Test des Outils

Dino Lopez Pacheco
dino.lopez@univ-cotedazur.fr

1 Introduction

Souvent, dans le passé, le réseau était étudié grâce aux simulateurs, qui s'appuyaient sur des implémentations basiques des protocoles (e.g. une implémentation de certaines composantes du protocole TCP) et de modèles mathématiques simulant des dispositifs physiques (e.g. simulation de la propagation des ondes radios dans l'air). Bien que des logicielles légers et pratiques, les simulateurs fournissent une vision idéale des réseaux, plus ou moins éloignés de la réalité.

Pour une étude proche de la réalité, il fallait utiliser des vrais infrastructure réseaux physiques. Cependant, cette approche est souvent irréaliste dans un cadre pédagogique composé d'un groupe nombreux de personnes ou dans un cadre d'étude des réseaux à large échelle.

Actuellement, l'étude des réseaux via des émulateurs est préféré. Un émulateur déploie par le bais des mécanismes de virtualisation légère une infrastructure réseaux au plus proche de la réalité, car ces équipements exécutent le vrai code et vrais protocoles d'une infrastructure physique. Sans compter que de plus, aujourd'hui, les grands centres de données (pour les services cloud), emploient les mêmes technologies virtuelles sur leur infrastructure physiques (sous la forme d'un réseau *overlay*).

Dans ce cours, nous allons utiliser l'émulateur réseau *Common Open Research Emulator* - CORE (https://coreemu.github.io/core/), qui tourne sur des systèmes Linux. Nous allons explorer cet outil aujourd'hui et tester son bon fonctionnement. Pour les TPs à venir, il est nécessaire de ne pas avoir de problèmes avec votre système de travail afin de pouvoir vous focaliser sur les sujets traités en cours.

2 Installation &Configuration d'une VM Linux

L'installation de CORE étant un processus qui peut être compliqué pour certain·e·s, nous avons installé CORE dans la VM Linux que vous utiliserez pour vos cours à Polytech.

Vous devez donc installer la VM fournie par le département. De plus, notez que l'exploration d'un réseau partagé par plusieurs utilisateurs est compliquée, au vu de la grande quantité d'information qui y transitent. Travailler dans une VM (qui lui tourne dans un réseau privé VirtualBox) vous évitera le tri d'une grande quantité de donnés.

2.1 Installation de VirtualBox et la VM Linux

Faites l'installation en suivant les instructions de l'onglet « Installation VM » du site moodle du cours.

1. Validez l'installation de la VM avec votre encadrant de TP

2.2 Installation de *Guest Additions*

Pour obtenir un écran de taille convenable (entre autres services) il faut installer les modules *Guest Additions* dans votre VM.

Nous avons installé Guest Additions dans la VM. Cependant, si la VM a été configuré dans un environnement différent à celui que vous avez dans votre laptop, il est possible que Guest Additions soit désactivé et que sa réinstallation soit nécessaire.

Allumez votre VM. Puis ouvrez le terminal et exécutez la commande

```
$ lsmod | grep vbox
```

Si vous voyez les 2 lignes suivantes (les chiffres peuvent différer), ce que les Guest Additions fonctionnent correctement.

```
vboxvideo              36864        0
vboxguest             364544        5
```

Si ce n'est pas le cas, vous devez réinstaller les Guest Additions. Pour cela, suivez les étapes 3 à 5 de ce tutoriel https://www.tecmint.com/install-virtualbox-guest-additions-in-ubuntu/

A la fin de l'installation, n'oubliez pas de redémarrer votre VM pour que les modules Guest Additions soit chargés.

2. Validez l'installation de Guest Additions avec votre encadrant de TP

2.3 Création d'un espace de partage entre votre VM et votre laptop

Souvent, il est très utile d'avoir les fichiers et dossiers de votre laptop accessibles depuis la VM (vous pourrez par exemple, écrire un script sur votre éditeur Windows préféré et l'exécuter sur la VM).

Vous devez à présent configurer un dossier partagé entre votre laptop et votre VM (e.g. un dossier de votre Desktop sur Windows). Pour cela, suivez le tutoriel disponible à cette adresse https://youtu.be/9-teQnZ8LEY . Une fois configuré le dossier partagé, vous devez l'accéder à travers l'explorateur de fichier (icône d'un dossier sur la barre à gauche) car dans notre version Ubuntu, les systèmes partagés ne s'affichent pas directement sur le bureau.

3. Validez la création du dossier partagé avec votre encadrant de TP

3 Exécution de CORE

3.1 Topologie « 1 LAN »

Maintenant que votre VM est au point, nous allons commencer l'exploration de CORE. Lancez un terminal sur la VM, puis devenez « root » avec la commande :

```
$ sudo su
```

Sur le promp root, exécutez la commande

```
# core-daemon
```

Ceci affichera quelques lignes sans retourner le prompt. C'est le *backend* de CORE qui s'exécute et qui sera le vrai responsable de la création des réseaux virtuels. Vous devez donc laisser cette fenêtre ouverte et ouvrir un nouvel onglet sur le terminal pour récupérer un prompt d'utilisateur standard.

