

TD Routage et Base de Donnée 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` réussit 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.
- 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 ?
- c) Quel serait le filtre pour capturer tous les paquets en provenance ou à destination du serveur 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.
- b) Quelle est la taille en octets de ces couches ?

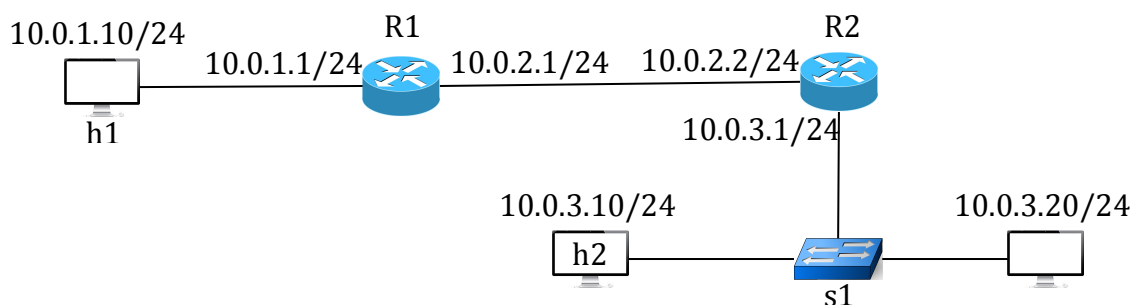
- c) Quelle est l'adresse MAC source et quelle est l'adresse MAC de destination ?
- d) Donnez les adresses IP source et destination
- e) Donnez le port source et destination, en format décimale, ainsi que tel que vous le visualisez sur la sous-fenêtre base de Wireshark (pour cela, il faut préalablement sélectionner la ligne indiquant le numéro de port sur la partie centrale)

2. Adressage IPv4

- f) Soit l'adresse 192.168.1.128/20
 - a. Combien de bits sont utilisés pour écrire l'adresse réseau ?
 - b. Combien de bits sont utilisés pour écrire l'adresse des hôtes ?
 - c. Quel est le masque de réseau correspondant ?
- g) On attribue une adresse de type B à notre organisation
 - a. Donnez un exemple d'adresse réseau qu'on aurait pu nous attribuer
 - 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
 - c. Quelles seraient les première et dernière adresses d'hôte valable du premier sous-réseau ?
 - d. Prenez l'un de vos sous-réseaux et découpez-le à nouveau en 2 sous-réseaux
 - 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.
- 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) ----|

3. Tables de routages



Voici les exercices à faire :

- a) Téléchargez et complétez le fichier `testrou tage.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 créés 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 exécutées.
- 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 exécutées.
- 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

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

- 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.

- i) Modifiez les tables de routage de R1 et R2 depuis la CLI afin de rentrer toutes les routes que vous avez choisi dans le point précédent.
- j) Si tout fonctionne correctement, ajoutez automatiquement toutes les routes sur R1 et R2 (i.e. directement sur le code Python)

Vous pouvez continuer à la prochaine section uniquement si vous avez fini cette section correctement, car vous aurez besoin du script `testrou tage.py` complètement opérationnel.

4. FDB des switches

- a) Exécutez votre programme `testrou tage.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

- 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.