TRAVAUX PRATIQUES SD-WMN

André Cédric BESSALA

Septembre 2023

Supervision : Pr,Dr-Ing. Thomas DJOTIO NDIE
Mme Judith NOUHO

Table de matières

Installation des outils

L'OS considéré ici est ubuntu18.04 LTS sur machine virtuelle virtual-box.

1. Installation de Mininet-wifi
   sudo apt-get update
   sudo apt-get upgrade
   sudo apt-get install git
   git clone https://github.com/intrig-unicamp/mininet-wifi
   cd mininet-wifi
   sudo util/install.sh -WInfv
   Vérifier l'installation : sudo mn –wifi
2. Installation de Ryu
   git clone git://github.com/osrg/ryu.git
   cd ryu
   python ./setup.py install
   Vérifier l'installation: ryu-manager –version
3. Installation D-ITG
   sudo apt-get install d-itg

Architecture de Ryu

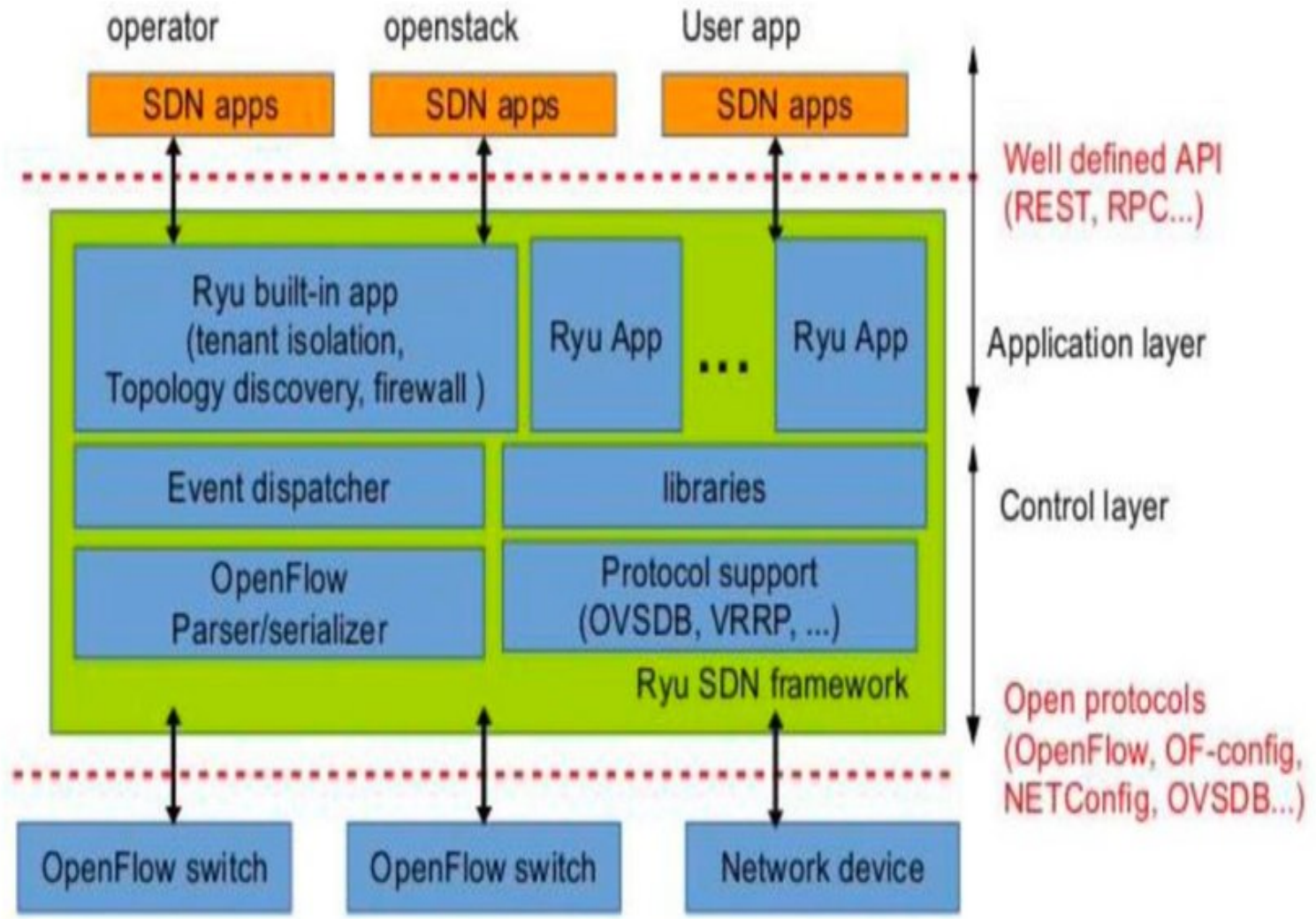

Figure: Arhitecture Ryu

Architecture de D-ITG

