

TP Introduction aux Réseaux

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

1 Introduction

Dans ce TP, vous trouverez quelques exercices qui vous permettront de réaffirmer vos connaissances acquises à propos de l'architecture physique des réseaux, ainsi que les phénomènes observés lors de l'échange de paquets (messages).

2 Architecture Internet

1. Les switches sont utilisés pour fournir un accès aux clients des réseaux locaux filaires. Supposez que vous avez des commutateurs (switches) avec 5 ports chacun. Nous réserverons cependant un port pour l'accès Internet (ce qui laisse donc 4 ports pour les clients). Quelle topologie réseau construire afin de donner accès internet à 20 clients ? combien de switches vous faut-il au total ? Dessinez votre topologie.

On doit donner accès aux clients et distribuer la bande passante au plus équitable possible. C'est donc une topologie en arbre (inversé) symétrique qu'on choisit naturellement. Avec un switch, on a 4 clients et on garde un port pour l'accès Internet. On augmente l'arbre d'un degré, et nous nous retrouvons avec 4 switches (= 16 ports disponibles). Il nous faut augmenter la profondeur de l'arbre, mais uniquement pour ajouter 2 switches de plus. Question extra : Combien de ports nous avons au total pour connecter des clients ?

2. Quelle profondeur doit avoir une topologie arbre pour pouvoir donner service à n clients si on possède de switches à p ports uniquement ?

Supposez que nos switches ont p ports et l'un de ces ports doit être utilisé pour l'interconnexion avec un switch. Donc, nous aurons $c = p - 1$ ports disponibles pour les clients.

Grâce à l'exercice précédent, on voit que, au niveau X de l'arbre, nous avons c^X ports disponibles. Il faut donc que $c^X \geq n$. Il ne nous reste qu'à utiliser les propriétés des logarithmes pour trouver notre bonne valeur de X .

3. Les routeurs sont de dispositifs qui permettent d'accéder à d'autres réseaux. De ce fait, chaque réseau doit posséder au moins 1 routeurs afin d'accéder à l'Internet. Ajoutez un routeur à votre topologie réseau de la question 1.

Connectons un switch en haut de l'arbre inversé.

4. Supposez que l'architecture réseau faite dans le point 3 doit être répliquée 4 fois, car par exemple, nous avons 4 départements indépendants dans des bâtiments séparés. Si nous savons que chaque routeur doit avoir au moins 2 liaisons vers d'autres routeurs afin d'éviter les isollements en cas de panne, proposez une topologie valable pour l'interconnexions de vos 4 sites.

Note : l'exercice comporte la phrase « au moins 2 liaisons ». Avec 3 routeurs ou moins, la topologie générée serait une topologie forcément circulaire (*ring*). Avec 4 routeurs ou plus, la topologie peut être circulaire ou maillée (*mesh*), selon vos choix. Pourquoi dit-on donc que la topologie du cœur de l'internet est maillée ? Parce que la

géographie fait que les réseaux circulaires sont souvent difficiles à créer. Puis, 2 liaisons entre routeurs est vraiment le strict minimum demandé ! Voir par exemple la topologie du réseau français RENATER https://www.researchgate.net/figure/RENATER-topology-with-45-sites-and-54-links-between-them_fig3_301413394

5. En partant du schéma entier d'interconnexion des 4 sites, identifiez les topologies réseaux qui ont été utilisés.

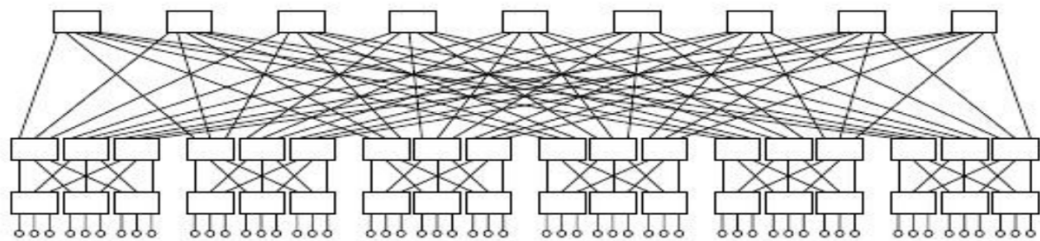
Il y a une topologie maillée, en arbre et étoile (sur chaque feuille de l'arbre)

