INSA **5ISS**
DGEI

TP1 : Introduction aux réseaux SDN/Openflow

Objectifs :

— Illustrer les principes de fonctionnement d'un réseau conforme au paradigme SDN/Openflow

---

1 Organisation du TP

Le réseau final à déployer sera constitué de 4 commutateurs Pica 8 compatibles Openflow et d'un contrôleur Openflow Ryu. A travers une série de petites manipulations, le but est de prendre en main le fonctionnement d'un réseau SDN/Openflow. Ce premier TP est organisé comme suit :

1. Mise en place d'une instance d'un bridge OVS (Open vSwitch) et illustration du fonctionnement par défaut d'un switch Openflow ;
2. Illustration de l'installation manuelle de règles Openflow dans les switches Openflow
3. Illustration de différents modes ("in-band" et "out-band") de rattachement du contrôleur aux commutateurs Openflow
4. Illustration du fonctionnement d'applications de contrôle simples pour la programmation du réseau en mode réactif et proactif

Vous serez donc répartis en 4 groupes d'étudiants. Chaque groupe aura en charge l'administration d'un switch Openflow.

2 Configuration de la salle au début du BE

Le matériel mis à votre disposition est le suivant :

— 4 commutateurs Pica 8 compatibles Openflow équipés de 48 ports Gigabits Ethernet

— Câbles Ethernet (paires torsadées non blindées avec connecteurs RJ45) croisés et non croisés

— 12 PCs de la salle GEI-101 équipés de plusieurs cartes Ethernet jouant le rôle de routeurs et/ou machines Linux.

Aucune configuration n'est mise en place sur les PC et routeurs. Tous les équipements sont physiquement déconnectés.

3 Questions

3.1 Fonctionnement par défaut d'un switch Openflow

L'objectif de cette première manipulation est de permettre à chaque groupe de créer une instance d'un switch OVS, de lui associer les ports physiques du commutateur auxquels sont rattachées leurs machines terminales puis d'illustrer le fonctionnement par défaut d'une instance d'un bridge (pont) OVS.

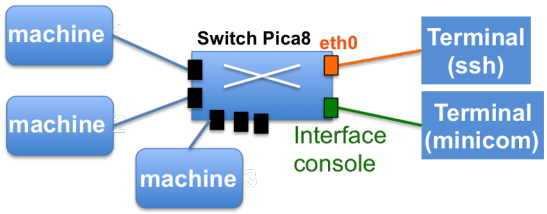

FIGURE 1 – configuration commutateur Pica8

1. Connectez les différentes machines de chaque rangée à un commutateur Pica 8 tel que présenté à la figure 3.1.
2. Via l'interface console du commutateur, rebooter le commutateur en lancant le script "**sudo picos_boot**", choisissez l'option "**2 - picOS : Open vSwitch/Openflow**" et fixez les adresses de l'interface de gestion (eth0) du switch et de sa passerelle par défaut à *192.168.1.10i/24 (i* : 1..3*) et 192.168.1.254/24*.
3. relancez les processus OVS en tapant **sudo service picos restart** ; Vérifiez que les processus *ovsdb-server, ovs-vswitchd* sont bien présents suite à "**ps -A**" ; Le switch OVS est opérationnel.

Ceci termine la phase d'initialisation. Chaque switch Pica8 peut être configuré depuis l'interface console ou depuis l'interface Ethernet de gestion via *ssh*.

L'étape suivante est de créer un instance d'un bridge logique et lui rattacher certains ports physiques du commutateur.

1. Créez une instance OVS nommée **br0** comme suit :

   ```
   admin@picOS-OVS$ ovs-vsctl add-br br0 -- set bridge br0 datapath_type=pica8
   ```

2. Attachez les ports physiques du commutateur au bridge nouvellement créé en vous basant sur la commande suivante (qui attache l'interface gigabit Ethernet ge-1/1/1 au bridge *br*0) :

   ```
   admin@picOS-OVS$ ovs-vsctl add-port br0 ge-1/1/1 -- set  interface ge-1/1/1
   type=pica8
   ```

3. Vérifiez la configuration en utilisant les commandes suivantes qui listent respectivement les configurations des bridges disponibles, l'état des interfaces rattachées au bridge *br*0 et les statistiques des différents ports [1]

   ```
   admin@picOS-OVS$ ovs-vsctl show
   admin@picOS-OVS$ ovs-ofctl show br0
   admin@picOS-OVS$ ovs-ofctl dump-ports br0
   ```

4. Vérifiez que vous disposez d'une connectivité IP entre les différentes machines de votre rangée. En consultant les entrées de la table des flux avec la commande suivante, une règle Openflow de faible priorité est par défaut installée avec comme action "normal" qui veut dire un acheminement équivalent à un pont de niveau 2.