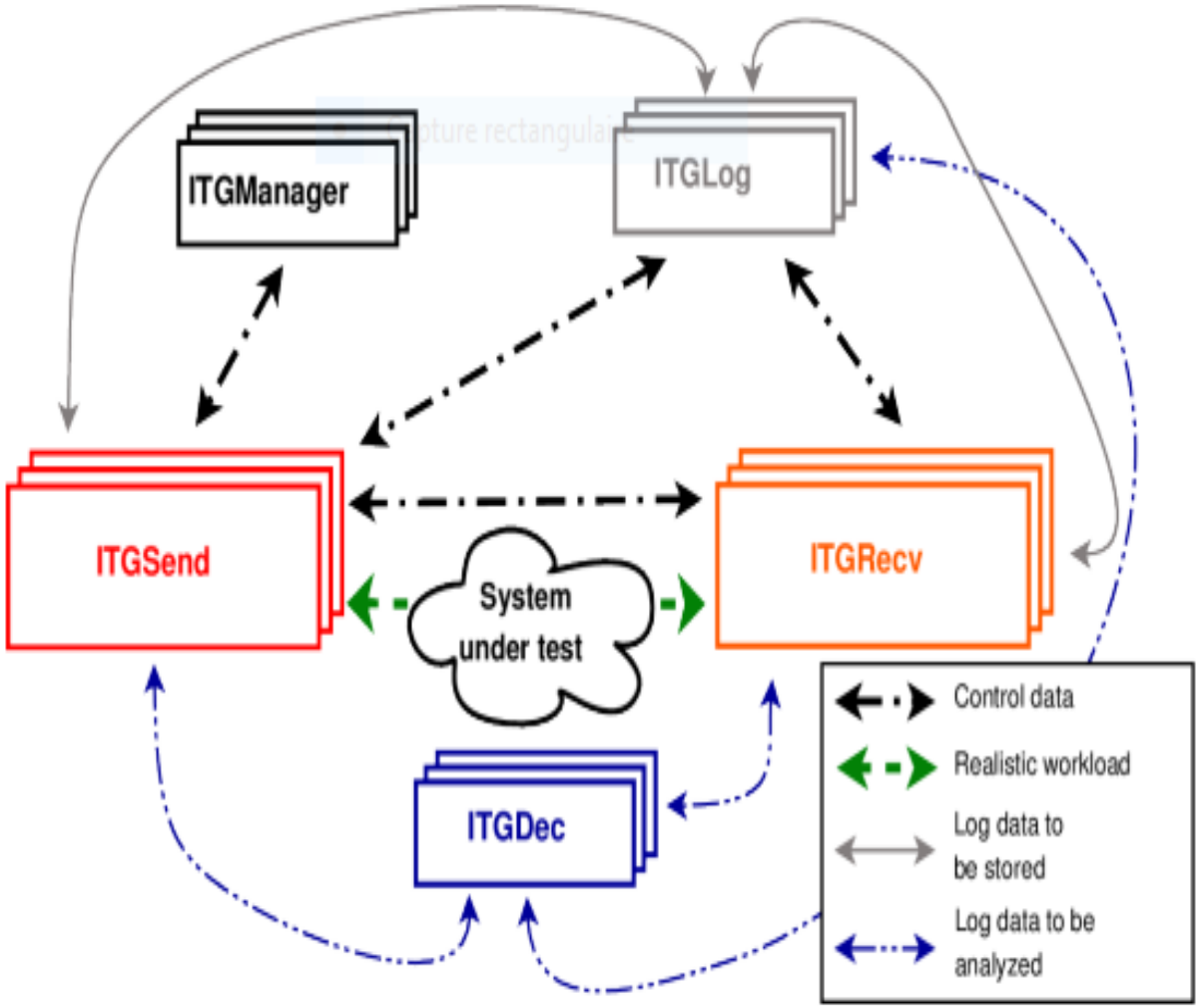

Figure: Architecture de D-ITG

Prise en main de Mininet-wifi

Vérifier si le service fonctionne : sudo service openvswitch-switch status

1. sudo mn & sudo mn –wifi
   -De quoi est composée la topologie par défaut de mininet (combien de switch openflow et combien d'hôtes) ?
   -À quoi servent les commandes pingall, iperf, xterm, ifconfig, dump, links et net de mininet-wifi ?
   -Les switchs de cette instance mininet son configurés en "learning switch" par défaut.
   .Qu'est qu'un "learning switch" ? Quel est le résultat de pingall lorsque le learning switch est utilisé ?
   .Quittons mininet-wifi et redémarrez en désactivant le contrôleur (–controller none). Quel est le résultat du pingall ? Quel semble donc être le rôle du contrôleur dans une architecture SDN ?