Ouvrez donc un nouvel onglet (click sur l'icône « onglet » en haut à gauche de la fenêtre terminal ou appuyez sur Ctrl+Shift+T). Exécutez maintenant le client CORE (le *frontend)* avec la commande

```
$ core-gui
```

Vérifiez que tout se passe bien. L'interface graphique de CORE doit s'afficher et sur le terminal, entre autres, vous devez avoir la ligne « Connecting to "core-daemon" (127.0.0.1:4038)...connected. ».

Pour créer un réseau virtuel, vous pouvez ajouter des équipements de niveau 3 (bouton « network-layer virtual nodes », icône « routeur » sur la barre à gauche), des équipements de niveau 2 (bouton « link-layer nodes », icône « switch »), et des liens entre les équipements (bouton « link tool »). Voici un screenshot pour vous aider à repérer ces éléments.

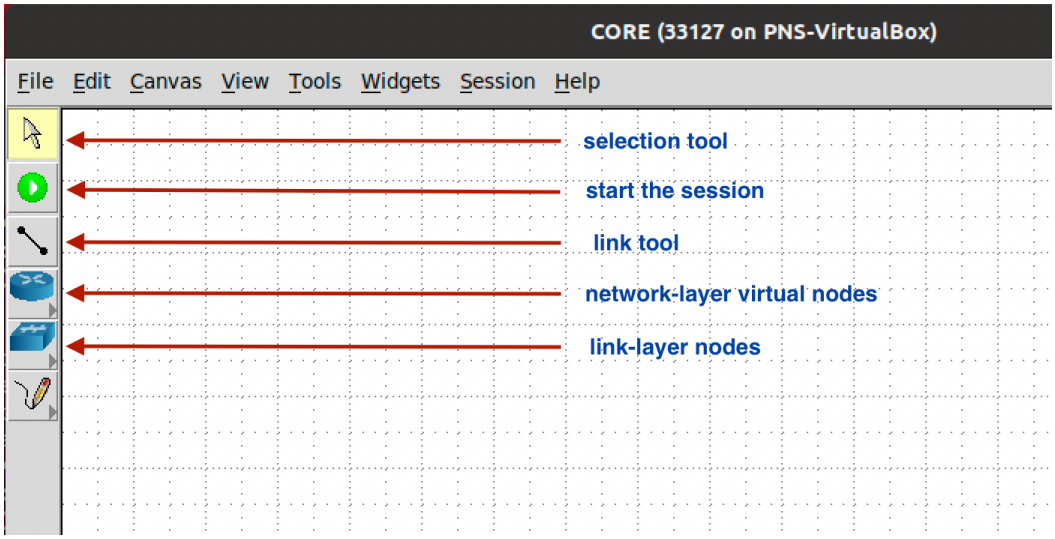

Figure 1. Interface graphique CORE

A présent, créez la topologie réseau ci-dessous. Clickez sur « network-layer », puis clickez sur « host » (icône « serveur »). Faites click sur l'espace de travail pour y placer 2 nœuds. Ensuite, clickez sur link-layer », puis « ethernet switch » (2ème icône de gauche à droite). Faites click sur l'espace de travail pour place un commutateur (switch). Pour finir, après avoir fait click sur « link tool », créez un lien entre les nœuds comme montré sur la figure (click sur le « host » et en maintenant pressé la souris, rejoindre le « switch »).

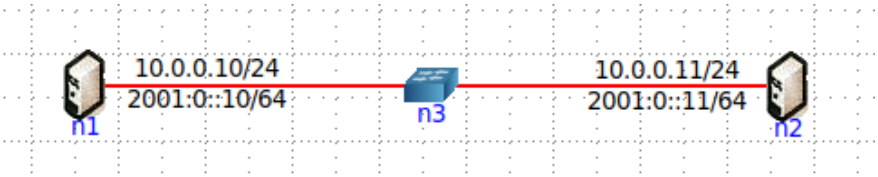

Figure 2. Topologie réseau avec 2 serveurs et 1 switch

Une fois la topologie réseau créée, vous pouvez la lancer en clickant sur le bouton « star the session ». Des cadres rouges devront être dessiné sur chaque dispositif réseau au moment de leur déploiement, puis les cadres deviendront verts. Les disparitions de cadres indiqueront que le réseau virtuel est créé et prêt à son utilisation.

Pour interagir avec le réseau, nous pouvons double-clickez sur les nœuds de niveau 3 (i.e. les équipements créés avec le bouton « network-layer virtual nodes ») afin d'obtenir un shell (terminal).

