Définitions des performances du réseau et exercices de mesure.

Notes :

* Les commandes précédées de "$" signifient que vous devez exécuter la commande en tant qu'utilisateur général - et non en tant qu'utilisateur root.
* Les commandes précédées de "#" signifient que vous devez travailler en tant qu'utilisateur root.
* Les commandes comportant des lignes de commande plus spécifiques (par exemple "rtrX>" ou "mysql>") signifient que vous exécutez des commandes sur des équipements à distance, ou dans un autre programme.
* Si une ligne de commande se termine par "\", ceci signifie que la commande se poursuit sur la ligne suivante et que vous devez la traiter comme une seule ligne.

Exercices Partie I

0. Ouvrez une session sur votre PC/machine virtuelle ou ouvrez une fenêtre de terminal en tant qu'utilisateur sysadm :

Remarque : Au cours de ces exercices, si la commande apt-get se plaint que certaines archives sont introuvables, vous devrez peut-être mettre à jour votre base de données des paquets apt. Pour cela, tapez :

```
$ sudo apt-get update
```

Indicateurs de performance du réseau

1. ping

ping est un programme qui envoie des paquets ICMP de demande d'écho à des hôtes cibles et attend une réponse ICMP de la part de ces derniers. Selon le système d'exploitation utilisé, vous pouvez observer des temps d'aller-retour (RTT, Round-Trip Time) minimaux, maximaux ou moyens, et parfois un écart type par rapport à la moyenne des réponses ICMP de l'hôte cible. Pour plus de précisions, voir :

http://en.wikipedia.org/wiki/Ping

Il est généralement déconseillé de bloquer ping.

En partant de ces éléments, essayez d'utiliser Ping de différentes manières :

```
    $ ping localhost
```

Appuyez sur ctrl-c pour arrêter le processus. Voici quelques résultats typiques de la commande ci-dessus :

```
    PING localhost (127.0.0.1) 56(84) bytes of data.
    64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.020 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.006 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.006 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.006 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.006 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.009 ms
    64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.007 ms
    ^C
    --- localhost ping statistics ---
    7 packets transmitted, 7 received, 0% packet loss, time 5994ms
    rtt min/avg/max/mdev = 0.006/0.008/0.020/0.005 ms
```

Question : pourquoi la première réponse ICMP a-t-elle mis 20 ms alors que les autres réponses ont été beaucoup plus rapides ? Il s'agit d'un type de délai. De quelle sorte de délai s'agit-il ?

2. traceroute

Vous avez peut-être déjà utilisé traceroute par le passé, mais avez-vous vraiment regardé ce qu'il fait ? Si ce n'est pas le cas, lisez ceci :

http://en.wikipedia.org/wiki/Traceroute

Il vous faudra peut-être installer d'abord la commande traceroute. Pour ce faire, tapez :

```
         $ sudo apt-get install traceroute
```

Une fois installé, essayez :

```
         $ traceroute nsrc.org
```

Ci-dessous un exemple du résultat de traceroute dans nsrc.org (les lignes bouclent en raison de leur longueur):

```
         traceroute to nsrc.org (128.223.157.19), 64 hops max, 52 byte
packets
          1  gw.ws.nsrc.org (10.10.0.254)  1.490 ms  1.069 ms  1.055 ms
          2  192.248.5.2 (192.248.5.2)  2.741 ms  2.450 ms  3.182 ms
          3  192.248.1.126 (192.248.1.126)  2.473 ms  2.497 ms  2.618 ms
          4  mb-t3-01-v4.bb.tein3.net (202.179.249.93)  26.324 ms  28.049
ms  27.403 ms
```

```
          5  sg-so-06-v4.bb.tein3.net (202.179.249.81)  103.321 ms  91.072
ms  91.674 ms
          6  jp-pop-sg-v4.bb.tein3.net (202.179.249.50)  168.948 ms
168.712 ms  168.903 ms
          7  tpr5-ge0-0-0-4.jp.apan.net (203.181.248.250)  172.789 ms
170.367 ms  188.689 ms
          8  losa-tokyo-tp2.transpac2.net (192.203.116.145)  579.586 ms
284.736 ms  284.202 ms
          9  abilene-1-lo-jmb-702.lsanca.pacificwave.net (207.231.240.131)
303.736 ms
             284.884 ms  530.854 ms
         10  vl-101.xe-0-0-0.core0-gw.pdx.oregon-gigapop.net
(198.32.165.65)  328.082 ms
             305.800 ms  533.644 ms
         11  vl-105.uonet9-gw.eug.oregon-gigapop.net (198.32.165.92)
336.680 ms  617.267 ms
            495.685 ms
         12  vl-3.uonet2-gw.uoregon.edu (128.223.3.2)  310.552 ms  421.638
ms  612.399 ms
         13  nsrc.org (128.223.157.19)  309.548 ms  612.151 ms  611.505 ms
```

Comprenez-vous ce que signifie chaque élément ? Si ce n'est pas le cas, consultez la page Wikipedia et tapez

```
$ man traceroute
```

pour plus d'informations. Qu'est-ce que cela signifie si vous voyez des lignes comme celles-ci ?

```
15  * * *
16  * * *
17  * * *
```

Ces lignes signifient que le dispositif distant ne répond pas aux demandes d'écho icmp, ou qu'il utilise une adresse de réseau privé (RFC 1918).

