TD Routage et Base de Donné de Retransmission

Réseaux : Configuration & Programmation

Dino López – dino.lopez@unice.fr

L'objectif de ce TP est de vous permettre de réaffirmer vos connaissances sur le routage et mécanismes de retransmission par la pratique. N'hésitez pas à revoir vos slides du cours et votre TP d'introduction à Mininet.

1. Les couches de la pile protocolaire TCP/IP

Exécutez Mininet (avec la topologie par défaut). Ouvrez ensuite une terminal (xterm) pour le host h1 et h2. Sur h2 créez un serveur web avec la commande « python -m SimpleHTTPServer ». Sur h1 exécutez la commande tcpdump comme suit « tcpdump "tcp port 8000" –i h1-eth0 –w /tmp/out-h1.pcap ». Enfin, depuis la CLI Mininet, exécutez une requête sur le serveur avec wget (e.g. « h1 wget 10.0.0.2 :8000 »). Si la commande wget réussi et que vous avez eu un message similaire à « 2016-01-27 14:20:47 (131 MB/s) - 'index.html' saved [615/615] » après son exécution, alors arrêtez tcpdump (aller sur xterm, puis Ctrl-C), et sortez de Mininet.

Avant de se focaliser sur les couches protocolaires, répondez aux questions suivantes qui vous permettront de mieux comprendre les stratégies de capture du trafic (très important lorsqu'on veut étudier les mécanismes du réseau).

a) L'argument "tcp port 8000" est un filtre de capture. Quel est l'objectif du filtre utilisé dans votre commande tcpdump ? Vous pouvez voir sur le site https://danielmiessler.com/study/tcpdump/ des exemples de filtre tcpdump afin de répondre à cette et aux 2 prochaines questions.

   Capturer tous les paquets TCP à destination/en provenance du port 8000

b) Sachant que le service IMAPS utilise le port 993 du protocole TCP, quel serait le filtre permettant de capturer tous les paquets à destination ou en provenance d'un serveur IMAPS ?

   "tcp port 993"

c) Quel serait le filtre pour capturer tous les paquets en provenance ou à destination du serveur www.unice.fr ?

   "host www.unice.fr"

Et voici maintenant des questions liées aux protocoles Internet.

a) La fenêtre Wireshark est divisé en 3 sous-fenêtres : une sous-fenêtre en haut qui contient une ligne par paquet capturé, une sous-fenêtre au milieu qui contient une description en format texte du contenu du paquet sélectionné dans la sous-fenêtre en haut, avec une section par couche protocolaire détectée (sauf la première section, qui ne fait référence à aucune couche protocolaire). Enfin, une sous-fenêtre basse qui contient le contenu du paquet (sélectionné en haut) tel

qu'il transite sur le réseau, en format hexadécimal et ASCII. Sélectionnez le paquet qui contient, au niveau de la colonne Info, « GET / HTTP/1.1 », et indiquez combien de couches vous pouvez y identifier et quel protocole vous identifiez par couche.

Couche Liaison de Données -> Protocole Ethernet
Couche Réseau -> Protocole IPv4
Couche Transport -> TCP
Couche Application -> HTTP

b) Quelle est la taille en octets de ces couches ?

Couche Liaison de Données -> 14o
Couche Réseau -> 20o
Couche Transport -> 32o
Couche Application -> 111o

c) Quelle est l'adresse MAC source et quelle est l'adresse MAC de destination ?

Src : 96 :49 :3f :90 :56 :9f
Dst : 86 :ec :c8 :b3 :df :ec

d) Donnez les adresses IP source et destination

Src : 10.0.0.1
Dst : 10.0.0.2

e) Donnez le port source et destination, en format décimale, ainsi que tel que vous le visualisez sur la sous-fenêtre base de Wireshark (pour cela, il faut préalablement sélectionner la ligne indiquant le numéro de port sur la partie centrale)

Src : 57122 -> df 22
Dst : 8000c-> 1f 40

2. Adressage IPv4

f) Soit l'adresse 192.168.1.128/20
   a. Combien de bits sont utilisés pour écrire l'adresse réseau ?

20

   b. Combien de bits sont utilisés pour écrire l'adresse des hôtes ?

12

   c. Quel est le masque de réseau correspondant ?

