TP Adressage et Latence

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

1 Introduction

Ce TP est moins technique que les TPs précédents. C'est cependant un passage obligatoire pour :

1. Vous sensibiliser aux performance réseaux. Souvent, on peut entendre, le réseau va vite ou le réseau est lent. Il est nécessaire de comprendre donc cette performance réseau (qui dépend en grande partie de la bande passante et de la latence) car il peut déterminer si une application est réalisable ou non.
2. Réaffirmer vos connaissances dans la manière des hôtes doivent être configurés pour faire partie du réseau. En effet, en cas de problème lors de la connexion à un réseau, la première chose à faire est de vérifier si la configuration de dispositif, obtenue dynamiquement ou non, est correcte.

2 Latence Réseaux

1. Soit un paquet de taille L (en bits) voyageant dans un lien de longueur l (en km), débit d (en bits/sec) et vitesse de propagation v (en km/sec).
    - Quel délai de propagation, d_{prop}, expérimente un tel paquet ?

    l/v

    - Quel délai de transmission, d_{trans}, expérimente un tel paquet ?

    L/d

    - Si dprop > dtrans, où se trouve le premier bit d'un paquet lorsque le dernier bit du paquet quitte le host émetteur ?

    Il se trouve sur le réseau

2. Supposez que vous effectuez une audioconférence par Internet. Vous utilisez donc une application VoIP. Votre application VoIP transforme votre voix en un flux de bits à un débit constant de 64Kbps et les regroupe en (crée des) paquets de données de 56 octets. Si votre ordinateur possède une liaison ADSL à 2Mbps et le délai de propagation entre votre ordinateur et celui de votre partenaire est de 10ms, quel est le délai expérimenté par un bit de donnée (le son encodé) entre sa création au niveau de votre ordinateur et sa réception par l'ordinateur de votre partenaire ?

    Puisqu'un paquet ne peut pas être envoyé tant qu'il n'est pas créé, il faut trouver le temps de création d'un paquet, ajouter le délai de transmission, et enfin, le délai de propagation. Donc (56*8/64x10^3) + (56*8/2x10^6)+10ms

3. Supposez qu'une rafale de N paquets de taille L arrive à un routeur avec débit R et capacité infinie dans le buffer
    - Donnez le délai moyen dans la file d'attente si les N paquets arrivent avec un temps d'espacement de L/R entre 2 paquets successifs.

Os puisque le routeur a le temps d'envoyer chaque paquet reçu avant de voir un autre paquet arriver

- Donnez le délai moyen dans la file d'attente si les paquets arrivent au même temps

(0 + L/R + 2L/R + … (N-1)L/R)/N = (L/RN)(1+2+…+N-1) = (L/RN)((N-1)N/2) = L*(N-1)/(2R)

3 Adressage

4. Quelle est la différence entre les adresses IPv4 de type « classful » et « classless » ?

   Dans les adresses de type classful, le masque de réseaux (et donc l'adresse réseau) est définit par l'adresse IP d'un hôte. Dans les adresses de type classless, nous ne pouvons pas inférer l'adresse réseau à partir de l'adresse IP d'une machine. En effet, dans un schéma « classless », le masque de réseau s'écrivent sur un nombre arbitraire de bits, définit par le format /X (format CIDR).

5. Quelle est la différence entre une adresse de type privée et une adresse de type publique ?

   Les adresses privées peuvent se répéter sur Internet, contrairement aux adresses publiques. Cependant, un paquet contenant une adresse de type privée (src et/ou dst) ne sera routé dans le cœur de l'internet.

6. Par quel moyen les dispositifs avec adresse privée peuvent envoyer des données dans le cœur de l'Internet ?

   Grâces aux NAT, qui substituent nos adresses IP privées par une adresse IP publique.

7. Soit l'adresse 192.168.1.128/20

   - Combien de bits sont utilisés pour écrire l'adresse réseau ?

   20

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

   12

   - Quel est le masque de réseau correspondant ?

   255.255.240.0

   - Quelle est l'adresse réseau de la machine ?

   192.168.0.0/20

   - Quelle est l'adresse réseau de broadcast ?

   192.168.15.255

8. On attribue une adresse de type B à notre organisation

   - Donnez un exemple d'adresse réseau qu'on aurait pu nous attribuer

   180.1.0.0

   - Découpez ce réseau en 8 sous-réseaux (donnez les adresses réseaux obtenues en format CIDR). 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

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

180.1.0.1 – 180.1.31.254

- 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

- Donnez l'adresse de diffusion du premier sous-réseau de cette nouvelle division

Si on considère notre nouveau sous-réseau 180.1.32.0/20, l'adresse de diffusion ou broadcast sera là 180.1.47.255. On obtient une telle adresse en prenant tous les bits de la partie réseau tel quel, mais en mettant chaque bit de la partie hôte à 1.

- 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)

9. Votre technicien en charge de créer 3 sous-réseaux vous propose la configuration disponible dans le fichier « topo-2.imn ». Expliquez si vous pouvez accepter ou non une telle configuration et pourquoi.

   Non. Le sous-réseau 192.168.64.0/28 recouvrant l'espace d'adressage des autres 2 sous-réseau, le routeur ne saura pas comment router les paquets vers ce réseau, car les réseaux avec CIDR /29 seront priorisé. Pourquoi ? suite à l'application de la politique du « longest prefix matching ».

