

# TP\_SDN

Night

April 2023

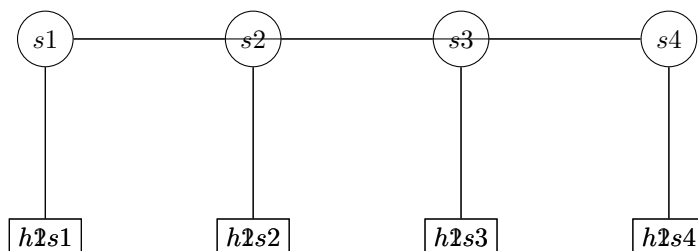
## 1 Ex1

```
ssh -CXY aurelien.mazaheri--petit@hpc.pedago.ensiee.fr
```

On lance dans mininet:

```
sudo mn --topo=linear,4,2 --controller=remote,ip=127.0.0.1:6633 --switch=ovsk,protocol=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1s1 h1s2 h1s3 h1s4 h2s1 h2s2 h2s3 h2s4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1s1, s1) (h1s2, s2) (h1s3, s3) (h1s4, s4) (h2s1, s1) (h2s2, s2) (h2s3, s3) (h2s4, s4)
*** Configuring hosts
h1s1 h1s2 h1s3 h1s4 h2s1 h2s2 h2s3 h2s4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
```

### 1.1 q1



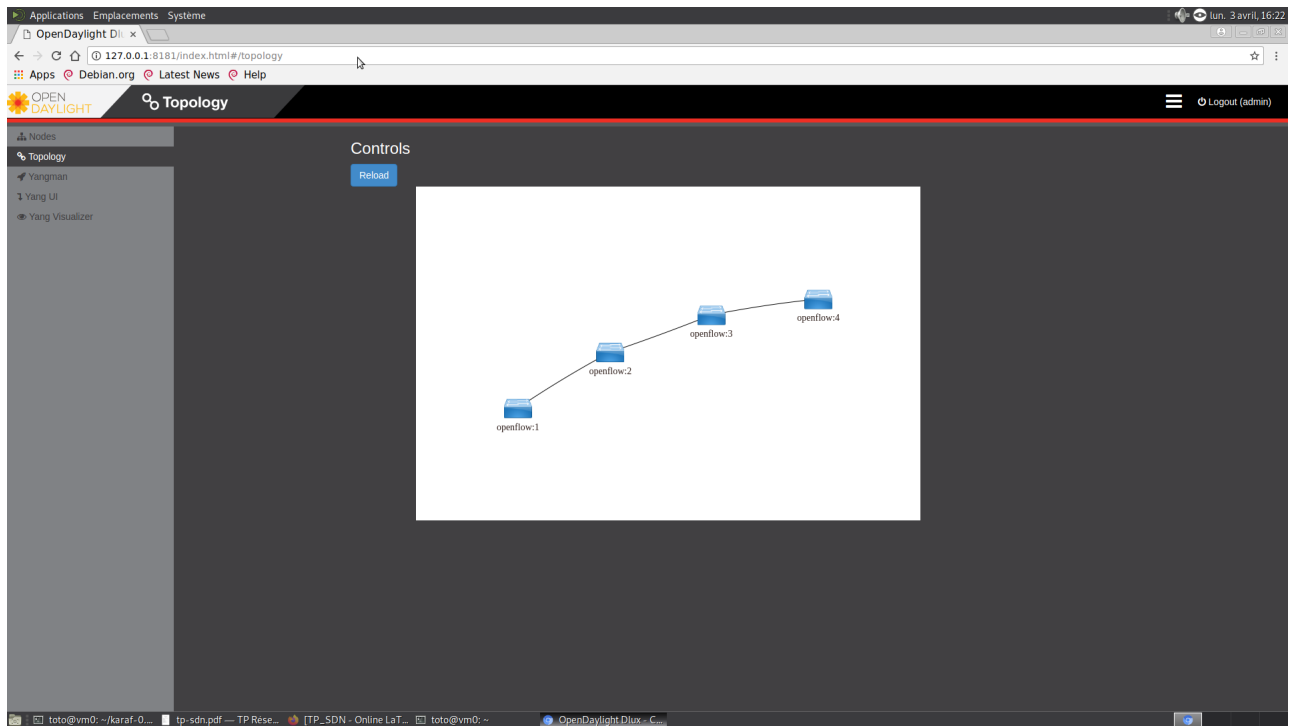
où :

