

Cours Virtualisation des réseaux

Nantes Ynov Campus – 2022-2023

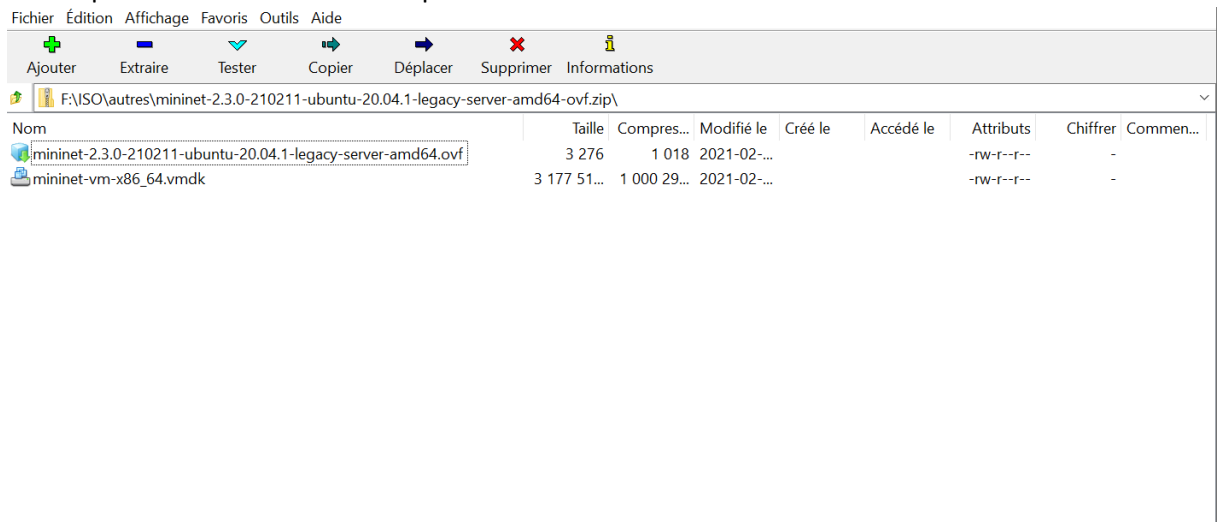
Activité Pratique 7

L'objectif de ce TP est de commencer à vous familiariser avec la plateforme Mininet. Mininet est un outil qui permet de déployer un réseau, composé de machines virtuelles (VMs par ses sicles en anglais) ou réelles, exécuter le vrai code des machines, utiliser des vrai commutateurs matériel ou logiciel, à l'aide d'un nombre réduit de commandes systèmes.

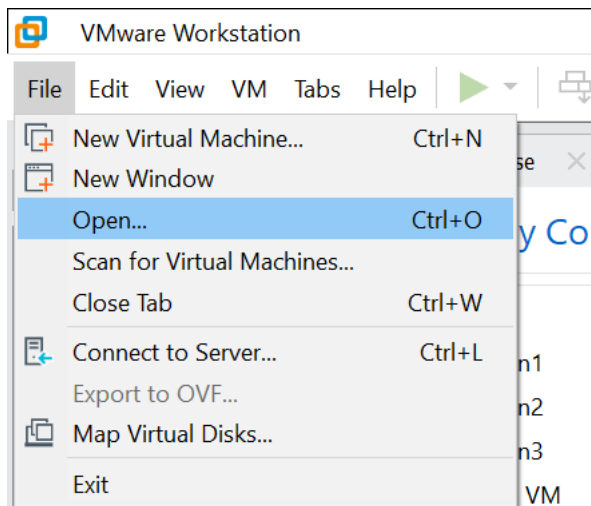
Lorsque le réseau déployés est composé uniquement par de VMs et de commutateurs logiciel (des switches SDN), elle pourrait être déployée à l'intérieur d'une seule machine (un laptop par exemple). Mininet utilise des technologies très avancés : Linux Containers et commutateurs SDN.

