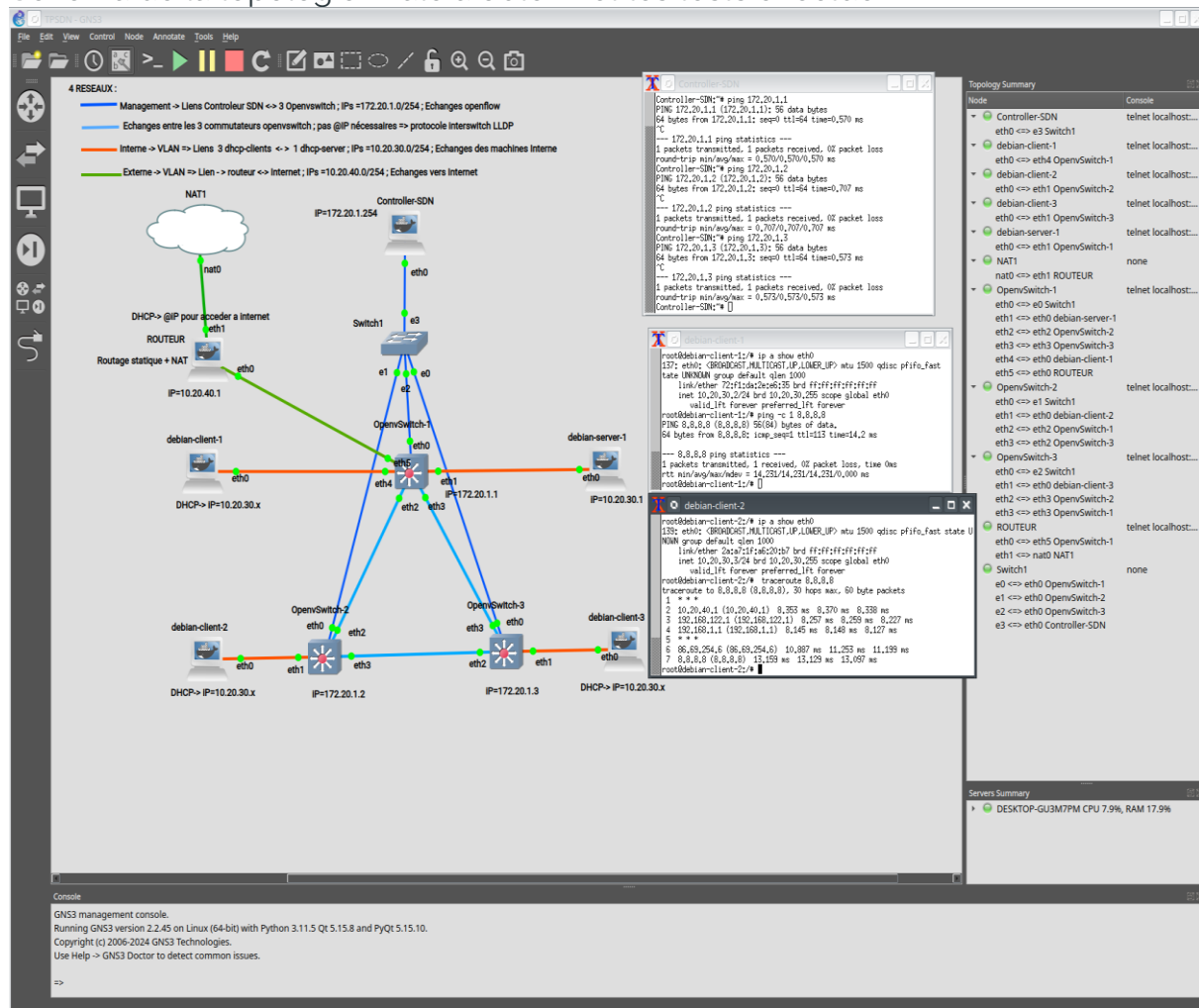


TP Openvswitch – SDN - OVN – Openflow

- Dans ce TP, l'objectif est d'utiliser l'outil GNS3 pour simuler des réseaux LAN gérés par OVS+Contrôleur_SDN et de configurer :
 - 3 commutateurs OpenVSwitch (OVS),
 - 3 postes en DHCP pour tester les liaisons,
 - 1 serveur DHCP pour paramétrer la configuration IP des postes,
 - 1 ROUTEUR pour assurer l'interconnexion avec Internet,
 - 1 Contrôleur SDN Faucet (c'est un contrôleur open-source pour les réseaux OpenFlow) pour configurer les réseaux gérés.