4. En Linux, la configuration d'une interface réseau est obtenu principalement avec la commande `ip`. Double-clickez sur les serveurs (n1 et n2 dans la Figure 2) et avec la commande « `ip addr show` » donnez la configuration de l'interface « eth0 ». Est-ce que

l'adresse IPv4 montré par CORE (10.0.0.10 pour n1 dans la Figure 2) est la même que vous obtenez avec la commande « ip » ?

5. Faisons maintenant un « ping » pour valider l'accessibilité de « n2 » depuis « n1 ». Ouvrez un terminal pour « n1 » et exécutez la commande « ping -c5 10.0.0.11 ». Faites également un ping vers l'adresse 10.0.0.100 (adresse inexistant). Donnez les résultats de la commande et expliquez la sortie du ping lorsque la machine ciblée existe.

   a) Un ping est une application très simple où un host envoie une requête (techniquement, il s'agit d'un message *ICMP echo request*) et le récepteur envoie une réponse (message *ICMP echo reply*)

   b) Faites un ping cette fois-ci sans le paramètre « -c ». Pour l'arrêter, vous devez faire un « Ctrl-c »

6. Ouvrez un terminal sur « n2 » et effacez son adresse IPv4 actuelle avec la commande « ip addr del 10.0.0.11/24 dev eth0 ». Vérifiez avec la commande « ip a s » (abreviation de « ip addr show ») que l'adresse IPv4 est partie. Ensuite donnez-lui l'adresse 10.0.0.100 avec la commande « ip addr add 10.0.0.100/24 ». Enfin, vérifiez par ligne de commande que la nouvelle adresse est présente. Lisez la configuration IPv4 donnée de « n2 » montrée par CORE. Voyez-vous l'importance de l'utilisation de la ligne de commande pour vérifier la configuration de votre réseau de test ?
7. Comment prouver par un ping depuis « n1 » que la nouvelle adresse de « n2 » est opérationnelle ?
8. Arrêtez le réseau virtuel en clickant le bouton « stop the session ».

3.2 Topologie « multi-LANs »

9. Après avoir arrêté l'exécution du réseau virtuel, modifiez votre topologie pour ajouter un 2 routeur et un 3ème serveur, comme montré dans la figure ci-dessus. ***Les adresses réseaux sont assignés au fur et à mesure que les liens sont placés. Les adresses peuvent donc différer sur votre réseau.***

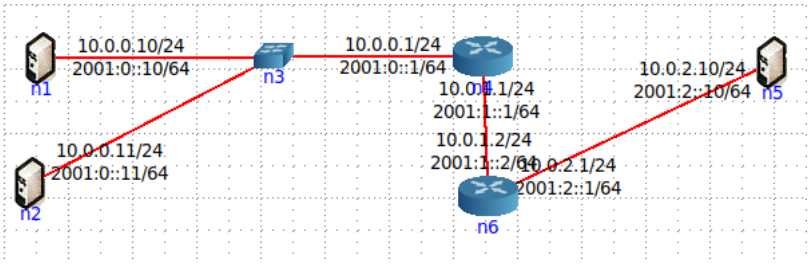

Figure 3. Topologie multi- LANs

10. Lancez le déploiement du réseau, attendez quelques seconds après le déploiement du réseau et vérifiez que vous pouvez pinger « n5 » depuis « n2 » , sur la Figure 3. Montrez par de captures d'écrans que le ping marche comme attendu.
11. Arrêtez le réseau virtuel, puis quittez l'interface graphique de CORE, mais laissez tourner le daemon.

Après avoir fait le dernier exercice, vous pouvez voir la « magie » de CORE, qui fournit une connectivité totale entre les éléments sans que vous n'ayez rien à faire. La partie réseau de ce cours a pour but de vous montrer les grands principes des réseaux IP afin que vous ayez conscience que lorsqu'on utilise le réseau, plusieurs éléments doivent se mettre en place pour que votre service (application distribuée) fonctionne correctement.

4 Exploration de quelques services réseau

Téléchargez le fichier « topo-2lans.imn ». Lancez à nouveau l'interface graphique de CORE et par le menu « File -> Open… » ouvrez le fichier .imn téléchargé. Lancez la création du réseau virtuel.

4.1 Obtention d'une adresse

12. Ouvrez un terminal sur « n2 » et exécutez la commande « # ps aux ». Vérifiez qu'une ligne commençant par « root » et finissant par « dhcpd » existe, continuez avec l'exercice suivant. Si ce n'est pas le cas, fermez tout, quittez CORE complètement et redémarrez votre VM avant de ressayer cet exercice.
13. Ouvrez un terminal sur « n1 » et donnez son adresse IPv4 si elle existe. Donnez la commande et la sortie que vous avez obtenue.
14. Sur le même terminal de « n1 » exécutez la commande « # dhclient -v eth0 ». Ensuite, vérifiez l'adresse IPv4. Donnez les informations obtenues.