Mise en place de la plateforme Mininet

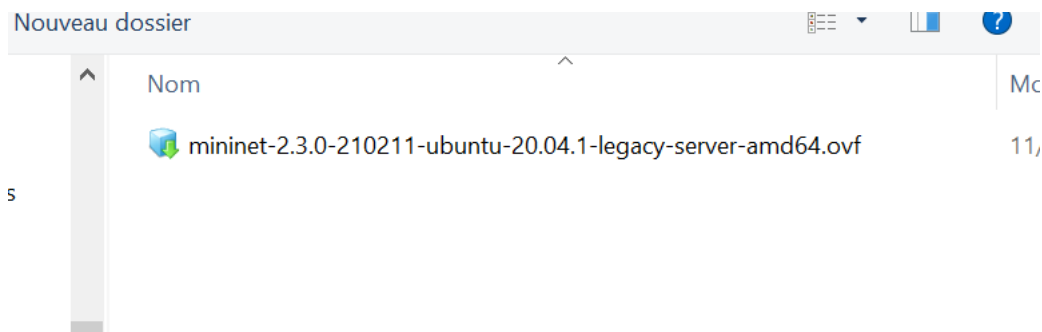
1. Téléchargez la VM mininet : mininet-2.3.0-210211-ubuntu-20.04.1-legacy-server-amd64-ovf.zip depuis le répertoire des ISO mis à disposition
2. Décompressez le fichier dans un répertoire de votre choix



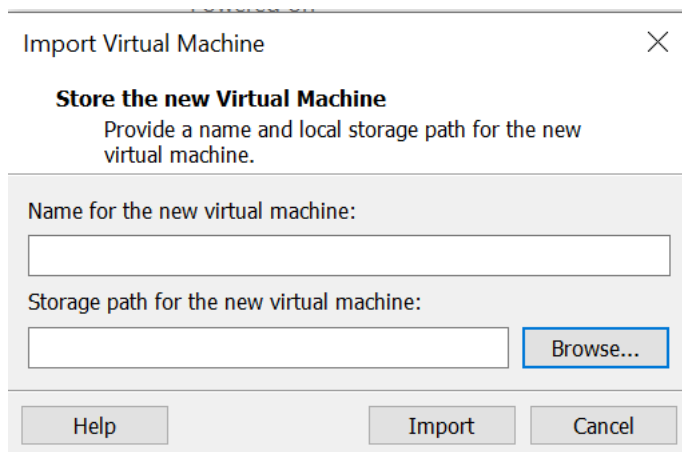
3. Démarrez votre hyperviseur pour importer la VM



4. Sélectionnez le fichier .ovf que vous venez de décompresser



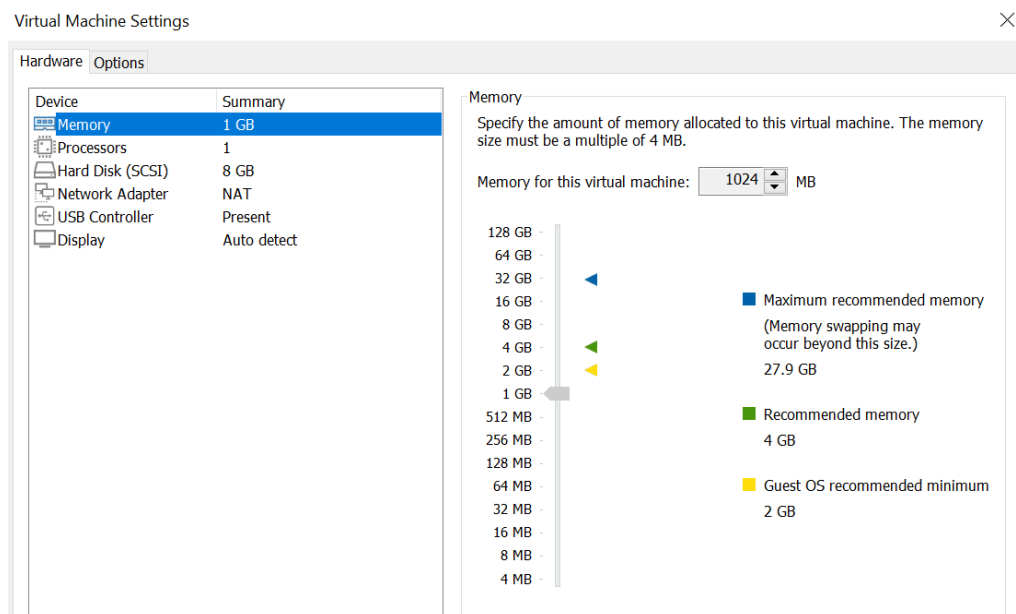
5. Cliquez sur suivant saisir le nom de la nouvelle VM et sélectionner l'emplacement de stockage puis cliquer sur importer



6. L'importation devrait se lancer pour insérer votre machine



7. A la fin de l'installation une nouvelle machine devrait apparaître dans votre inventaire
8. La VM créée ne possède pas d'environnement graphique (aucun Window Manager n'est installé). Cependant, des logiciels utilisant le serveur X sont présentes (e.g. xterm, firefox, wireshark) et que vous utiliserez probablement plus tard dans ce cours. Comment allez-vous faire donc pour utiliser ces applications ? La solution consistera à utiliser la VM par ssh et connecter votre VM à un réseau « Réseau privé hôte ».