Schéma de la topologie finale à obtenir et les tests effectuer :



L'objectif est donc de créer en 3 étapes :

1. Un réseau de management entre un contrôleur SDN et 3 commutateurs OpenVswitch (de niveau 2 et 3).
2. Un réseau (VLAN) interne entre 1 serveur dhcp et 3 clients dhcp
3. Un réseau (VLAN) externe pour accéder à internet et configurer, grâce au contrôleur SDN, les VLAN et des routeurs virtuelles pour permettre le routage entre le réseau interne et Internet.

ETAPE 1

Configurer les réseaux bleus de MANAGEMENT

- Installer un Switch simple "non OVS" : Commutateur-de-gestion
- Installer 3 conteneurs Docker OpenVSwitch
- Installer 1 conteneur Docker Faucet
- Interconnecter ces 5 équipements avec le LAN MANAGE

Configurer l'@IP des 3 équipements openvswitch précisées dans schéma précédent (@IP=172.20.1.X) en "Modifiant" les "Configuration du réseau".

```
# Static config for eth0 - Management interface
auto eth0
iface eth0 inet static
    address 172.20.1.1
    netmask 255.255.255.0
```

Configurer l'@IP du controleur FAUCET

```
# Static config for eth0 - Faucet SDN
auto eth0
iface eth0 inet static
    address 172.20.1.254
    netmask 255.255.255.0
# Faucet needs a dummy DNS nameserver,
# otherwise it will fail to start
up echo nameserver 0.0.0.0 > /etc/resolv.conf
```

Lancer les 4 équipements et assurez-vous sur la console des commutateurs qu'ils ont des adresses IP de gestion comme suit :

- OpenvSwitch-1 : 172.20.1.1
- OpenvSwitch-2 : 172.20.1.2
- OpenvSwitch-3 : 172.20.1.3
- Controleur-SDN: 172.20.1.254

Vérifier les interconnexions depuis la console du controleur-SDN : ping 172.20.1.X ...

Configurer sur la console de chaque pont OVS l'IP du Controleur SDN FAUCET :

ATTENTION : Chaque pont doit avoir un id différent et il faut préciser le port qui mène au contrôleur .

```
$ ovs-vsctl set bridge br0 other-config:datapath-id=000000000000000001 -- set bridge br0
fail_mode=secure -- set-controller br0 tcp:172.20.1.254:6653 -- set controller br0 connection-
mode=out-of-band -- --if-exists del-port eth0
```

Verifier sur la console de chaque pont OVS :

```
$ ovs-vsctl get-controller br0
# => tcp:172.20.1.254:6653
```

ETAPE 2

Configurer le réseau orange Interne

- Installer 1 conteneur Docker dhcp-server
- Installer 3 conteneurs Docker dhcp-client
- Interconnecter ces 4 équipements aux Swtchs OVS pour contruire le VLAN Interne
- Configurer l'@IP du serveur DHCP (@IP=10.20.20.1) en "Modifiant" sa "Configuration du réseau".
- Lancer les 4 nouveaux équipements

Remarque : Pour éviter de lancer le Controlleur SDN faucet à la main la prochaine fois, vous pouvez spécifiez la commande de démarrage qui doit être exécutée au démarrage du conteneur (menu configurer) : `sh -c 'cd; faucet & sleep 10; ash -i -l'`

Depuis la console, configurer le Controleur SDN FAUCET (fichier : /etc/faucet/faucet.yaml.