4.2 Liste des entrées dans une table de routage

15. La commande pour visualiser les tables de routages est « ip route show ». A la place de show, on pourrait utiliser aussi « add » ou « del » pour ajouter ou effacer un entrée de la table de routage. Ouvrez un terminal sur le routeur « n5 » et exécutez la commande « ip r s » (r s = « route show »). Montrez par un screenshot le résultat. Nous verrons dans une session dédiée au routage IPv4 ce qui est une table de routage et vous comprendrez mieux les détails de la sortie.

4.3 Test du navigateur Firefox

16. Ouvrez un terminal sur « n3 », puis exécutez la commande « # python3 -m http.server 80 » pour créer un serveur HTTP minimale. La commande ne renvoie pas le prompt. Laisser tourner votre commande.
17. Fermez toute fenêtre Firefox ouverte. Allez sur le terminal de « n1 » et exécutez « # firefox ». Si vous voyez un message « Error: cannot open display: :0 », exécutez sur un terminal de votre VM (i.e. en dehors du réseau virtuel), la commande « xhost + ». Puis, ressayez à nouveau la commande « Firefox » sur le terminal de « n1 ».

    a) Il est normal que Firefox ne trouve pas une connexion Internet puisque notre réseau virtuel ne possède aucune sortie vers le monde externe.

18. Une fois Firefox ouvert depuis « n1 », tapez sur la barre d'adresse l'adresse IPv4 de « n3 ». Vérifiez que le serveur HTTP fonctionne correctement en naviguant avec Firefox sur le système de fichiers de « n3 ».
19. Ouvrez un 2ème terminal sur « n3 » et exécutez la commande « # wireshark ». Sur la fenêtre qui s'ouvre, double-clickez sur « eth0 ». L'application Wireshark commencera une capture de tout le trafic vu par l'interface « eth0 » de « n3 ».
20. Continuez à explorer le système de fichier de « n3 » depuis Firefox et essayez d'identifier le trafic réseau correspondant sur Wireshark.
21. Fermez Wireshark, le navigateur Firefox et sur le terminal où le serveur HTTP python tourne, exécutez Ctrl+c pour arrêter le serveur.

4.4 Test du service SSH

22. Ouvrez un terminal sur « n3 » si ce n'est pas encore ouvert, puis donnez un mot de passe à son utilisateur root (qui n'est pas le root de la VM) en exécutant la commande « #

passwd ». Vous devrez taper 2 fois le même mot de passe. Même si rien ne s'affiche sur le terminal, ce que vous tapez sera correctement capté par la commande.

23. Ouvrez un terminal sur « n2 » si ce n'est pas encore ouvert. Exécutez la commande « ssh root@10.0.0.10 » piur vous logger sur « n3 ». Vous devrez accepter (« yes ») les certificats de sécurité et utiliser le mot de passe root donné sur le point précédent.
24. Vérifiez par la commande « hostname » que vous êtes bel et bien sur « n3 ». Finissez la session SSH par la commande « exit » et revenir sur « n2 ».

4.5 Test de bande passante avec iperf

25. Sur un terminal de « n3 », exécutez un serveur iperf avec la commande « iperf -s -i1 » (où i1 indique qu'on voudrait avoir une synthèse des tatistiques 1 fois par second). Le prompt n'est pas renvoyé. Laissez donc tourner iperf.
26. Sur le terminal « n2 », exécutez un client iperf avec la commande « iperf -c 10.0.0.10 ». Ce client enverra autant de donnés que possible vers un 10.0.0.10 en utilisant le protocole TCP. L'objectif étant de voir quel débit maximal est atteint au niveau du récepteur (le serveur iperf) pour déterminer la capacité du goulot d'étranglement.
27. Attendez la fin de l'exécution du client iperf et regardez côté serveur quel est le débit (et donc la capacité du goulot d'étranglement) entre « n2 » et « n3 ».
28. Avec CORE, on aurait pu également observer le débit instantané transitant par les liens et déclencher une alarme lors que ce débit atteint une certaine capacité. Faite click sur « Widgets -> Configure Throughput » et déclarez un seuil (*threshold*) à 1000 kbps (i.e. 1Mbps). Puis activez l'alarme (click sur « Widgets -> Throughput »).
29. Répétez l'expérience avec iperf et vérifiez que l'alarme se déclenche correctement lors de la transmission des données. Vous avez ici un moyen facile de détecter le chemin emprunté par le trafic réseau.
30. Fermez tous les terminaux et arrêtez le réseau virtuel (click sur « stop the session »). Arrêtez le daemon CORE en appuyant sur Ctrl+c.

Toujours arrêter le réseaux virtuel avant de fermer l'interface graphique et daemon de CORE