9. Démarrer la machine virtuelle



10. Une fois la machine démarrée connectez-vous en utilisant les identifiants :

User : mininet

Password : mininet

11. Récupérer l'adresse IP attribuée à la machine à l'aide de la commande ifconfig

12. La VM devrait être sur le même réseau que votre machine physique si ce n'est pas le cas vérifiez vos paramètres réseaux comme indiqué au point 10

13. Afin de se connecter en ssh à la VM nous allons utiliser putty que vous pouvez trouver à cette adresse : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Téléchargez la version exécutable :

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download them directly from the links below. (Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(signature)

pscp.exe (an SCP client, i.e. command-line secure file copy)

32-bit:	pscp.exe	(or by FTP)	(signature)
64-bit:	pscp.exe	(or by FTP)	(signature)

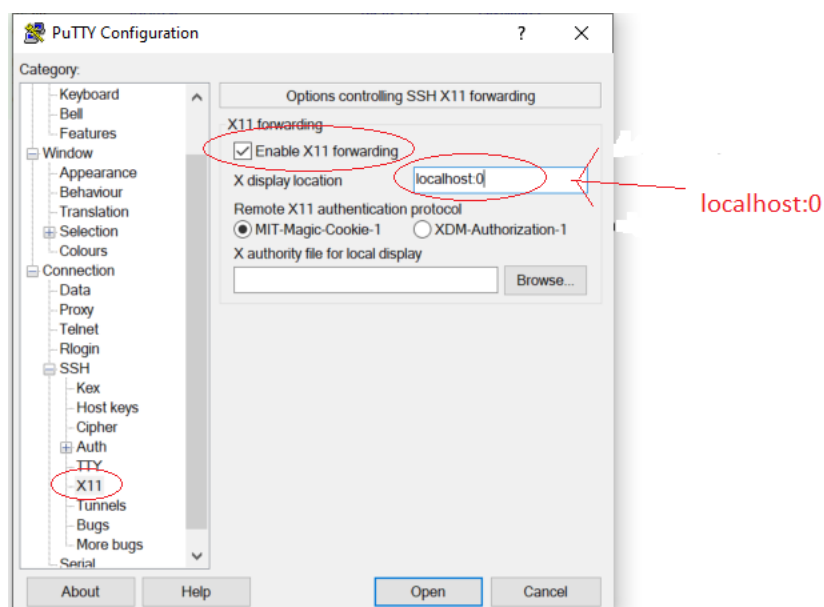
psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)

32-bit:	psftp.exe	(or by FTP)	(signature)
64-bit:	psftp.exe	(or by FTP)	(signature)

14. Afin de pouvoir lancer les applications graphiques installées sur la VM, télécharger et installer Xming que vous pouvez trouver à cette adresse :

<https://sourceforge.net/projects/xming/files/latest/download>

15. Une fois les applications installées, lancer putty en le configurant comme suit et en saisissant en hostname l'adresse ip de la VM :

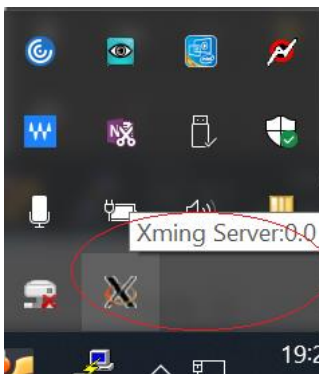


16. Saisir les mêmes identifiants que sur la VM mininet/mininet vous devriez arriver sur l'invite ci-dessous :

```
mininet@mininet-vm: ~  
login as: mininet  
mininet@192.168.110.128's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
New release '22.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Wed Feb 15 20:58:26 2023 from 192.168.110.1  
mininet@mininet-vm:~$
```

Test de fonctionnement

Dans l'invite de commande saisir : `sudo wireshark &` cela devrait lancer automatiquement le chargement de l'application wireshark dans une nouvelle fenêtre. Si ce n'est pas le cas vérifier que Xming est bien lancé dans votre barre d'adresse en bas à droite :



Premier contact avec Mininet

Premièrement, retenez bien que vous devez exécuter Mininet avec les droits d'administrateur (i.e. avec la commande `sudo`).