Ce fichier permet de configurer les switchs OVS:

- les connexions avec les switchs OVS de niveau 2 (VLANs Interne et Externe)
- Les connexions de niveau 3 (routage + routeur virtuel qui devra généré et configuré)

```
$ cp /etc/faucet/faucet.yaml /etc/faucet/faucet.yaml.orig
$: cat > /etc/faucet/faucet.yaml
Controller-SDN:~# cat /etc/faucet/faucet.yaml
vlangs:
  interne:
    vid: 100
    description: "vlan interne"
    faucet_mac: "00:00:00:00:00:11"
    faucet_vips: ["10.20.30.254/24"]
  externe:
    vid: 200
    description: "vlan acces internet"
    faucet_mac: "00:00:00:00:00:22"
    faucet_vips: ["10.20.40.254/24"]
  routes:
    - route:
        ip_dst: "0.0.0.0/0"
        ip_gw: "10.20.40.1"

routers:
  router-interne-externe:
    vlans: [interne, externe]
```

```
dps:
  openvswitch-1:
    dp_id: 0x1
    hardware: "Open vSwitch"
    stack:
      priority: 1
    interfaces:
      2:
        name: "eth1"
        native_vlan: interne
      5:
        name: "eth4"
        native_vlan: interne
      6:
        name: "eth5"
        native_vlan: externe
      4:
        name: "eth3"
        description: "Link openvswitch-1 - openvswitch-3"
        stack:
          dp: openvswitch-3
          port: 4
      3:
        name: "eth2"
        description: "Link openvswitch-1 openvswitch-2"
        stack:
          dp: openvswitch-2
          port: 3

  openvswitch-2:
    dp_id: 0x2
    hardware: "Open vSwitch"
    interfaces:
      2:
        name: "eth1"
        native_vlan: interne
      3:
        name: "eth2"
        description: "Link openvswitch-2 - openvswitch-1"
        stack:
          dp: openvswitch-1
          port: 3
      4:
        name: "eth3"
        description: "Link openvswitch-2 - openvswitch-3"
        stack:
          dp: openvswitch-3
          port: 3
```

```
openvswitch-3:
  dp_id: 0x3
  hardware: "Open vSwitch"
  interfaces:
    2:
      name: "eth1"
      native_vlan: interne
    3:
      name: "eth2"
      description: "Link openvswitch-3 - openvswitch-2"
      stack:
        dp: openvswitch-2
        port: 4
    4:
      name: "eth3"
      description: "Link openvswitch-3 - openvswitch-1"
      stack:
        dp: openvswitch-1
        port: 4 ^D
```

Testez que votre fichier de configuration est bon. Le succès affichera une sortie au format JSON de votre configuration. L'échec affichera un message d'erreur :

```
$ check_faucet_config /etc/faucet/faucet.yaml
```

Déclenchez le controleur SDN pour recharger et utiliser la nouvelle configuration.

```
$ kill -HUP ryu-manager ou arrêter et redémarrer le conteneur
```

Configurer le serveur DHCP

```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.orig
```

```
# cat > /etc/dhcp/dhcpd.conf
```

```
subnet 10.20.30.0 netmask 255.255.255.0 {
  # specifie un domaine spécifique au sous-reseau
  option domain-name "mondomaine.org";
  option broadcast-address 10.20.30.255; # adresse de diffusion
  range 10.20.30.2 10.20.30.253; # plage adresses IP dynamiques
  option routers 10.20.30.254; # routeur par défaut
  default-lease-time 3600;
  max-lease-time 7200;
}
```

Lancer le serveur dhcpd : `#/etc/init.d/isc-dhcp-server start`

Remarque : Pour éviter de lancer le serveur à la main la prochaine fois, vous pouvez spécifier la commande de démarrage qui doit être exécutée au

démarrage du conteneur (menu configurer) : `/gns3/bin/busybox sh -c 'export TERM=vt100;cd;service isc-dhcp-server start;while true; do /gns3/bin/busybox sh;done '`

Après avoir relancer les conteneurs dhcp-client-x et depuis leurs consoles, vérifier les @IP obtenues par DHCP :

- Dhcp-clientx : 10.20.30.x

ETAPE 3

Configurer le réseau vert Externe

- Installer 1 conteneur Docker dhcp-server avec 2 interfaces nommé ROUTEUR
- Installer un accès NAT
- Interconnecter ces 2 équipements au Swtch OVS1 pour contruire le VLAN Externe
- Interconnecter le deuxième port du ROUTEUR au réseau NAT
- Configurer le port du ROUTEUR sur le réseau NAT en DHCP et l'autre port avec une @IP 10.20.40.1
 - `auto eth0`
 - `iface eth0 inet static`
 - `address 10.20.40.1`
 - `netmask 255.255.255.0`
 - `broadcast 10.20.40.255`
 - `auto eth1`
 - `iface eth1 inet dhcp`
 - `hostname routeur`

Dans cette configuration, nous allons nous intéresser au module netfilter et à iptables qui est une interface permettant de paramétrer ce module.