- h1s1, h1s2, h1s3, h1s4 : les hôtes 1 connectés respectivement aux ports 1, 2, 3, 4 du switch s1.
- h2s1, h2s2, h2s3, h2s4 : les hôtes 2 connectés respectivement aux ports 1, 2, 3, 4 du switch s4.
- s1, s2, s3, s4 : les switches interconnectés.

## 2 Ex2 (partie 3)

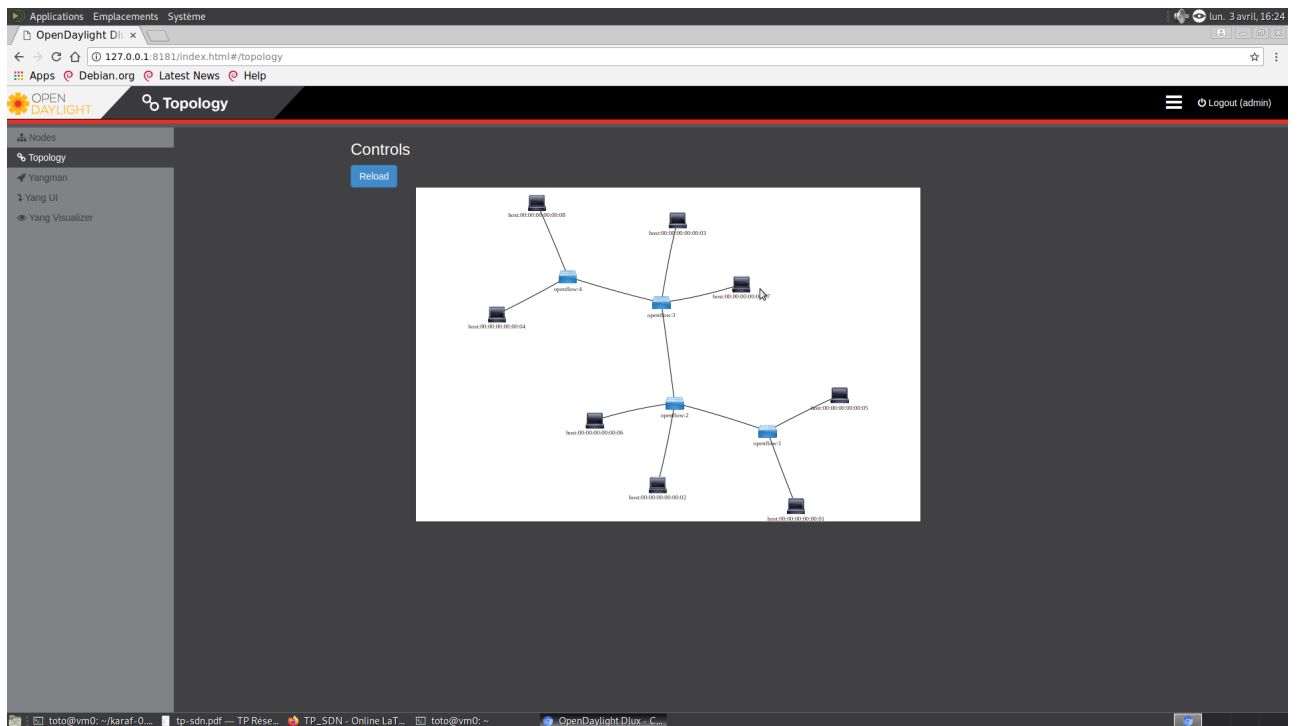
### 2.1

On observe que la topologie est erronée au premier lancement:



### 2.1.1

La topologie créée s'affiche en relançant avec :  
`pingall`



## 3 Ex3 (partie 4.2 postman)

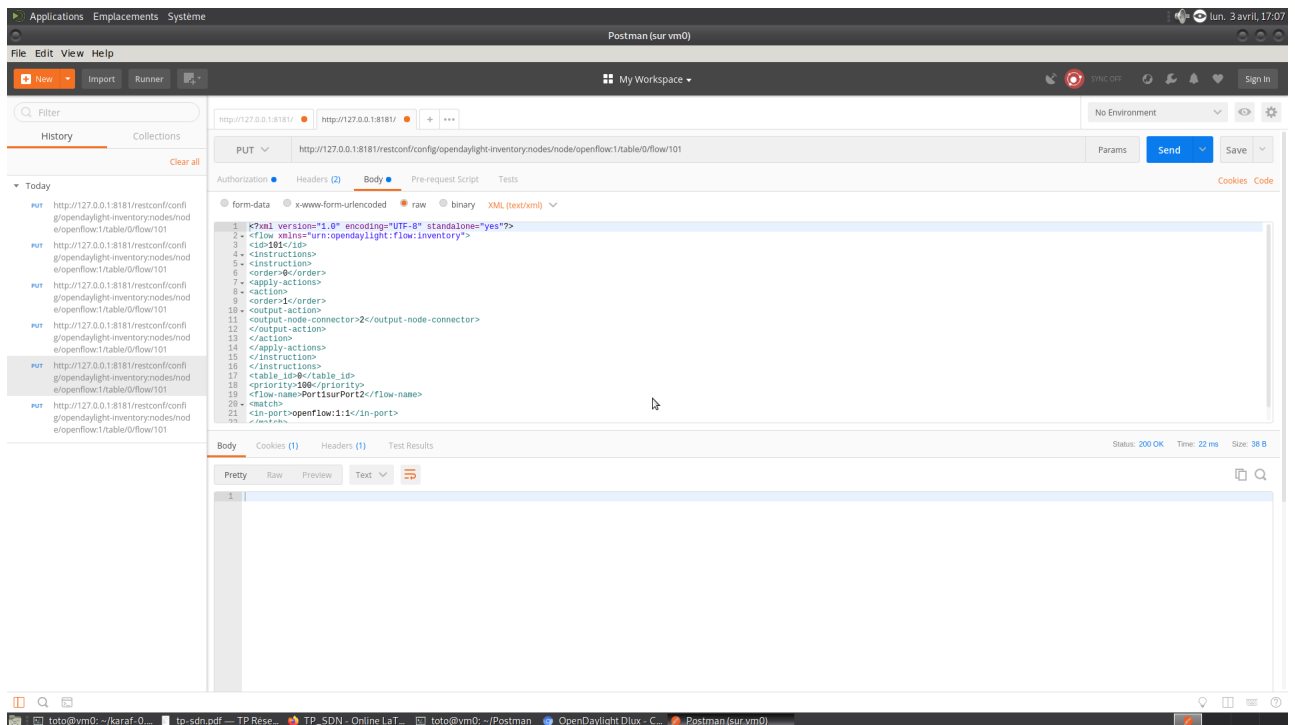
`cd /home/toto/Postman`

./Postman&

### 3.1

On configure l'authentification en Basic Auth (admin/admin) et on envoie le fichier XML suivant à l'url donnée:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flow xmlns="urn:opendaylight:flow:inventory">
<id>101</id>
<instructions>
<instruction>
<order>0</order>
<apply-actions>
<action>
<order>1</order>
<output-action>
<output-node-connector>2</output-node-connector>
</output-action>
</action>
</apply-actions>
</instruction>
</instructions>
<table_id>0</table_id>
<priority>100</priority>
<flow-name>Port1surPort2</flow-name>
<match>
<in-port>openflow:1:1</in-port>
</match>
</flow>
```



Un pingall nous assure que tout s'est bien passé:



```
<priority>2</priority>
<barrier>false</barrier>
</flow>
```

On observe bien que la règle a été appliquée : le texte `cookie=0x3`, `duration=11.211s`, `table=2`, `n_packets=0`, `n_bytes=0`, `idle_timeout=3400`, `hard_timeout=1200`, `priority=2`, `dl_src=00:00:00:00:00:04` `actions=drop` se trouve tout en bas lorsqu'on effectue la commande vérification. Le ping ne fonctionne pas :