   ```
   admin@picOS-OVS$ ovs-ofctl dump-flows br0
   ```

   En cas d'absence de la règle, la rajouter comme suit :

   ```
   admin@picOS-OVS$ ovs-ofctl add-flow br0 ,action=normal
   ```

3.2 Installation manuelle de règles Openflow dans les switches Openflow

L'objectif de cette deuxième manipulation est d'illustrer l'installation manuelle de règles Openflow. Nous considérerons la même topologie que la question précédente.

1. Dans un premier temps, annulez la règle Openflow par défaut en utilisant :

   ```
   admin@picOS-OVS$ ovs-ofctl del-flows br0
   ```

1. des ports peuvent être retirés, ou modifiés avec les commandes : ovs-vsctl del-port br0 <nom-port> et ovs-ofctl mod-port br0 <nom-port> action [up,down, packet-in, no-packet-in ..]

2. Installez les règles Openflow nécessaires pour permettre d'établir la connectivité IP entre les deux machines d'extrémité d'une même rangée. Vous vous baserez sur les règles Openflow qui permettent de diriger tout le trafic arrivant sur un port vers un autre port de sortie. Vous pouvez vous baser sur la commande suivante (qui dirige tout le trafic arrivant sur le port d'entrée 1 vers le port de sortie 2) :

   ```
   admin@picOS-OVS$ ovs-ofctl add-flow br0 in_port=1,actions=output:2
   ```

   Quelle est la limite d'une telle règle ?

3. Le niveau de granularité de la règle précédente est trop large pour permettre un contrôle précis du comportement du réseau. Nous utiliserons dans ce qui suit une règle capable de filtrer sur les adresses IP sources pour ne permettre de laisser passer que le trafic émanant de machines bien précises. Annulez la règle Openflow précédente et installez les nouvelles règles qui assurent la connectivité IP entre les machines d'extrémité selon le modèle suivant :

   ```
   admin@picOS-OVS$ ovs-ofctl del-flows br0
   admin@picOS-OVS$ ovs-ofctl add-flow br0 in_port=<port d'entrée>,arp,
   actions=output:<port de sortie>
   admin@picOS-OVS$ ovs-ofctl add-flow br0 in_port=<port d'entrée>,ip,
   nw_dst=<@IP destination>,actions=output:<port de sortie>
   ```

4. En vous référant à la figure 3.2 impliquant deux devices *D1, D2* et une Gateway *GW* matérialisés par les 3 machines d'une même rangée, identifiez puis installez les règles Openflow nécessaires pour permettre de :
   — laisser transiter du trafic UDP entre la GW et chaque device avec un numéro de port utilisé au niveau des Device égal à 5683 (par défaut utilisé par un serveur CoAP)
   — établir une session ssh depuis la GW vers chaque device.
   la règle de correspondance peut potentiellement vérifier les valeurs des adresses réseau (IP) source ou destination (reps. $nw\_src =< @IPsrce >$, $nw\_dst =< @IPdest >$) ou les numéro de port source ou destination (resp. $tp\_src =< numrodeportsource >$, $tp\_dst =< numrodeportdestination >$).

5. Testez les règles installées en vous assurant que le trafic attendu est bien pris en charge par le switch ? Vérifiez que tout autre trafic est bloqué et notamment l'absence d'une connectivité IP entre les deux devices et l'impossibilité d'établir une session ssh depuis chaque device vers la GW ?

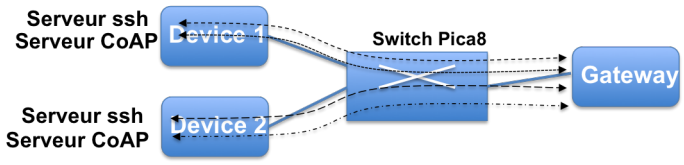

FIGURE 2 – Réseau à programmer

Jusqu'ici la programmation des commutateurs Pica8 a été effectuée manuellement et n'illustre pas l'aspect "Logiciel" (contrôle par logiciel) du paradigme SDN. Cet aspect est rendu possible par la présence d'un contrôleur SDN (logique) qui expose une API aux applications de contrôle réseau. Dans la suite de ce TP, le but est d'illustrer le rattachement d'un contrôleur au réseau et puis d'illustrer la programmation du réseau au travers d'applications de contrôle simples. Pour la suite du TP, nous utiliserons la topologie réseau de la figure 3.5.

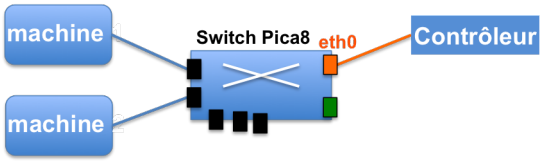

FIGURE 3 – Réseau avec contrôleur à déployer par rangée

3.3 Rattachement d'un contrôleur Openflow RYU

Nous considérerons un contrôle "hors-bande", en d'autres termes, le trafic Openflow entre le contrôleur et les switches Openflow est transporté par un réseau différent que le réseau utilisé pour le transport des données utilisateur. Pour ce faire nous connecterons les contrôleurs au réseau de gestion utilisé pour configurer les switches avec les sessions ssh. Pour ce faire,