On rappelle que netfilter est un module du noyau Linux qui fournit :

- des fonctions de pare-feu et notamment le contrôle des machines qui peuvent se connecter, sur quels ports, de l'extérieur vers l'intérieur, ou de l'intérieur vers l'extérieur du réseau ;
- de traduction d'adresse (NAT) pour partager une connexion internet (masquerading), masquer des machines du réseau local, ou rediriger des connexions ;
- et d'historisation du trafic réseau.

Netfilter intercepte les paquets réseau à différents endroits du système (à la réception, avant de les transmettre aux processus, avant des les envoyer à la carte réseau, etc.). Les paquets interceptés passent à travers des chaînes qui vont déterminer ce que le système doit faire avec le paquet. En modifiant ces chaînes on va pouvoir bloquer certains paquets et en laisser passer d'autres. Chaque chaîne

possède des tables associées. La plus courante est la table « filter » qui permet de faire le filtrage des paquets.

Les trois chaînes principales :

- une chaîne INPUT pour filtrer les paquets à destination du système,
- une chaîne OUTPUT pour filtrer les paquets émis par les processus du système,
- et une chaîne FORWARD pour filtrer les paquets que le système doit transmettre (par ex. lorsqu'on fait du NAT).

En ajoutant des règles dans ces chaînes on pourra laisser passer ou jeter les paquets suivant certains

critères. Il est possible de créer par ailleurs ses propres chaînes.

La différence entre Netfilter et iptables est la suivante :

- Netfilter est un module, et fonctionne en mode Noyau. C'est lui qui intercepte et manipule les paquets IP avant et après le routage.
- iptables est la commande qui permet de configurer Netfilter en espace Utilisateur.

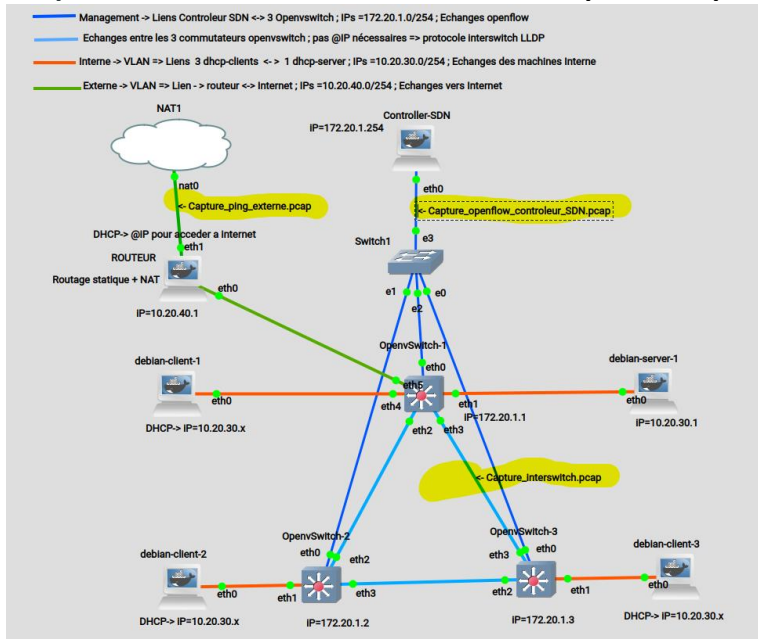
Ajoutez une règle routage pour router les trames revenant au réseau interne et une règle dans la table nat de sorte à mettre en place du SNAT dynamique (*masquerading*) permettant de remplacer l'adresse IP source privée par l'adresse IP externe associée à l'interface eth0 du nœud ROUTEUR pour tout paquet sortant via l'interface eth1.