```
mininet> h1s4 ping h1s1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.4 icmp_seq=1 Destination Host Unreachable
From 10.0.0.4 icmp_seq=2 Destination Host Unreachable
From 10.0.0.4 icmp_seq=3 Destination Host Unreachable
From 10.0.0.4 icmp_seq=4 Destination Host Unreachable
From 10.0.0.4 icmp_seq=5 Destination Host Unreachable
From 10.0.0.4 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6127ms
```

Pour modifier la priorité de la table, on remplace `<priority>2</priority>` par `<priority>10</priority>`. On observe bien que la règle a été appliquée : [...] `priority=10` [...], mais le ping ne fonctionne toujours pas. On remet la priorité à 2 puis on modifie la table à 0 : `<table_id>0</table_id>`. On le fait aussi dans l'URL : `/table/0/`. On lançant la commande vérification, on remarque que la ligne indiquant l'application de notre requête est désormais l'avant-dernière et non la dernière, toutefois les ping ne fonctionnent toujours pas. On remet ensuite la priorité à 10, sans modifier la table. La modification s'applique correctement et devient le second élément de la liste. Les ping ne fonctionnent toujours pas. On peut conclure de tout cela que la table et la priorité servent à déterminer l'application des règles et qu'il est important de faire attention aux deux en cas de problème

## 5 4.4 Flux bloquant par IP

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
<strict>false</strict>
<instructions>
<instruction>
<order>0</order>
<apply-actions>
<action>
<order>0</order>
<drop-action/>
</action>
</apply-actions>
</instruction>
</instructions>
<table_id>0</table_id>
<id>10</id>
<cookie_mask>255</cookie_mask>
<installHw>false</installHw>
<match>
<ethernet-match>
<ethernet-type>
<type>2048</type>
</ethernet-type>
</ethernet-match>
```

```

<ipv4-destination>10.0.0.2/32</ipv4-destination>
</match>
<hard-timeout>1200</hard-timeout>
<cookie>1</cookie>
<idle-timeout>3400</idle-timeout>
<flow-name>FooXf1</flow-name>
<priority>200</priority>
<barrier>false</barrier>
</flow>

```

Cette requête est correctement appliquée au switch 2 : on voit la ligne cookie=0x1, duration=70.996s, table=0, n\_packets=0, n\_bytes=0, idle\_timeout=3400, hard\_timeout=1200, priority=200, ip,nw\_dst=10.0.0.2 actions=dropen première position de la table. Il est devenu impossible de ping h1s2 :

```

mininet> h2s2 ping h1s2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
23 packets transmitted, 0 received, 100% packet loss, time 22517ms

```

## 6 4.5 Flux bloquant sur de multiples critères

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
<strict>false</strict>
<instructions>
<instruction>
<order>0</order>
<apply-actions>
<action>
<order>0</order>
<dec-nw-ttl/>
</action>
</apply-actions>
</instruction>
</instructions>
<table_id>2</table_id>
<id>131</id>
<cookie_mask>255</cookie_mask>
<match>
<ethernet-match>
<ethernet-type>
<type>2048</type>
</ethernet-type>
<ethernet-destination>
<address>ff:ff:29:01:19:61</address>
</ethernet-destination>
<ethernet-source>
<address>00:00:00:11:23:ae</address>
</ethernet-source>
</ethernet-match>
<ipv4-source>17.1.2.3/32</ipv4-source>
<ipv4-destination>172.168.5.6/32</ipv4-destination>
<ip-match>
<ip-protocol>6</ip-protocol>
<ip-dscp>2</ip-dscp>

```