255.255.240.0

g) On attribue une adresse de type B à notre organisation
   a. Donnez un exemple d'adresse réseau qu'on aurait pu nous attribuer

180.1.0.0

b. Découpez ce réseau en 8 sous-réseaux (donnez les adresses réseaux obtenues). Supposez que l'utilisation des adresses réseau « zéro » et « tous à 1 » est permit

180.1.0.0/19, 180.1.32.0/19, 180.1.64.0/19, 180.1.96.0/19, 180.1.128.0/19, 180.1.160.0/19, 180.1.192.0/19, 180.1.224.0/19

c. Quelles seraient les première et dernière adresses d'hôte valable du premier sous-réseau ?

180.1.0.1 – 180.1.31.254

d. Prenez l'an de vos sous-réseaux et découpez-le à nouveau en 2 sous-réseaux

Si on prend le réseau 180.1.32.0/19, on le découperait sous la forme 180.1.32.0/20 et 180.1.48.0/20

e. Montrez graphiquement l'architecture de votre réseau (i.e. un routeur avec ses liens) et indiquez sur chaque lien dessiné l'adresse réseau auquel il appartient, en format CIDR.

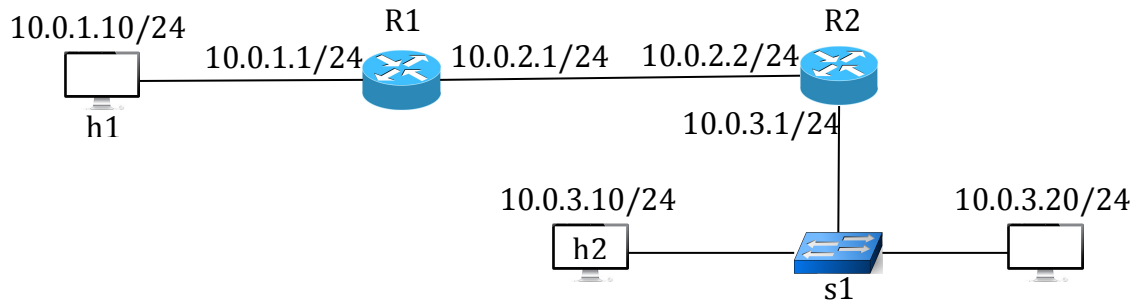
On aurait notre routeur R avec 9 liens attachés (= 7 sous-réseaux /19 + 2 sous-réseaux/20)

h) Sachant que le LAN principale de votre entreprise est le 192.168.64.0, votre technicien réseau vous propose la configuration ci-dessous afin de créer 2 sous-réseaux. Devez vous approuver cette proposition et pourquoi ?

```
LAN01 (192.168.64.128/25) --R1-- LAN PRINC SUR 192.168.64.0/24
LAN02 (192.168.64.0/25) ----|
```

Non. L'utilisation du réseau « 0 » sur 2 liens de R1 produirait une incapacité du routeur à retransmettre correctement les paquets entre les réseaux LAN02 et LAN PRINC

3. Tables de routages



Voici les exercices à faire :

a) Téléchargez et complétez le fichier testroutage.py pour créer le réseau de test montré dans la figure ci-dessus. Comme vous pouvez le voir, le script possède

une *main* qui permet le lancement du réseau virtuel directement par l'interpréteur Python. Notez svp que les routeurs R1 et R2 doivent être crées avec la méthode addNode() également, comme pour les clients du réseau.

b) Configurez depuis votre programme Python les interfaces réseaux, comme décrit dans l'image. Vérifiez depuis la CLI que tout marche correctement. Donnez les commandes que vous avez exécutez.

Un exemple disponible à http://www.i3s.unice.fr/~lopezpac/cours/2015-2016/resprogconf/corr/testroutage.py contient les réponses jusqu'à ce point-ci

c) Exécutez un ping depuis h1 vers h2 en ajoutant le paramètre –c3. Si la commande ping finit avec un message d'erreur « connect: Network is unreachable », alors ajoutez une route par défaut (celui qui permet d'atteindre tout autre réseau externe) sur h1. Faites de même avec h2 et h3. Donnez les commandes que vous avez exécutez.