- `ip route add 10.20.30.0/24 via 10.20.40.254 ;`
- `iptables -v -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth1 -j MASQUERADE;`
- `iptables -v -A FORWARD -i eth1 -o eth0 -d 10.0.0.0/8 -m state --state RELATED,ESTABLISHED -j ACCEPT`

Remarque : Pour éviter de lancer le serveur à la main la prochaine fois, vous pouvez spécifier la commande de démarrage qui doit être exécutée au démarrage du conteneur (menu configurer) : `/gns3/bin/busybox sh -c 'export TERM=vt100;cd;ip route add 10.20.30.0/24 via 10.20.40.254 ; iptables -v -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth1 -j MASQUERADE; iptables -v -A FORWARD -i eth1 -o eth0 -d 10.0.0.0/8 -m state --state RELATED,ESTABLISHED -j ACCEPT;while true; do /gns3/bin/busybox sh;done`

Après avoir lancé tous les équipements : ROUTEUR, Controleur-SDN, 3 dhcp-clients, 1 dhcp-server, 3 openvswitchs; Lancer un traceroute 8.8.8.8 depuis la console d'un poste

dhcp-client. J'ai installé 3 sondes wireshark pour récupérer les trames.



Voici les trames que vous devriez obtenir pendant ce ping 8.8.8.8 :

1- Trames ICMP ECHO sur le lien vers Internet

Apply a display filter: <Ctrl>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------------|-----------------------|----------|--------|--|
| 6 | 9.984221 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 7 | 12.000165 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 8 | 13.004308 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 9 | 15.318124 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=1/256, ttl=62 (reply in 10) |
| 10 | 15.317438 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=1/256, ttl=115 (request in 9) |
| 11 | 16.000288 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 12 | 16.314581 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=2/512, ttl=62 (reply in 13) |
| 13 | 16.357008 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=2/512, ttl=115 (request in 13) |
| 14 | 17.315457 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=3/768, ttl=62 (reply in 15) |
| 15 | 17.357744 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=3/768, ttl=115 (request in 14) |
| 16 | 17.004308 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 17 | 18.315088 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=4/1024, ttl=62 (reply in 18) |
| 18 | 18.325187 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=4/1024, ttl=115 (request in 17) |
| 19 | 19.319125 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=5/1280, ttl=62 (reply in 20) |
| 20 | 19.328859 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=5/1280, ttl=115 (request in 19) |
| 21 | 20.000236 | c2:68:4c:22:00:47 | Spanning-tree-(for... | STP | 52 | Conf. Root = 32768/8/52:54:00:36:72:8b Cost = 0 Port = 0x0001 |
| 22 | 20.319678 | 192.168.122.3 | 8.8.8.8 | ICMP | 98 | Echo (ping) request id=0x0005, seq=6/1536, ttl=62 (reply in 23) |
| 23 | 20.329668 | 8.8.8.8 | 192.168.122.3 | ICMP | 98 | Echo (ping) reply id=0x0005, seq=6/1536, ttl=115 (request in 22) |

Frame 18: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ..., id 0
 Ethernet II, Src: RealtekU36:72:8b (52:54:00:36:72:8b), Dst: e6:38:73:17:08:92 (e6:38:73:17:08:92)
 Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.122.3
 Internet Control Message Protocol
 Type: 8 (Echo (ping) reply)
 Code: 0
 Checksum: 0x575c [correct]
 [Checksum Status: Good]
 Identifier (BE): 5 (0x0005)
 Sequence Number (BE): 1 (0x0001)
 Sequence Number (LE): 256 (0x0100)
 [Request frame: 5]
 [Response time: 7.388 ms]
 Timestamp from icmp data: Jan 14, 2024 22:06:48.000000000 CET
 [Timestamp from icmp data (relative): 0.001486000 seconds]
 Data (48 bytes)
 64180100000000000101112131415161718191a1b1c1d1e1f202122232425262728292a2b...