\$ sudo mn

Si tout s'est bien passé, à la fin vous devez voir l'invite de commande Mininet (le mode CLI de Mininet) « mininet> », et une topologie réseau devrait être créée. La topologie par défaut est appelé minimal. Toutes les commandes disponibles dans la CLI mininet peuvent être listé avec la commande « help »

```
mininet> help
```

Les clients qui ont été créés dans notre réseau virtuel peuvent être listés avec la commande « nodes »

```
mininet> nodes
```

Par convention, les clients du réseau sont appelés h1, h2, ... hX. Les switches (commutateurs) sont appelés s1, s2, ... sX, et, à oublier pour l'instant, les contrôleurs des switches sont appelés c0, c1, ... cX.

1 - Combien de switches et clients sont disponibles dans votre réseau virtuel ?

Les informations liées à chaque nœud créé peuvent être listé avec la commande « dump ».

2 - Décrivez ce que vous obtenez comme information des switches et clients du réseau.

Les liens créés peuvent être listés avec la commande « net »

3 - Dessinez la topologie créée.

La puissance de Mininet réside, entre autres, sur le fait que les clients sont des images de la machine hôte (là où Mininet est installé) et ils ont accès à l'espace système de son utilisateur. Il est donc possible d'exécuter « à l'intérieur » des clients plusieurs commandes disponibles dans l'OS. Pour cela, vous devez précéder la commande par le nom de la machine virtuel qui l'exécutera. Par exemple, pour voir la configuration des interfaces réseaux disponibles dans le client h1, on exécute

```
mininet> h1 ifconfig -a
```

4 - Ajoutez à votre dessin de la topologie réseau les adresses IP associées à chaque client, ainsi que le masque de réseau

5 - Le client h2 peut être atteint par le client h1 ? Vérifiez-le par la commande ping. Donnez l'instruction complète que vous avez utilisée

Pour sortir de mininet, utilisez la commande « exit ». Important : si lors d'un test, la topologie n'a pas été créée correctement, ou si Mininet a finis avec un code d'erreur, vous devez le « nettoyer » avec la commande :

```
$ sudo mn -c
```

Pour éviter d'indiquer à chaque fois qu'elle machine doit exécuter quelle commande, et si vous voulez travailler avec une console proche de celle des machines réelles, vous pouvez

1. Exécuter Mininet avec l'option « -x »
2. Exécuter la commande xterm et indiquer le client à attacher. mininet> xterm h1

La première option ouvrira automatiquement une fenêtre xterm pour chaque client, switch et contrôleur présent dans la topologie.

D'autres topologies réseaux

Mininet fournit, en plus de la topologie « minimal », la topologie « single », « linear » et « tree ».
Pour charger l'une de ces topologies, utilisez l'option « --topo ». Par exemple :

```
$ sudo mn --topo single
```

« single » tout court donne la même topologie que minimal, mais on peut ajouter également comme argument un chiffre, qui indique le nombre de clients à créer. Par exemple single,3.

6 - Exécutez les commandes suivantes et dessinez la topologie créée :

- \$ sudo mn --topo linear

- \$ sudo mn --topo linear,3

- \$ sudo mn --topo tree

- \$ sudo mn --topo tree,2

- \$ sudo mn --topo tree,2,fanout=3. À quoi servent les paramètres 2 et fanout=3 ?

Pour aller encore plus loin

Et si on souhaitait travailler avec une topologie autre que tree, linear ou simple ? Bien, dans ce cas-là, il faut créer une topologie personnalisée à l'aide de l'API Python de Mininet. Supposez que vous voulez créer un client h1 connecté directement au client h2 et ce dernier connecté à un switch s1. Voici le code Python à écrire

```
from mininet.topo import Topo

class Test_Topo(Topo):
    def __init__(self):
        "Create P2P topology"
        # Initialiser la topology
        Topo.__init__(self)
        # Ajouter les notes et les switches
        h1 = self.addNode('h1')
        h2 = self.addNode('h2')
        s1 = self.addSwitch('s1')
        # Ajouter les liaisons
        self.addLink(h2, s1)
        self.addLink(h1, h2)
    topos = {
        'toptest': (lambda: Test_Topo())
    }
```

