Télémétrie des équipements réseau

Jean-Marc Pouchoulon

Mai 2025

1 Objet du TP

Avant de commencer on va lister les outils et protocoles utilisés dans ce TP:

- *GNMI* est un protocole de communication pour la gestion des équipements réseau, basé sur *gRPC* et *Protobuf*. Il est utilisé pour la configuration, la surveillance et la collecte de données des équipements réseau. GNMI tire partie des templates *YANG* pour la modélisation des données, permettant ainsi une gestion plus efficace et standardisée des équipements.
- *OpenConfig* est une organisation qui promeut l'utilisation de modèles YANG ouverts pour la gestion des équipements réseaux.
- *Gnmic* est un client GNMI écrit en Go, développé par Nokia et qui permet d'interagir avec des équipements réseau compatibles GNMI . Il est possible de créer des clusters gnmic pour interroger plusieurs équipements en même temps.
- *gRPC* est un protocole d'appel de procédure distante développé par Google, qui utilise le protocole HTTP/2 pour la transmission de données. Il est utilisé pour la communication entre les services et les applications, permettant une communication efficace et rapide entre les différents composants d'une application distribuée.
- *Protobuf* est un format de sérialisation de données développé par Google, qui permet de définir des structures de données et de les sérialiser en un format binaire compact. Il est utilisé par gRPC.
- *pyenv* est un outil de gestion des versions de Python, qui permet d'installer et de gérer plusieurs versions de Python sur une même machine et pour un utilisateur donné. Il permet de créer des *venv* aussi.
- *Sflow* est un protocole de surveillance de réseau qui permet de collecter des données sur le trafic réseau et de les analyser pour détecter des anomalies ou des problèmes de performance.
- *NetLab* est un framework de test d'équipements réseau, qui permet de simuler des environnements réseau complexes et de tester des configurations et des comportements d'équipements réseau. Netlab génére des configurations réseaux pour un autre logiciel de simulation *containerlab* qui permet de créer des infrastructures réseau basées sur des conteneurs Docker.

2 Installation de l'environnement du TP

2.1 Installation de *pyenv* et de *Python 3.11*

```
curl -fsSL https://pyenv.run | bash
```

Modifiez votre .bashrc pour activer pyenv au démarrage de votre shell (à mettre avant fzf):

```
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - bash)"
eval "$(pyenv virtualenv-init -)"
```

```
source ~/.bashrc
```

Installez la version de Python 3.11 (*gnmic* ne fonctionne pas à la date de création du TP avec Python 3.12):

Tout d'abord, installez les dépendances nécessaires à la compilation de Python 3.11 :

```
sudo apt install -y make build-essential libssl-dev zlib1g-dev \
    libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm \
    libncurses5-dev libncursesw5-dev xz-utils tk-dev libffi-dev \
    liblzma-dev python3-openssl libssh-dev openvswitch-switch sshpass
```

Répérez la dernière version de Python 3.11 et installez-la :

```
pyenv install -l | grep 3.11
pyenv install 3.11.11
# creation d'un venv netlab
pyenv virtualenv 3.11 netlab
pyenv activate netlab
```

2.2 Installation de Netlab

Installer NetLab en suivant les instructions suivantes :

```
# upgrade de pip
pip install pip --upgrade
# installation de netlab et ansible dans le venv netlab
pip install ansible-pylibssh networklab[ALL] ansible
```

2.3 Installation de gnmic

Créez une directory "tpgnmic", placez-vous dans cette directory et installez gnmic :

```
mkdir tpgnmic
cd gnmic
bash -c "$(curl -sL https://get-gnmic.openconfig.net)"
```

3 Création d'une infrastructure containerisée de routeurs

On va générer un fichier topology.yml pour créer une infrastructure de test avec 2 routeurs Arista et 2 clients Linux.

```
cat << EOF > ./topology.yml
message:
  2 nodes arista avec BGP et OSPF et deux clients LINUX.
tools:
# graphe de la maquette voir http://127.0.0.1/graphite
 graphite:


module: [ ospf,bgp ]
defaults.device: eos
```

```
provider: clab

nodes:
 ceos1:
   image: registry.iutbeziers.fr/\begin{bashcode}
   ospf.area: 0
   bgp.as: 65100
   mgmt.ipv4: 192.168.121.101
 ceos2:
   image: registry.iutbeziers.fr/ceos:4.33.1F
   ospf.area: 0
   bgp.as: 65200
   mgmt.ipv4: 192.168.121.102
 linux1:
   device: linux
   image: registry.iutbeziers.fr/network-multitool:mai2025
   mgmt.ipv4: 192.168.121.103
 linux2:
   device: linux
   image: registry.iutbeziers.fr/network-multitool:mai2025
   mgmt.ipv4: 192.168.121.104
links:
   - ceos1-ceos2
   - ceos1-linux1
   - ceos2-linux2
EOF

# creation de l'infrastructure
netlab create topology.yml
# creation de l'infrastructure
netlab up topology.yml
netlab netlab report addressing
# connection au routeur containérisé Arista
netlab connect ceos1
# récupération des "capacités"
gnmic -a  192.168.121.102:6030 -u admin -p admin  --insecure capabilities
```

4 Utilisation de gnmic

4.1 Configuration de gnmic

```
  # Vérifier le bon fonctionnement en récupérant les "capacités" du routeur:
gnmic -a  192.168.121.102:6030 -u admin -p admin  --insecure capabilities
```

5 Utilisation de gnmic

1. Créez un fichier de configuration gnmic.yaml dans la directory projet afin d'éviter de resaisir les paramètres de connexion.

   Vous travaillerez sur routeur ceos1 pur les questions suivantes.
   1. Récupérer les capacités du routeur ceos1.
   2. Récupérer l'état de l'interface Ethernet1
   3. Récupérer les compteurs de paquets de l'interface Ethernet1
   4. Récupérez le "neighbor" BGP

5. Récupérez l'état de la session BGP
6. Récupérez toutes les informations de l'équipement.
7. Récupérez l'état de la mémoire de l'équipement.
8. Récupérez le nombre de CPU de l'équipement au format "flat"
9. Explorer l'API avec gnmic prompt
10. Souscrivez à l'état des compteurs de l'interface Ethernet1 en échantillonant toutes les 5 secondes.
11. Récupérer les routes BGP
12. Récupérer les processus de l'équipement.Quel est leur nombre?
13. Récupérer la description de l'interface Ethernet1, changez la en "effaçable" puis supprimez-la.
14. utiliser l'évenement on_change pour être prévenu de la supression de l'AS BGP.
15. Faite un "shut" de l'interface Ethernet1 et un "no shut".

6 Gnmi et Prometheus

Paramétrez gnmic.yaml et prometheus.yml afin que gnmic puisse être utilisé comme un exporter Prometheus.