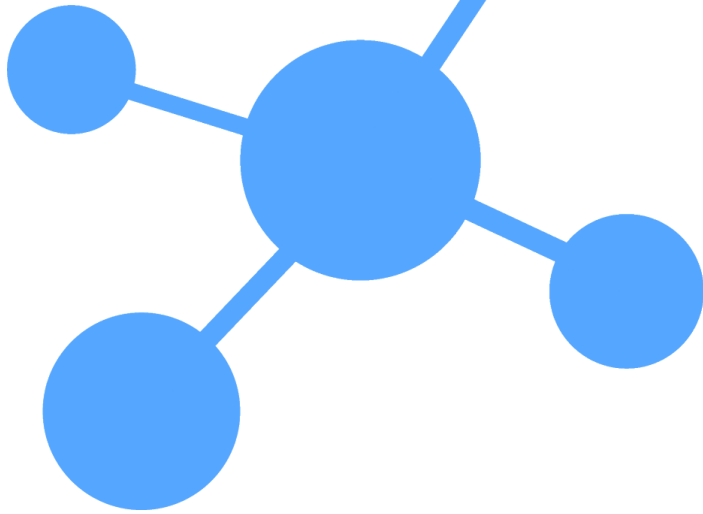


H2020 iP Over IcN- the betTer IP (POINT)

Design Description

Mininet Deployment

POINT



Author: Mays AL-Naday

[1. Overview](#)

[1.1 Logging](#)

[2. Requirements](#)

[3. Deployment and Configuration](#)

[3.1 Mininet Setup Configuration](#)

[3.2 Generate ICN Configurations](#)

1. Overview

The mininet deployment is a python based application that allows constructing clusters of mininet on multiple hosts and connect them to each other and to external networks. The topology of the mininets need to be defined in the tool, at the moment only one topology is defined, that is Direct Acyclic Graph (DAG), but other can be supported by developing the appropriate classes. Additionally, the tool also initiates mininet hosts that act as IP end points (particularly clients), each of which is connect directly to the respective NAP with an IP data-based link, while also connected directly to the main switch for remote management. Figure 1 shows an example mininet cluster on one host, that comprises two pairs of NAPs.

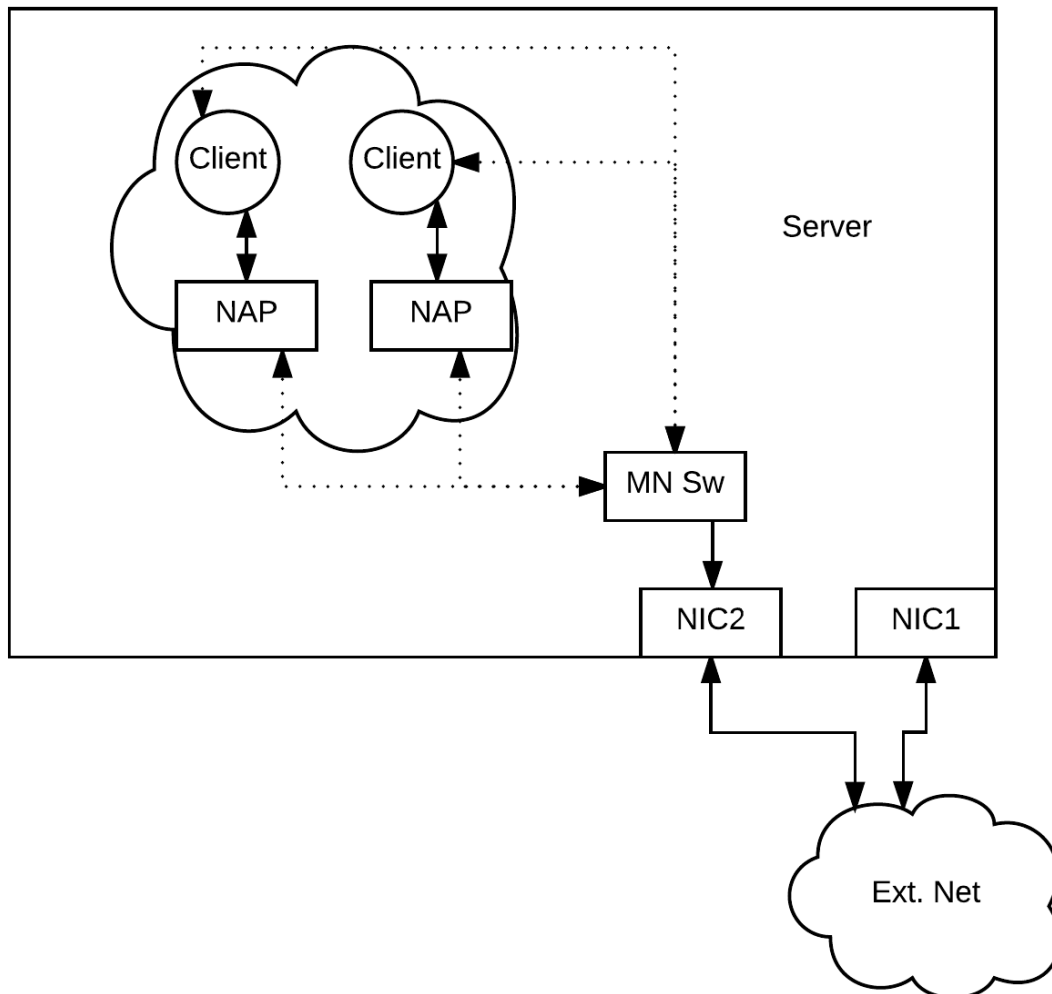


Figure 1: Example Mininet cluster that consists of two NAPs on one server for POINT emulation deployment.