Sauvegardez dans un fichier toptest.py le code précédent et envoyez-le sur la VM mininet, dans le répertoire /home/mininet/mytopos/ (répertoire à créer s'il n'existe pas). Ensuite exécutez-le avec la commande

```
$ sudo mn --custom /home/mininet/mytopos/toptest.py --topo toptest
```

7 - Si le réseau est créé correctement ouvrir un terminal pour chaque client et explorez la configuration des cartes réseaux. Combien d'interface disposent h1 et h2 ? Quels sont les adresses assignées ? Configurez par le CLI l'interface réseau du client h1 et l'interface du client h2 qui est connecté à h1. Assignez l'adresse 192.168.0.1 et 192.168.0.2 avec un netmask 255.255.255.0.

Comment fait-on pour configurer une carte réseau en ligne de commande ? On utilise la commande `ifconfig` (voir <http://www.tecmint.com/ifconfig-command-examples/>) Vérifiez par la commande `ping` la connectivité entre les machines `h1` et `h2`.

Et pour prouver que vous avez tout compris, créer la topologie suivante (dans un fichier `trois_sw.py`) et vérifiez la connectivité par la commande `ping` : `h1 ----- s1 ----- s2 ----- s3 ----- h2`

Et c'est presque la fin...

Pouvoir créer un réseau personnalisé est une bonne chose, mais devoir le configurer à la main à chaque utilisation, n'est pas très confortable. Pour résoudre ce problème-là, on utilisera encore l'API Python. Le code suivant assigne une adresse IP à l'interface `h2-eth1` de `h2` et exécute un `ping` depuis la machine `h1` vers la machine `h2`. Comme vous pouvez l'apprécier, pour exécuter une commande quelconque sur une machine avec l'API, on fait appel à la méthode `cmd()` et on donne comme argument la commande à exécuter sous la forme d'une chaîne de caractères.

```
from mininet.net import Mininet
from mininet.topo import Topo
from mininet.node import Node
from mininet.cli import CLI
from mininet.link import TCLink
from mininet.log import setLogLevel

class Test_Topo(Topo):
    def __init__(self):
        "Create P2P topology"

        # Initialize topology
        Topo.__init__(self)

        # Add hosts and switches
        h1 = self.addNode('h1')
        h2 = self.addNode('h2')
        s1 = self.addSwitch('s1')

        # Add links
        self.addLink(h2, s1)
        self.addLink(h1, h2)

topos = {
    'toptest': (lambda: Test_Topo())
}

def topTest():
    topo = Test_Topo()
    net = Mininet(topo=topo, link=TCLink)
    net.start()
    h1 = net.get('h1')
    h2 = net.get('h2')
    h1int0 = h1.intf('h1-eth0')
    h2int1 = h2.intf('h2-eth1')
    h1.setIP('11.0.0.1', 8, h1int0)
    h2.setIP('11.0.0.3', 8, h2int1)
    CLI(net) # sans cette ligne, on ne verrait jamais le CLI
    net.stop() # ne pas oublier de détruire le réseau

if __name__ == '__main__':
    setLogLevel('info')
    topTest()
```

Cependant, notez bien que pour exécuter l'ensemble du code ajouté, on fera appel à Python cette fois-ci.

```
$ sudo python /home/mininet/mytopos/toptest.py
```

Note : Si lors de l'exécution de code Python, vous tombez sur un message qui indique que le contrôleur n'a pas pu être contacté, créez un lien, appelé « controller » vers la commande `ovs-controller` (e.g. `ln -s /usr/bin/ovs-controller /usr/bin/controller`).

8 - Exécutez le code présenté ci-dessus. Est-ce que le client h1 arrive à atteindre le client h2 ? Si ce n'est pas le cas, en passant par le CLI faites les modifications nécessaires pour que la commande ping réussisse. Enfin, si vous avez trouvé la solution, modifiez le code pour implémenter les commandes que vous avez exécutées par le CLI.

Faites de même avec le code `trois_sw.py` : Ajoutez des adresses IP de type 192.168.1.X et exécutez un ping entre les 2 clients du réseau en passant par l'API Python

Pour récupérer vos fichiers (`toptest.py` et `trois_sw.py`) à la fin du TP connectez-vous à la VM mininet avec filezilla pour récupérer vos fichiers en local via sftp.