10. Téléchargez le fichier « topo-1.imn ». Déployez le réseau. Ensuite :
    - Créez un serveur iperf dans le nœud « n3 » avec la commande « iperf -s »
    - Créez un client iperf dans le nœud « n4 » avec la commande « iperf -c *adresse_IP_de_n3* -i1 ».
    - Est-ce que la session iperf s'exécute correctement ? Observez attentivement la sortie de vos commandes. Si la commande iperf réussi, c'est qu'il existe un chemin bidirectionnel entre « n3 » et « n4 » (en effet, iperf envoie plusieurs paquets TCP du client vers le serveur, et quelques paquets aussi, depuis le serveur vers le client).

    Oui, la commande réussie. Il y a donc un chemin bidirectionnel entre « n3 » et « n4 ».

    - Maintenant, faite un ping depuis « n3 » vers « n4 ». Expliquez pourquoi la commande ping vous envoie un tel message sur le terminal.

    Cette fois-ci la commande ne réussit pas et vous montre le texte « Réseau de destination injoignable ». Ceci est dû au fait que le réseau de « n4 » se trouve derrière un NAT et donc ce réseau est caché pour le reste de l' « Internet ». Sans aller dans les

détails de configuration du routeur de « n4 », on sait que ce routeur est aussi un NAT car le serveur iperf a rapporté que le client iperf possédait l'adresse 134.59.64.2 (donc l'adresse IPv4 privé a été substitué par une adresse publique).

4 Multiplexage et premier contact avec les sockets UDP et TCP

On prend de l'avance sur notre cours et avant de programmer une socket, nous allons commencer par observer son fonctionnement. Les exercices ci-dessous vous permettront de mieux comprendre le cours suivant à propos de sockets TCP et UDP au même temps que nous illustreront par la pratique la notion de « Multiplexage » vu en cours.

Les sockets sont le point de contact d'une application distribuée. En effet, avant de pouvoir envoyer ou recevoir une requête, une machine doit ouvrir une socket (qui sera associée explicitement à un numéro de port -e.g. 80 pour un serveur http- ou implicitement -e.g. Firefox qui associe un onglet de navigation à un port quelconque-).

Dans les exercices suivants, vous utiliserez la commande « socat », qui permet de créer des sockets de n'importe quel type. « socat » reçoit 2 arguments. Le 2ème argument est une adresse (ou socket) de lecture et le premier une adresse (ou socket) où l'on écrit ce qu'on reçoit.

11. Avec CORE, créez et déployez un réseau avec 3 serveurs connectés tous dans un seul LAN (même réseau IP). Appelons ces serveurs « svr1 », « svr2 » et « svr3 ».
12. Ouvre le terminal de « svr3 » et créez un serveur UDP, à l'écoute du port 5555 à l'aide de la commande « socat -T3600 STDIN UDP-L:5555 ». **Laissez tourner le serveur et lissez bien les points suivants pour comprendre la commande**.
    - Cette commande indique qu'on crée un serveur UDP à l'écoute du port 5555 (d'où le « UDP-L :5555 »). Ce qu'on reçoit par cette socket, on le récrira sur STDIN.
    - Le paramètre « -TX » indique que le nouveau processus est arrêté après X seconds d'inactivité de la socket distante.
13. Ouvrez un 2ème terminal sur le même serveur du point précédent et essayez de créer à nouveau un serveur UDP avec la même commande. Dites ce qui se passe et pourquoi.

    On ne peut pas ouvrir à nouveau le même port. Le port 5555 - UDP est déjà utilisé.
14. Que se passe-t-il si au lieu de créer une socket UDP d'écoute sur le port 5555, on essayait de créer une socket TCP sur le port 5555 (« TCP-L ») sur ce même serveur « svr3 » ? Donnez la commande que vous devez exécuter pour tester ce scénario et dites pourquoi vous pouvez ou ne pouvez pas ouvrir cette socket.

    On peut ouvrir une socket TCP et UDP sur le même numéro de port. Pour rappel, il n'y a pas d'ambiguïté sur quelle application en envoie les données entrantes (le type de protocole de transport fait la différence).
15. Arrêtez le serveur TCP sur « svr3 » avec Ctrl-C.
16. Que le serveur soit écrit dans un langage X n'empêchera pas un client développé dans un langage Y de s'y connecter. L'importance est de parler le même protocole. Prouvez que c'est bien le cas en connectant un client netcat à notre serveur socat avec la

commande « nc -u *adresse_IP_svr3 port_svr3* ». Le paramètre « -u » indique qu'il faut utiliser le protocole UDP pour envoyer des données à la machine distante.

- Donnez des screenshots pour prouver vos manips. Échangez quelques messages entre le client et le serveur

Client UDP à créer avec « nc -u -v 10.0.0.1 5555 » et sans le « -u » pour TCP. Le client TCP échoue sa demande de connexion car même si l'adresse IP du serveur et le port sont correctes, le protocole de transport est différent.

17. Sans arrêter le client UDP (nc), arrêtez le serveur UDP avec Ctrl-c. Démarrez à nouveau le serveur UDP avec la commande socat. Ensuite, envoyez quelques messages depuis le client vers le serveur. Est-ce que le serveur a lu les messages du client ?

    Oui, le client UDP envoie des données à l'adresse indiqué et le serveur arrive à tout lire.

18. Est-ce qu'il est possible de démarrer un client UDP alors que le serveur UDP n'est pas encore lancé ? Donnez une réponse en faisant une expérience avec « socat » et « nc ».

    Oui, c'est possible. UDP est un protocole non-connecté. Donc, l'absence du serveur ne pose pas de problème pour le démarrage d'un client.

19. Avec le protocole TCP, le serveur doit être démarré après ou avant le client ? Utilisez « socat » en mode serveur TCP. Pour créer un client « nc » TCP, enlevez le paramètre « -u » tout simplement.

    Avant le client. Dans le cas contraire, la tentative de connexion depuis le client échoue. TCP = communication en mode connecté.