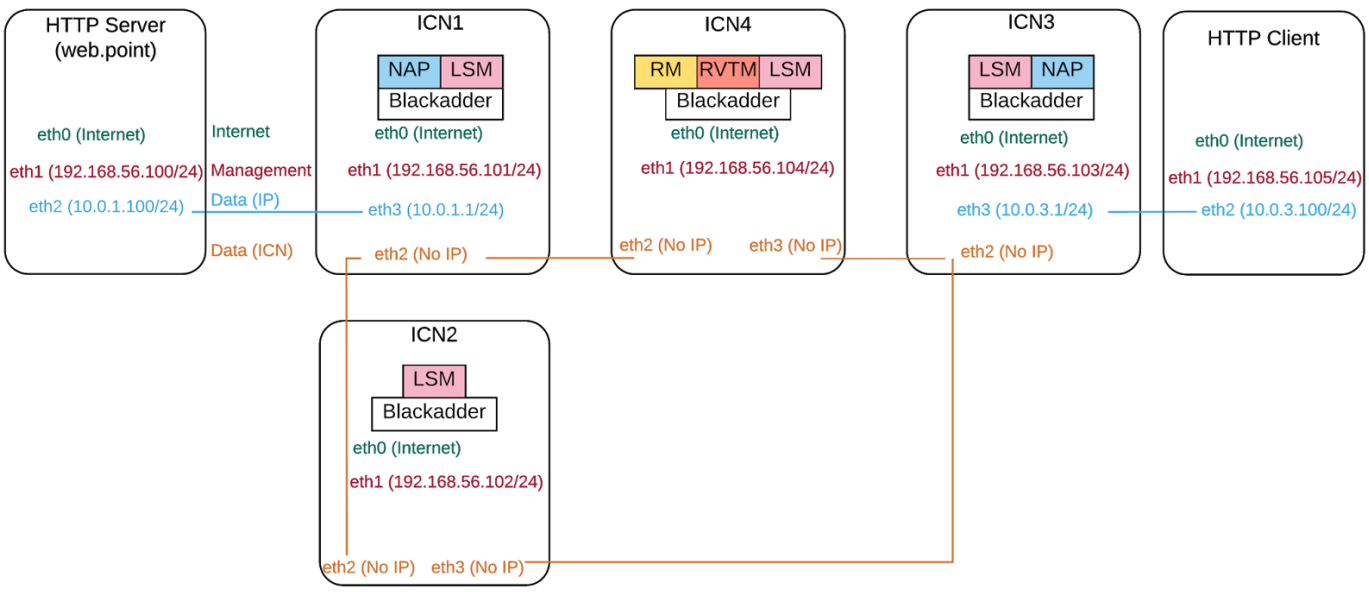

Figure 1: Example of a POINT network that supports resiliency and path management

Each of the ICN nodes need four ethernet interfaces, while the IP end nodes, need three interfaces. The assignment of interfaces is as follows:

- **Eth0:** provides internet connectivity for update, download,..etc. It can be bridged or NATed.
- **Eth1:** assigned IP address (192.168.56.x) as per the figure above. The interface is used for remote login to the node
- **Eth2:** in the ICN nodes, it is used for ICN data links, in the IP nodes it is used for IP data.
- **Eth3:**
    - Used for ICN data in the ICNs that don't run the NAP application.
    - Used for IP data in the ICNs that run the NAP application

2. Configuration

2.1 Prerequisites

- Set Up six machines (virtual or physical, doesn't matter) with linux ubuntu 14.04 64bit server edition.
- On the ICN nodes, install: openssh server, git, build-essentials, click router and the POINT platform. For detailed instructions of the installation & configuration of ICN nodes, please refer to the main HowTo of this repository.
- On the client machine add a static DNS configuration of the web server for the FQDN `web.point`. To do so, open the hosts file with sudo privileges:
  ```
  $ sudo nano /etc/hosts
  ```
  Add the following line, then save and close the file:
  ```
  10.0.1.100 web.point
  ```
- On the server machine, install apache2 as follows:
  ```
  $ sudo apt update
  $ sudo apt install apache2
  ```
  Without any further configuration, this gives a HTTP server with default configurations. You can customize the configuration of your server if you like, but for the purpose of this example I will continue to use the default configs.
    - On the same machine, place a large file (50MB+) of your choice in the `/var/www` directory, the file will be used later in a progressive download session

2.2 Interfaces

In each IP node, configure the interfaces in the network configuration file (don't forget to replace 'x' in the IP address as shown in the figure):

```
~$ sudo nano /etc/network/interfaces
auto eth0
```

```
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
      address 192.168.56.x
      netmask 255.255.255.0
      broadcast 192.168.56.255

auto eth2
iface eth2 inet static
      address 10.0.x.2
      netmask 255.255.255.0
      broadcast 10.0.x.255
```