1. Dans un premier temps, si ce n'est déjà fait, connectez une interface de la machine qui joue le rôle de contrôleur au réseau de gestion en lui affectant une adresse IP appropriée
2. Pour chaque bridge créé, déclarez le contrôleur auquel il est associé à l'aide de la commande suivante

```
admin@picOS-OVS$ ovs-vsctl set-controller br0 tcp:<@IP contrôleur>:6633
```

6633 étant le numéro de port par défaut sur lequel écoute le contrôleur RYU

3. lancer le contrôleur RYU en mode verbose

   ```
   controleur# ryu-manager --verbose
   ```

   Vérifiez l'état de la connexion avec le contrôleur, les échanges entre contrôleur et switch et le contenu de la table des flux à l'aide des commandes suivantes :

   ```
   admin@picOS-OVS$ovs-vsctl show
   admin@picOS-OVS$ovs-ofctl snoop br0
   admin@picOS-OVS$ovs-ofctl dump-flows br0
   ```

4. Pour le moment, aucune application de contrôle réseau n'a été lancée au niveau du contrôleur. En testant la connectivité IP entre les noeuds de votre rangée, vérifiez au niveau de la console du contrôleur que des événements relatifs à l'arrivée des paquets sont remontés au contrôleur sans que celui-ci ne propose un traitement.

3.4 Etude d'une application de contrôle réseau

La distribution RYU fournit un ensemble d'applications simples pour illustrer le développement d'applications de contrôle réseau. L'application **simple switch** cherche à rendre les commutateurs pilotés par le contrôleur comme des commutateurs classiques de niveau 2 (avec de l'auto-apprentissage pour la localisation des noeuds et de l'inondation sur toutes les interfaces lorsque le destinataire n'est pas connu). Ainsi, l'application **simple_switch** traite chaque paquet *"packet-in"* remonté au contrôleur par un bridge suite à l'arrivée d'un paquet dont le destinataire est inconnu, en demandant au bridge de relayer les paquets correspondants sur tous ses autres ports de sortie. Lorsque le destinataire transmet une trame en réponse (incluant son adresse MAC), l'application rajoute une entrée dans la table de flux qui dirige le flux de paquets depuis le port d'entrée initial vers le port de sortie depuis il est possible d'atteindre le destinataire.

1. lancer l'application **simple switch** [2] à l'aide

   ```
   controleur# ryu-manager -verbose simple_switch_14.py
   ```

2. Lancez un ping entre deux machines terminales de la rangée et vérifiez les entrées rajoutées dans la table des flux ?

2. localisée depuis le répertoire d'installation du contrôleur RYU dans */usr/local/lib/python*2.7*/dist − packages/ryu/app*

3. Etudiez le code source de *simple_switch_14.py* pour retrouver le comportement de l'application
4. Modifiez très simplement le code (en commentant très simplement certaines parties du code), pour faire en sorte que les trames d'un même flux de paquets à destination d'une machine donnée soient systématiquement renvoyées vers le contrôleur pour que ce dernier indique le chemin à suivre ? Mesurez l'impact sur la durée d'un aller-retour avec l'utilitaire *ping* ?

3.5 Exercice final

La figure 3.5 représente la topologie à déployer par chaque rangée en utilisant les 4 switches Pica8 disponibles. Chaque rangée aura à instancier un bridge avec une interface vers une machine de la rangée et qui sera connecté à deux autres bridges déployés sur deux autres switches Pica8 selon une topologie en maillage partiellement total. Les bridges de chaque rangée seront associés à un même contrôleur propre à chaque rangée. Le contrôleur peut être lancé sur une des machines terminales.

1. Dans un premier temps, ne connectez pas physiquement le lien en pointillé pour ne pas créer la boucle physique ?
2. Lancez l'application **simple_switch_14** et vérifiez la connectivité IP entre les machines terminales ?
3. Observez les règles Openflow installées par le contrôleur ?
4. Connectez le dernier lien, relancer l'application **simple_switch_14** puis vérifiez la connectivité entre machines terminales ? Qu'observez vous ? Quelle solution proposez-vous ? Consultez la liste des applications disponibles dans le répertoire d'installation du contrôleur et identifiez une application qui pourrait résoudre le problème.

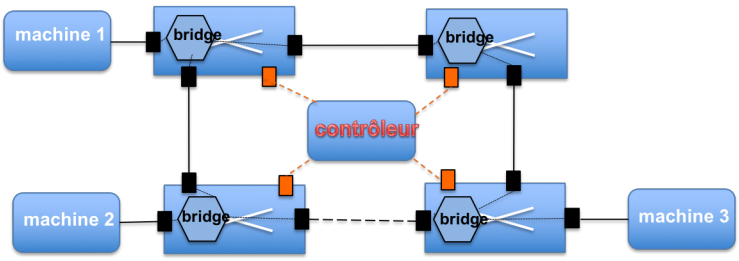

FIGURE 4 – Réseau à déployer par rangée