6. Les réseaux n'ont pas une capacité infinie. Décrivez comment un réseau à commutation de circuit et un réseau à commutation de paquets peuvent expérimenter des phénomènes de congestion.

Plusieurs clients émettant au total plus des paquets que ce qui peut être transféré par les routeurs ou switches, conduisent à des problèmes de congestion dans un réseau à commutation de paquets.

Plusieurs clients essayant de créer un circuit virtuel satureront les switches et conduiront également à un problème de saturation.

7. Combien de serveurs au total on pourrait héberger dans un réseau k-6 fat tree ? dessinez le réseau.



3 Capture de trafic réseau

Pour debugger ou analyser un protocole ou une application répartie, il est souvent nécessaire de capturer le trafic transitant par le réseau pour s'assurer que les choses se passent comme prévu. Voici quelques exercices qui ont pour but de vous apprendre à maîtriser l'outil de capture tcpdump.

Pour les exercices de ce TP, la page web suivante peut s'avérer très utile <https://docs.netgate.com/pfsense/en/latest/book/packetcapture/using-tcpdump-from-the-command-line.html>

8. Dans votre machine virtuel Linux

- Trouvez le nom de l'interface avec l'adresse IP « 10.0.X.X/24 ». Donnez aussi l'adresse MAC (adresse disponible dans la ligne « link/ether »).

```
dlopez@dl-vbox:/media/sf_EPU/enseignement/2020-2021/IPA/videos$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:19:bc:0f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83493sec preferred_lft 83493sec
    inet6 fe80::ae44:ae63:ad01:6129/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ignorer les informations de « lo »

Nom interface : enp0s3

Adresse IP : 10.0.2.15

- En mode super utilisateur (e.g. en exécutant « `sudo su` »), exécutez `tcpdump` de la manière suivante « `tcpdump -i youriface -w /tmp/test1.1.pcap` », où *youriface* est le nom de l'interface que vous avez trouvé dans le point précédent. Laissez le programme tourner, puis, visitez un site internet quelconque (e.g. twitter.com) à l'aide de votre navigateur.
- `tcpdump` est un outil pour capturer le trafic réseau (un *sniffer*). A l'aide de la man page, expliquez à quoi sert chaque option passez en paramètre.

On capture tout le trafic vu par l'interface *youriface* et l'enregistre dans le fichier `/tmp/test1.1.pcap`

- Arrêtez `tcpdump` avec « `Ctrl-C` » dans le terminal, puis ouvrez le fichier pcap avec la commande « `wireshark /tmp/test1.1.pcap` ». Expliquez brièvement les informations disponibles dans chaque division de la fenêtre Wireshark. Vous devez faire un screenshot de votre fenêtre Wireshark et indiquer dessus les informations qu'on y retrouve.

Interface graphique vu en cours. De manière simple, la partie supérieure montre tout le trafic capturé. La partie au milieu montre des détails sur les protocoles composant un paquet capturé. La partie base montre le contenu brut du paquet.

9. Comme vous l'avez sans doute compris, `tcpdump` a capturé tout le trafic réseau disponible sur votre interface, ce qui rend les choses très compliquées à la lecture et peut conduire à un fichier de trace trop volumineux. Il faut donc utiliser des filtres de capture afin de mieux cibler le trafic réseau à enregistrer.

Quelques mots clés pour écrire des filtres de capture :