Depuis la CLI, et pour h1 : h1 route add default gw 10.0.1.1 dev h1-eth0
Faire de même avec les autres hosts

d) Une fois que h1, h2 et h3 possèdent une route par défaut, un ping entre h1 et h3 devrait terminer avec un message similaire à « X packets transmitted, 0 received, 100% packet loss, time YYYYms ». Ceci indique tout simplement que les routeurs ne sont pas opérationnelles, mais c'est ce que vous devez faire dans les exercices à venir. Vérifiez donc que vous n'obtenez plus le message « Network is unreachable » et si c'est le cas, continuez.

e) Donnez les tables de routage de h1 et h3, en suivant le format de la question *g*

Table de Routage h1

Adresse Réseau	Passerelle (*gateway*)	Interface
10.0.1.0/24	0.0.0.0 (ou *)	h1-eth0
0.0.0.0 (ou *)	10.0.1.1	h1-eth0

Faire de même pour h3

f) Depuis la CLI, cherchez quel client peut « pinger » quel autre client.

Pour l'instant, uniquement h2 et h3 peuvent se pinger entre eux-mêmes

g) Activez le routage sur R1 et R2 avec la commande « sysctl -w net.ipv4.ip_forward=1 ». Montrez la table de routage sur R1 et R2.

Depuis la CLI, précéder la commande avec « R1 » ou « R2 » selon le cas.
Table de Routage R2

Adresse Réseau	Passerelle (*gateway*)	Interface
10.0.2.0/24	0.0.0.0 (ou *)	R2-eth0
10.0.3.0/24	0.0.0.0 (ou *)	R2-eth1

Faire de même pour R1

h) En utilisant la table ci-dessous, donnez la table de routage que les routeurs R1 et R2 devraient avoir afin de fournir une connectivité totale du réseau.

Exemple, Table de Routage R2

Adresse Réseau	Passerelle (*gateway*)	Interface
10.0.2.0/24	0.0.0.0 (ou *)	R2-eth0
10.0.3.0/24	0.0.0.0 (ou *)	R2-eth1
10.0.1.0/24	10.0.2.1	R2-eth0

Faire de même avec R1

i) Modifiez les tables de routage de R1 et R2 depuis la CLI afin de rentrer toutes le routes que vous avez choisi dans le point précédent.

Puisque R2 contient une route pour les adresses 10.0.2.0 et 10.0.3.0, il nous reste uniquement ajouter la route pour 10.0.1.0. Depluis la CLI en exécutera donc : R2 route add -net 10.0.1.0 gw 10.0.2.1 dev R2-eth0
Faire de même avec l'autre routeur.

j) Si tout fonctionne correctement, ajoutez automatiquement toutes les routes sur R1 et R2 (i.e. directement sur le code Python)

A l'aide la méthode cmd() il faut uniquement réutiliser les commandes précédentes

Vous pouvez continuer à la prochaine section uniquement si vous avez finit cette section correctement, car vous aurez besoin du script testroutage.py complètement opérationnel.

4. FDB des switches

a) Exécutez votre programme testroutage.py et exécutez un ping depuis le client 10.0.3.10 vers les clients 10.0.3.20 et 10.0.1.10.
b) Donnez la FDB attendu du switch s1

@MAC	Port
MAC de h2 (ex. 00 :00 :00 :00 :00 :02)	Port où h2 est connecté. Ex. 1
MAC de h3 (ex. 00 :00 :00 :00 :00 :03)	Port où h3 est connecté. Ex. 2
MAC de R2-eth1 (car connecté à s1)	Port où R2-eth1 est connecté.

c) Depuis un terminal externe à Mininet, exécutez la commande « brctl showmacs s1 » et donnez la FBD obtenue (ignorez les interfaces locales). Vérifiez que votre réponse au point b) correspond exactement à ce que vous obtenez sur la machine.

port no	mac addr	is local?	ageing timer
2	1a:6e:0a:82:4d:d1	no	203.28
1	ca:34:7f:23:28:5f	no	203.28

La commande renvoie une sortie légèrement différente. En effet, le switch connais bien l'emplacement du host h2 et R2. Ceci est dû au fait que le host h3 est inactif depuis pas mal de temps et s1 a effacé l'entrée correspondante. En faisant un ping entre h1 et h2, suivi d'un ping entre h1 et h3 devra donc donner la solution proposée en b).
Comme vu en cours, l'adresse MAC de hosts externes (ex. h1) n'apparaissent pas dans la liste, mais uniquement celle de la passerelle.