The network of NIC1 allows remote access to the hosting server and provides connectivity during transition time of the mininet (i.e. bootstrap or shutdown). The network of NIC2 provides the host as well as the mininet cluster remote reachability from other nodes. The

latter is also used (on a Layer 2 comms) for ICN communications. When setting up the mininet cluster, the link between a client and its respective NAP is only accessible from the NAP or the client, but when the POINT network is fully deployed, the client also becomes accessible through this link from other clients (or servers) connected to the POINT network. Having such setup of a mininet cluster on one or multiple nodes, the ICN topology that comes on top may look like that shown in Figure 2 for two mininet clusters.

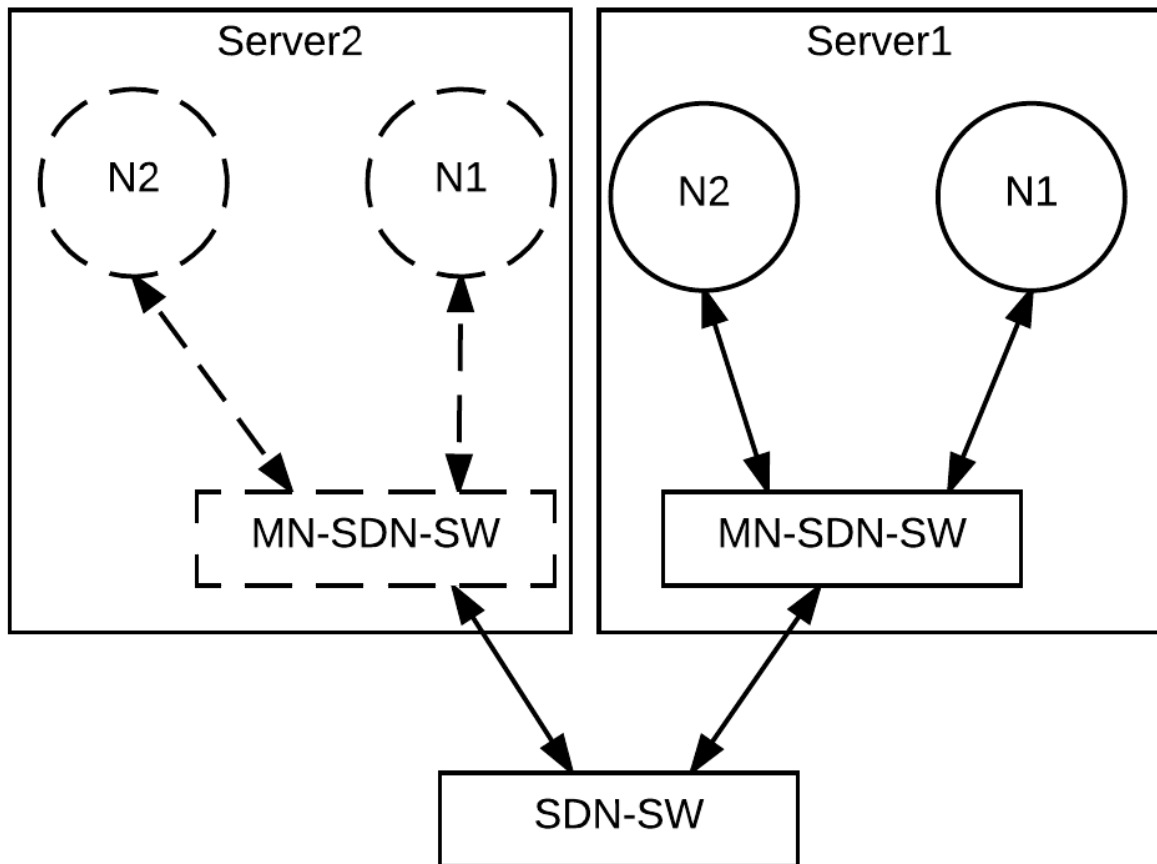


Figure2: Example ICN topology of two mininet clusters

1.1 Logging

The tool allows for four levels of logging during deployment, that can be configured from the command line:

- DEBUG: extensive logging
- INFO: brief logging, default level if no setting provided
- WARNING: only shows warnings
- ERROR: only shows errors

2. Requirements

- Each server has at least one physical interface and one virtual interface (i.e. vlanned interface)

- The virtual interface is assigned to NIC1 of Figure 1 and it is used for remote management of the server
- The physical (untagged) interface is used for NIC2 of Figure 1 and is used for remote management of the mininet cluster
- The Mininet hosting servers are connected to other nodes in the ICN network, either SDN switches or software boxes running Blackadder.
- An IP subnet and address range to be used for remote management of mininet
- Another IP subnet to be allocated to IP-over-ICN network (i.e. client/NAP pairs).

3. Deployment and Configuration

The tool allows for constructing clusters on multiple servers. It also allows for generating ICN configuration files that can be used by the ICN deployment tool. For that it uses two - JSON based - configuration templates.

3.1 Mininet Setup Configuration

This template provides the set of hosting servers to deploy mininet cluster, with configuration elements that allow for cluster-based customisation. Notably, the template allows for defining the subnets for mininet remote management and IP-over-ICN communications, along with the starting IP address. The tool would then sequentially assign the IP addresses for the mininet hosts.

3.2 Generate ICN Configurations

This template allows for defining a common ICN topology that would be deployed on multiple servers, along with the respective hosting servers. The tool use that to generate the ICN configuration file used by the ICN deployment tool. The topology template must define the edge node that connects the mininets to the rest of the ICN network.