Définition de topologies customisées

La topologie utilisée jusqu'ici correspond à : mn –topo=tree,depth=1,fanout=2

1. Si l'on modifie maintenant cette topologie et que l'on crée une topologie d'une profondeur de 3 et d'un fanout de 4, combien y-a-t-il de switches ? Combien d'hôtes ? Combien de liens et enfin combien de contrôleurs ? A quel switch est relié l'hôte 25 ? Afficher cette topologie.
2. 3 APIs sont essentielles à la définition d'une topologie : addSwitch , addHost et addLink .
3. Créer un fichier python dans lequel vous allez grâce à ces différentes fonctions créer une topologie qui correspondra à la topologie décrite à la question 1.

Utilisation de Ryu dans son fonctionnement de base

- Dans le terminal 1, ryu run ryu/ryu/app/simple_switch_13.py
- Dans le terminal 2 démarer une topologie customisée, sudo python myapp.py
  1. Quel est le résultat d'un pingall ?
  2. Grâce à une commande vue précédemment, indiquez les liens entre les différentes interfaces (s1-eth1:h1-eth0, etc.).
  3. En modifiant votre fichier de topologie , supprimez le lien entre s1 et s2. Essayez à nouveau d'effectuer un pingall , que se passe-t-il ?

Openflow

Comme vous le savez, une architecture SDN est composée de trois couches principales : Application - Contrôle - Infrastructure.

1. Fonctionnement de switches traditionnels; Rappelez le fonctionnement des switches L2 traditionnels : -Existe-t-il une séparation entre le plan de contrôle et le plan des données ?
   -Quel type de données contient la "Forwarding Table" ? Quel type de données sont traitées au niveau 2 ?
   -Comment cette table est elle mise à jour ?
2. Switch SDN basés sur Openflow; listez les principaux messages qu'OpenFlow doit permettre d'échanger :Pensez à indiquer l'émetteur (contrôleur ou switch) et le destinataire (contrôleur ou switch) ainsi que leur raison d'être.

Openflow en pratique

1. Dans un terminal 1, Relancer un contrôleur Ryu avec un switch de niveau 2 : ryu-manager ryu/ryu/app/simple_switch_13.py
2. Dans un terminal 2, sudo mn –controller=remote –switch=ovsk,protocols=OpenFlow13 –topo=linear,6
3. Ce que nous allons maintenant vouloir faire est observer les échanges se produisant entre les différents switches, ainsi qu'entre les switches et le contrôleur. Pour ce faire nous allons lancer Wireshark et observer les échanges qui se produisent en local (interface loopback)
4. Lancez maintenant la commande pingall: Quel type de commandes OpenFlow sont capturées par wireshark, d'après la partie théorique quelle est leur rôle ?
5. Si vous relancez à nouveau la commande pingall, quelle différence observez vous avec la question précédente ? Pourquoi ?

OpenFlow en pratique

D'après ce qui précède, dire :

- Comment fonctionne donc ces switches SDN ? Quelle est la principale différence avec les switches traditionnels (legacy devices fonctionnant sans SDN) ?
- Quel type de données sont traitées ici par le "forwarding plane" (voir contenu packetIn et packetOut) ? Quel est le rôle du contrôleur ici ?

Remarque : En utilisant en ligne de commande l'outil ovs-ofctl il est également possible de superviser et de gérer les switches OpenvSwitch du réseau SDN.

OpenFlow en pratique et OpenvSwitch

1. Qu'est ce qu'un switch OpenvSwitch, et que peut on en faire ? A quoi servent les composants et outils ovs-vsctl, ovs-dpctl, ovsdb-server et ovs-ofctl ? (docs.openvswitch.org/en/latest/intro/what-is-ovs/)
2. Quelles informations permettent par exemple de récupérer les commandes suivantes ?:
   - sudo ovs-vsctl show
   - sudo ovs-ofctl -O OpenFlow13 show s1
   - sudo ovs-ofctl -O OpenFlow13 dump-flows s1
   - sudo ovs-appctl fdb/show s1
   - sudo ovs-dpctl dump-flows

Références

- https://learning.knetsolutions.in/docs/ryu/
- https://overlaid.net/2017/02/15/openflow-basic-concepts-and-theory/
- https://www.aussiebroadband.com.au/blog/difference-layer-3-layer-2-networks/
- https://osrg.github.io/ryu-book/en/html/openflow_protocol.html
- https://ryu.readthedocs.io/en/latest/ofproto_v1_3_ref.html