```

<ip-ecn>2</ip-ecn>
</ip-match>
<tcp-source-port>25364</tcp-source-port>
<tcp-destination-port>8080</tcp-destination-port>
<in-port>0</in-port>
</match>
<hard-timeout>1200</hard-timeout>
<cookie>8</cookie>
<idle-timeout>3400</idle-timeout>
<flow-name>FooXf8</flow-name>
<priority>2</priority>
<barrier>false</barrier>
</flow>

```

Il m'a été impossible d'appliquer la requête, j'ai sans arrêt reçu l'erreur suivante sans en trouver l'origine

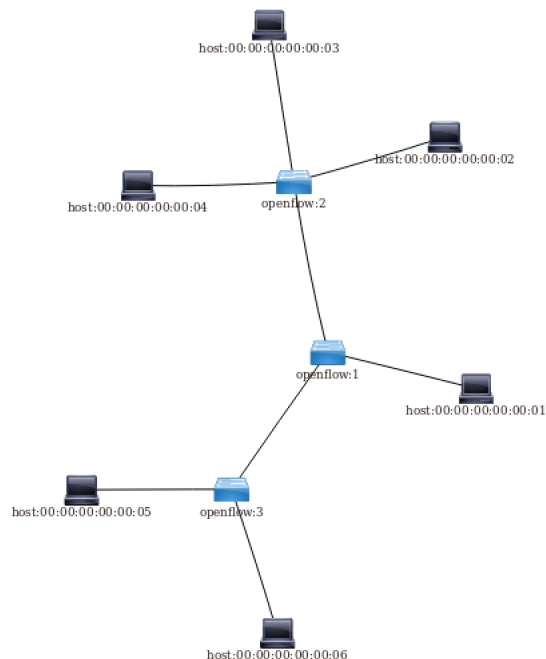
```

{
  "errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "malformed-message",
        "error-message":
        "Error parsing input: Schema for node with name adress and namespace urn:opendaylight
        "error-info": "[...] (coup car trop long)"
      }
    ]
  }
}

```

## 7 4.6 Nettoyage

Après le nettoyage des tables, on revient à l'état d'origine avec justes les règles de forwarding, le drop final en dernier et l'action CONTROLLER en premier.



## 8 5.1 Création d'un flux basé sur les ID VLANs

```
<flow xmlns="urn:opendaylight:flow:inventory">
<strict>false</strict>
<instructions>
<instruction>
<order>0</order>
<apply-actions>
<action>
<order>0</order>
<dec-nw-ttl/>
</action>
</apply-actions>
</instruction>
</instructions>
<table_id>2</table_id>
<id>138</id>
<cookie_mask>255</cookie_mask>
<match>
<ethernet-match>
<ethernet-type>
<type>2048</type>
</ethernet-type>
<ethernet-destination>
<address>00:00:00:00:00:01</adress>
</ethernet-destination>
<ethernet-source>
<address>00:00:00:00:00:03</adress>
</ethernet-source>
</ethernet-match>
<vlan-match>
<vlan-id>
<vlan-id>300</vlan-id>
<vlan-id-present>true</vlan-id-present>
</vlan-id>
</vlan-match>
</match>
<hard-timeout>1200</hard-timeout>
<cookie>15</cookie>
<idle-timeout>3400</idle-timeout>
<flow-name>FooXf15</flow-name>
<priority>2</priority>
<barrier>false</barrier>
</flow>
```

Il m'a été impossible d'exécuter la requête ci-dessus : j'ai de manière répétée obtenu l'erreur suivante

```
{
"errors": {
"error": [
{
"error-type": "protocol",
"error-tag": "malformed-message",
"error-message":
"Error parsing input: Schema for node with name adress and namespace urn:opendaylight
"error-info": "[...] (coup car trop long)"
}]
}
```



}

## 9 5.2 Création d'un flux basé sur les ID VLANs

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
  <strict>false</strict>
  <flow-name>FooXf108</flow-name>
  <id>101</id>
  <cookie_mask>255</cookie_mask>
  <cookie>108</cookie>
  <table_id>0</table_id>
  <priority>200</priority>
  <hard-timeout>1200</hard-timeout>
  <idle-timeout>3400</idle-timeout>
  <installHw>false</installHw>
  <instructions>
    <instruction>
      <order>0</order>
      <apply-actions>
        <action>
          <order>0</order>
          <output-action>
            <output-node-connector>FLOOD</output-node-connector>
            <max-length>60</max-length>
          </output-action>
        </action>
      </apply-actions>
    </instruction>
  </instructions>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
      <ethernet-destination>
        <address>00:00:00:00:00:02</address>
      </ethernet-destination>
      <ethernet-source>
        <address>00:00:00:00:00:04</address>
      </ethernet-source>
    </ethernet-match>
  </match>
</flow>
```