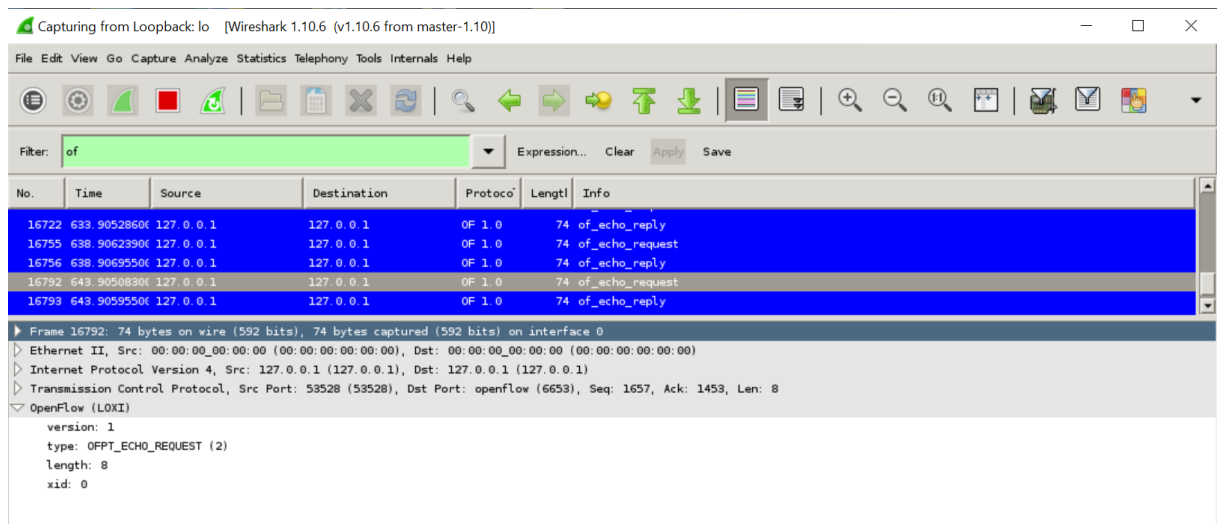
ANNEXE

Wireshark est un analyseur de paquets open source (GNU) populaire. Ses “dissectors” ou décodeurs de protocoles permettent d'interpréter le trafic du réseau.

Fonctionnalités de Wireshark

Les fonctionnalités principales de Wireshark sont :

- Disponibles pour les systèmes UNIX et Windows.
- Capturer les paquets de données en “live” qui passent en live sur les interfaces à partir de n'importe quel type de supports : Ethernet, Wi-Fi, Bluetooth, Frame-Relay, ATM, HDLC, USB, ... Voir NetworkMedia.
- Ouvrir des fichiers de captures de paquets réalisés avec tcpdump/WinDump, Wireshark et bien d'autres programmes.
- Importer des paquets venant de fichiers texte contenant les charges en hexa de paquets de données.
- Display packets with very detailed protocol information.
- Enregistrer des paquets de données capturés.
- Exporter certains ou tous les paquets capturés dans différents formats.
- Filtrer les paquets sur base de différents critères.
- Rechercher des paquets sur base de différents critères.
- Coloriser des paquets sur base de différents critères.
- Créer différentes statistiques



Quelques commandes utiles Mininet

- `mininet> nodes` - Affiche les noeuds dans le réseau. h correspond à un host, s à un switch et c à un contrôleur.
- `mininet> net` - Affiche la topologie du réseau. Quelle machine est connectée à quel host et le nom de l'interface réseau de connexion.
- `mininet> dump` - Affiche les adresses IP de chaque machine ainsi que le nom de la carte réseau.
- `mininet> h1 ping h2` - Demande à l'hôte h1 d'effectuer un ping sur l'hôte h2.
- `mininet> h1 ifconfig -a` - Effectue et affiche les résultats de la commande ifconfig sur la machine souhaitée.
- `mininet> pingall` - Permet de tester la connectivité du réseau. Toutes les machines vont se ping entre elles.
- `mininet> link s1 h1 down` - Enlève le lien réseau entre s1 et h1.
- `mininet> link s1 h1 up` - Remet le lien réseau entre s1 et h1.
- `mininet> xterm h1` - Ouvre une fenêtre en ligne de commande sur l'hôte h1.

D'autres commandes et informations sont disponibles à l'adresse :

https://github.com/Hoopaa/tutoriel_mininet