2- Trames LLDP +ECHO sur les liens entre les commutateursOpenVswitchs

Capture_interswitch.pcapng

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------------|-------------------|----------|--------|---|
| 23 | 36.324524 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=0/0, ttl=64 (reply in 24) |
| 24 | 36.325408 | 10.20.30.2 | 10.20.30.4 | ICMP | 102 | Echo (ping) reply id=0x4d00, seq=0/0, ttl=64 (request in 23) |
| 25 | 37.161695 | 00:00:00:00:00:11 | 2a:f7:1f:a6:20:b7 | ARP | 68 | Who has 10.20.30.3? Tell 10.20.30.254 |
| 26 | 37.161739 | 00:00:00:00:00:11 | 72:fi:da:2e:e6:35 | ARP | 68 | Who has 10.20.30.2? Tell 10.20.30.254 |
| 27 | 37.321907 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=1/256, ttl=64 (reply in 28) |
| 28 | 37.324766 | 10.20.30.2 | 10.20.30.4 | ICMP | 102 | Echo (ping) reply id=0x4d00, seq=1/256, ttl=64 (request in 27) |
| 29 | 38.001217 | 0e:00:00:00:00:01 | LLDP_Multicast | LLDP | 98 | MA/0e:00:00:00:00:01 IN/4 15 SysN=openvswitch-1 |
| 30 | 38.002900 | 0e:00:00:00:00:01 | LLDP_Multicast | LLDP | 98 | MA/0e:00:00:00:00:01 IN/4 15 SysN=openvswitch-3 |
| 31 | 38.322348 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=2/512, ttl=64 (reply in 32) |
| 32 | 38.325595 | 10.20.30.2 | 10.20.30.4 | ICMP | 102 | Echo (ping) reply id=0x4d00, seq=2/512, ttl=64 (request in 31) |
| 33 | 39.322547 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=3/768, ttl=64 (reply in 34) |
| 34 | 39.325646 | 10.20.30.2 | 10.20.30.4 | ICMP | 102 | Echo (ping) reply id=0x4d00, seq=3/768, ttl=64 (request in 33) |
| 35 | 40.322923 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=4/1024, ttl=64 (reply in 36) |
| 36 | 40.326124 | 10.20.30.2 | 10.20.30.4 | ICMP | 102 | Echo (ping) reply id=0x4d00, seq=4/1024, ttl=64 (request in 35) |
| 37 | 41.323230 | 10.20.30.4 | 10.20.30.2 | ICMP | 102 | Echo (ping) request id=0x4d00, seq=5/1280, ttl=64 (reply in 38) |

Frame 29: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
 Ethernet II, Src: 0e:00:00:00:00:01 (0e:00:00:00:00:01), Dst: LLDP_Multicast (01:80:c2:00:00:00)
 Destination: LLDP_Multicast (01:80:c2:00:00:00)
 Source: 0e:00:00:00:00:01 (0e:00:00:00:00:01)
 Type: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc)
 Trailer: 330000
 Link Layer Discovery Protocol
 Chassis Subtype = MAC address, Id: 0e:00:00:00:00:01
 Port Subtype = Interface name, Id: 4
 Time To Live = 15 sec
 System Name = openvswitch-1
 Port Description = Link openvswitch-1 - openvswitch-3
 Unknown - Unknown (1)
 Unknown - Unknown (2)
 End of LLDPDU

0000 01 80 c2 00 00 0e 00 00 00 01 88 cc 02 07
 0010 04 0e 00 00 00 01 04 02 05 34 06 02 00 0f 0a
 0020 0d 6f 70 65 6e 76 73 77 69 74 63 68 2d 31 08 22
 0030 4c 69 6e 6b 28 6f 70 65 6e 76 73 77 69 74 63 68
 0040 2d 31 28 2d 28 6f 70 65 6e 76 73 77 69 74 63 68
 0050 2d 33 fe 05 0e 00 00 01 31 fe 05 0e 00 00 02 33
 0060 00 00 1 3

Non-printable ASCII characters may only appear inside double-quotes. Packets: 92 - Displayed: 92 (100.0%) Profile: Default