« `port X` » permet de capturer le trafic en provenance ou à destination du port numéro X ; « `tcp` » capture le trafic TCP ; « `udp` » le trafic UDP ; « `host adresse_IP_ou_nom_de_machine` » capture le trafic en provenance ou à destination de l'adresse ou nom de machine donné ; « `net` » capture le trafic en provenance ou à destination de l'adresse réseau spécifiée.

Il est possible de précéder les mots clés « `port` », « `host` », « `net` » par les mots clés « `src` » et « `dst` » pour capturer uniquement le trafic en provenance et à destination respectivement de l'argument donné.

Les opérateurs logiques « `and` » et « `or` » permettent de créer un filtre avec de multiples expressions, et « `not` » de trouver le complément d'une expression. L'utilisation de parenthèses permet également de définir l'ordre d'évaluation de filtres.

- Expliquez à quoi sert donc la commande « `tcpdump "tcp port 80" -i youriface -w /tmp/test2.1.pcap` », où « `tcp port 80` » est en fait le filtre de capture.

On capture tous les paquets TCP en provenance ou à destination du port 80. En d'autres termes, on capture le trafic HTTP.

- Quel filtre de capture permet de capturer uniquement le trafic DHCP ? quel filtre utiliser pour capturer le trafic DNS ?

DHCP = `udp and (port 68 or port 67)`

DNS = `port 53`

- Comment capturer le trafic en provenance ou à destination de la machine www.unice.fr ?

`host www.unice.fr`

- Comment capturer le trafic à destination du réseau `134.59.1.0/24` ?

`dst net 134.59.1.0/24`

- Comment capturer tout le trafic qui n'est pas de type ICMP (ICMP est le protocole qui vous permet de faire un ping –entre autres–) ?

`not icmp`

10. Donnez un filtre pour capturer un trafic HTTP et HTTPS à destination de la machine cliente. Prouvez que ça marche correctement en déployant la topologie `topo-1lan-4hosts.imn` avec CORE. Copiez sur « `n3` » les fichiers « `cert.pem` », « `key.pem` » et « `https.py` ». Donnez les permissions 644 et 600 à « `cert.pem` » et « `key.pem` » respectivement.

- Lancez le serveur HTTPS sur « `n3` » avec la commande « `python3 https.py` ». Le passphrase est "azerty", sans les "".
- Lancez le serveur HTTP sur « `n4` » avec la commande « `python3 -m http.server 80` »
- Ouvrez un terminal sur « `n1` » et lancez `tcpdump` avec votre filtre de capture.
- Ouvrez un 2^{ème} terminal sur « `n1` » et lancez firefox. Visitez le serveur HTTP puis le serveur HTTPS. La première fois que vous visiterez le serveur HTTPS, vous devez dire à firefox qu'il n'y a aucun problème de sécurité en acceptant le certificat auto-signé.
- Fournissez une capture d'écran de wireshark montrant le trafic HTTP et HTTPS à destination de « `n1` ».

`dst host 10.0.0.13 and tcp and (port 80 or port 443)`

4 Traffic et délai

11. Nous vous proposons maintenant d'expérimenter l'impact du trafic sur le délai observé entre 2 hosts. En réutilisant le réseau `topo-1lan-4hosts.imn`

- Lancez un ping depuis « `n2` » vers « `n4` » (« `ping adresse_ip_de_n4` ») et laissez tourner la commande
- déployez sur « `n3` » un serveur iperf (« `iperf -s -i0.2` »)

- sur « n1 », déployez le client iperf (« iperf -c *adresse_ip_de_n3* -t15 »)
- montrez graphiquement l'évolution du délai (vous pouvez utiliser LibreOffice Calc) et identifiez l'intervalle pendant laquelle iperf a été exécuté et expliquez vos observations.

Comportement attendu : si la bande passante de 100Mbps et le délai de propagation sont bien respectés, ping doit trouver un délai (temps) de 20ms, mais ce délai doit augmenter lors que le transfert avec iperf est en marche.