In each ICN node not running the NAP, configure the interfaces in the network configuration file (don't forget to replace 'x' in the IP address as shown in the figure):

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

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
      address 192.168.56.x
      netmask 255.255.255.0
      broadcast 192.168.56.255

auto eth2
iface eth2 inet manual

auto eth3
iface eth3 inet manual
```

In the ICNs running the NAP, put the same configurations for `eth0-2`, except for `eth3` which need to be configured with the IP-over-ICN subnet, as follows (don't forget to replace x with the corresponding subnet address):

```
auto eth3
iface eth3 inet static
      address 10.0.x.100
      netmask 255.255.255.0
      broadcast 10.0.x.255
```

2.3 Credentials

To allow the deployment tool to use the configuration file for constructing the ICN setup and access all three nodes , the latter 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**.

2.4 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 Nodex in the setup:

```
~$ ssh-keygen -t rsa -b 2048
~$ ssh-copy-id point@192.168.56.x
```

3. Configuration and Deployment

Deploying a resilient point network consists of three parts, first run the ICN network with the relevant extension depending on which solution to operate; second, activate the detection mechanism by running the Link State Monitoring (LSM) app on all nodes; and third, run the NAP to provide IP-to-ICN translation and vice versa.

3.1 NAP Configuration

First configure the NAPs with the relevant route prefixes and the `web.point` FQD, as follows:

- In the cNAP, open the nap.cfg: `$ nano /etc/nap/nap.cfg`, then:
  - Configure the NAP IP side, by setting:
    ```
    networkAddress = "10.0.3.0";
    netmask = "255.255.255.0";
    ```
  - add the route prefix of the sNAP in the '*routingPrefixes*' section, then save and close the file:
    ```
    {
          networkAddress = "10.0.1.0";
          netmask = "255.255.255.0";
    }
    ```
- In the sNAP, open the nap.cfg and:
  - Configure the NAP IP side:
    ```
    networkAddress = "10.0.1.0";
    netmask = "255.255.255.0";
    ```
- add the cNAP route in the *routingPrefixes'* section:
  ```
  {
        networkAddress = "10.0.3.0";
        netmask = "255.255.255.0";
  ```

```
}
```

- Add the FQDN of the server in the "*FQDN Registrations*" section, then save and close the file:

```
{
        fqdn = "web.point";
        ipAddress = "10.0.1.100";
        port = 80;
}
```

3.2 Network Deployment

Once the NAP configs are set up, the network can be deployed. Notice that the deployment tool can also be used to deploy the NAP and LSM applications, in addition to the TM and blackadder, if the relevant flags are provided. For this example setup, the network configuration can be found in `~/blackadder/deployment/examples/resilient_icn.cfg`

3.1.1 Path Management

To run the PM solution, deploy the network as follows:

```
$ ./deploy -c
~/blackadder/deployment/examples/resilient_ip_http_over_icn.cfg -r
-n -l -x -p
```

The flags mean:
`-r` for running the LSM
`-n` for running the NAP
`-l` for keeping logs of blackadder, the LSM and the TM. NAP logs are maintained differently in `/var/log/nap.log`
`-x` running the TM with an extension
`-p` the extension flag used by the TM

3.1.1 Central Resiliency

To run the RM solution, deploy the network:

```
$ ./deploy -c ~/blackadder/deployment/examples/resilient_icn.cfg
-r -x -r
```

Where the second '`-r`' is the resiliency flag used by the TM, careful not to confuse it with the first '`-r`' which is used by the deployment tool to deploy the LSM.

4. Test and Validation

On the client:

- Open a terminal and `ping web.point`, to make sure you are can reach the server

- Now request the large file from the server:
  `$ wget http://web.point/FILE_NAME`
  Notice: Even though the two possible paths in the network are of equal length, because ICN2 has lower `nodeID=2` than ICN4 of `nodeID=4`, the path will be established over ICN2.
- While the file is being downloaded at the client:
  - Break the delivery path by shutting down `eth3` of `ICN2`:
    `ICN2$ sudo ifdown eth3`
- Notice that the file download stops for few seconds before it resumes again.
- Look at the logs of the TM and you would notice that the TM has rerouted the path over `ICN4`
- If you reach to this level, i.e. the file resumes to download, then you have completed the example successfully.
- Once you finish your experiments, don't forget to put `eth3` back up:
  `ICN2$sudo ifup eth3`

Challenge: Next, see if you can run video applications that halt and resume!

For technical support or if you get stuck, contact us on: point-support@list.aalto.fi , referencing the example name as the email subject.