3-Trames OpenFlow sur le lien du Contrôleur-SDN

Capture_openflow_contrôleur_SDN.pcapng

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|--------------|--------------|----------|--------|-------------------------|
| 1 | 0.000000 | 172.20.1.254 | 172.20.1.1 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 2 | 0.000004 | 172.20.1.254 | 172.20.1.2 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 3 | 0.000069 | 172.20.1.254 | 172.20.1.3 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 4 | 0.000576 | 172.20.1.1 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 6 | 0.000727 | 172.20.1.2 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 8 | 0.000863 | 172.20.1.3 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 10 | 3.001394 | 172.20.1.254 | 172.20.1.1 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 11 | 3.001531 | 172.20.1.254 | 172.20.1.2 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 12 | 3.001583 | 172.20.1.254 | 172.20.1.3 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 13 | 3.003821 | 172.20.1.1 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 14 | 3.003944 | 172.20.1.2 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 15 | 3.004150 | 172.20.1.3 | 172.20.1.254 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |
| 19 | 5.576003 | 172.20.1.254 | 172.20.1.1 | OpenFlow | 204 | Type: OFPT_PACKET_OUT |
| 20 | 5.576054 | 172.20.1.254 | 172.20.1.1 | OpenFlow | 204 | Type: OFPT_PACKET_OUT |
| 21 | 5.576068 | 172.20.1.254 | 172.20.1.2 | OpenFlow | 204 | Type: OFPT_PACKET_OUT |
| 22 | 5.576082 | 172.20.1.254 | 172.20.1.2 | OpenFlow | 204 | Type: OFPT_PACKET_OUT |
| 23 | 5.576093 | 172.20.1.254 | 172.20.1.3 | OpenFlow | 204 | Type: OFPT_PACKET_OUT |

Frame 20: 204 bytes on wire (1632 bits), 204 bytes captured (1632 bits) on interface -, id 0
 Ethernet II, Src: 12:68:f7:0b:63:13 (12:68:f7:0b:63:13), Dst: be:91:ef:18:aa:6e (be:91:ef:18:aa:6e)
 Internet Protocol Version 4, Src: 172.20.1.254, Dst: 172.20.1.1
 Transmission Control Protocol, Src Port: 6653, Dst Port: 37600, Seq: 153, Ack: 17, Len: 138
 OpenFlow 1.3
 Version: 1.3 (0x04)
 Type: OFPT_PACKET_OUT (13)
 Length: 138
 Transaction ID: 2378851936
 Buffer ID: OFP_NO_BUFFER (4294967295)
 In port: OFPP_CONTROLLER (4294967293)
 Actions length: 16
 Pad: 000000000000
 Action
 Type: OFPAT_OUTPUT (0)
 Length: 16
 Ports: 4
 Max length: 0
 Pad: 000000000000
 Data
 Ethernet II, Src: 0e:00:00:00:00:01 (0e:00:00:00:00:01), Dst: LLDP_Multicast (01:80:c2:00:00:00)
 Link Layer Discovery Protocol

0000 be 91 ef 18 aa 6e 12 68 f7 0b 63 13 00 00 45 00
 0010 00 0e 02 41 40 00 40 06 dc 41 ac 14 01 fe ac 14
 0020 01 01 19 fd 93 30 70 27 33 41 f6 78 ad 52 00 18
 0030 0c 09 c9 59 00 00 01 01 00 00 06 7c 02 0b 4d e9
 0040 ac 11 04 0d 00 8a 8d ca 66 68 ff ff ff ff ff ff
 0050 ff fd 00 10 00 00 00 00 00 00 00 00 00 00 00
 0060 00 04 00 00 00 00 00 00 00 01 88 cc 02 07 04 0e
 0070 0e 00 00 00 00 01 88 cc 02 07 04 0e 00 00 00 00
 0080 01 04 02 45 34 06 02 00 0f 0a 0d 6f 70 65 6e 76
 0090 72 77 69 74 63 68 2d 31 08 22 4c 69 6e 6b 28 6f
 00a0 70 65 6e 76 73 77 69 74 63 68 2d 31 20 2d 20 6f
 00b0 70 65 6e 76 73 77 69 74 63 68 2d 33 fe 05 0e 00
 00c0 00 01 31 fe 05 0e 00 00 02 33 00 00 1 3

Packets: 118 - Displayed: 118 (100.0%) Profile: Default