Ping

NETMET Lab Exercises 2

Introduction

In this lab we are going to make a deep analysis of how the network tool *Ping* is working.
We will first use Ping along with a packet sniffer in order to understand what type of request is sent and what the responses are.
Then, we will put our hands on Python (>= 3.6) in order to mimic the Ping tool.

Discover by experimentation

In this section we will use Ping and understand by experimentation the underlying network concepts of this tool.
To see what Ping is doing on the network, we will use a network sniffer. You can use *Wireshark* (https://www.wireshark.org/) in your computer along with *tcpdump* on a EdgeNet pod.

1. Get the list of ready nodes

```
kubectl get nodes --kubeconfig edgenet-kubeconfig.cfg | grep ‘\sReady\s’
```

2. Create a pod in a ready node and connect to it

```
kubectl run <YOUR_NAME> --image=dioptraio/education --kubeconfig
edgenet-kubeconfig.cfg --overrides='{ "apiVersion": "v1", "spec": { "nodeSelector": {
"kubernetes.io/hostname": "<SELECTED_NODE>" }}}' --requests='cpu=1m,memory=16Mi’
```

3. Attach to it

```
kubectl exec -it <YOUR_NAME> --kubeconfig edgenet-kubeconfig.cfg -- /bin/bash
```

4. Experiment
    a. Ping
    b. tcpdump

    ```
    tcpdump -w dump.pcap
    ```

5. Get the dump file

```
kubectl cp <YOUR_NAME>:dump.pcap dump.pcap
```

Questions :

- What protocol(s) Ping is using to gather this information ?
- Detail the fields of the upper protocol.
- How is Ping calculating the round trip time (RTT) ?
- Give 3 reasons for not having a Ping response.

Code Ping logic

If you have some time left, code the logic of a Ping.

Help : This work can be done quite easily with the Python programming language.
I suggest you look at the documentation of Scapy which is a famous network prototyping Python library.

Clean-up

Before forget it, let's remove our vantage point(s):

```
kubectl delete pod <YOUR_NAME> --kubeconfig edgenet-kubeconfig.cfg
```