Comme vous pouvez le voir, traceroute peut être utilisé pour déterminer où ont lieu les problèmes entre deux points de terminaison d'un réseau.

Essayez d'exécuter de nouveau traceroute sur le même hôte (nsrc.org). Il faudra probablement beaucoup moins de temps.

3. mtr

L'outil mtr associe ping et traceroute en un seul et même affichage mis à jour de manière dynamique.
Avant d'utiliser mtr, vous devrez peut-être d'abord l'installer :

```
$ sudo apt-get install mtr-tiny
```

Maintenant essayez :

```
$ mtr nsrc.org
```

Le résultat de la commande varie selon les différentes versions de Linux et d'UNIX, mais vous obtiendrez en général un résumé indiquant les paquets perdus sur chaque noeud du chemin jusqu'à l'hôte cible distant, le nombre de paquets ICMP de demande d'écho envoyés, le dernier temps RTT vers l'hôte, le RTT minimal, maximal et moyen ainsi que l'écart type des temps RTT.

L'affichage du pourcentage de paquets perdus dans ce format permet de localiser beaucoup plus facilement les problèmes de réseau.

4. ping avec taille de paquet variable

Par défaut, ping envoie des datagrammes IP d'une taille de 84 octets :

* 20 octets d'en-tête IP
* 8 octets d'en-tête ICMP
* 56 octets de remplissage

Vous pouvez cependant envoyer des paquets plus grands avec l'option -s. Avec `-s 1472` vous obtiendrez un datagramme IP de 1500 octets, ce qui est le maximum avant fragmentation pour la plupart des réseaux (MTU = unité de transmission maximale)

Ce mécanisme simple peut être utilisé pour déboguer toutes sortes de problèmes, et permet même de faire la distinction entre les délais de transmission et les délais de propagation.

Choisissez un hôte localisé à quelques sauts de vous. Tapez d'abord :

```
$ traceroute nsrc.org
```

Cherchez maintenant une machine se trouvant à plus de deux sauts et notez son adresse IP. Pourquoi ? Parce qu'un saut correspond à votre routeur virtuel et que cet exercice ne donnera pas de résultats fiables avec un matériel virtuel. Le second saut correspond au routeur de passerelle de notre réseau privé. Il est trop près et la différence en termes de temps de ping sera probablement trop faible pour être utile. Nous allons désigner la machine vers laquelle vous avez choisi d'envoyer un ping en tant que PING_MACHINE.

Envoyez 20 pings standard à cette adresse :

```
$ ping -c20 PING_MACHINE
```

Notez le temps d'aller-retour (RTT) *moyen* relevé (t1).

Envoyez maintenant 20 pings de taille maximale :

```
$ ping -c20 -s1472 PING_MACHINE
```

Notez à nouveau le temps d'aller-retour (RTT) *moyen* relevé (t2).

Le délai de propagation est le même dans les deux cas, par conséquent le temps aller-retour plus long doit être dû au délai de transmission.

Vous pouvez désormais estimer le délai de transmission et donc la bande passante de la liaison entre deux points

```
    Allongement du temps de transmission   =  t2 - t1
    augmentation du nombre de bits envoyés           =  (1500-84) * 8 * 2
= 22656
```

(Multipliez par 2 car le temps d'aller-retour implique d'envoyer le paquet deux fois)

Répartissez le nombre de bits par le temps pour obtenir une estimation du débit (bits par seconde). N'oubliez pas de convertir d'abord les millisecondes en secondes.

Exemple :

t2 = 1,71
t1 = 1,14

t2-t1 = 0,57

0.57 ms = 0,00057 sec

22656 bits / 0,00057 sec = 39747368,42 bps

Vous pouvez ensuite convertir le résultat en Kbps, Mbps, etc.

En procédant de même pour les sauts suivants, il est possible d'estimer la bande passante sur chaque saut, même les plus éloignés de vous. Il existe un outil, nommé "pathchar", qui fait cela automatiquement, mais vous devez le compiler à partir des fichiers sources. Quelques outils binaires spécifiques au SE sont disponibles à l'adresse :

ftp://ftp.ee.lbl.gov/pathchar/

La page Web comprenant la documentation est accessible à l'adresse suivante :

http://www.caida.org/tools/utilities/others/pathchar/

Exercices Partie II
==========

Analyse du réseau
---

1. lsof et netstat

Vérifiez les services présents sur votre machine. Vous pouvez pour cela vous baser sur la présentation.

Ou voir avec "man lsof", "man netstat", "lsof -h" et "netstat -h" les options disponibles (elles sont nombreuses !). Pensez à utiliser sudo lorsque vous utilisez lsof et netstat afin de vous doter des permissions nécessaires pour tout voir.

Il vous faudra peut-être installer lsof. Pour cela, tapez :

```
$ sudo apt-get install lsof
```

* avec lsof, déterminez les services IPv4 qui écoutent sur votre machine.
* avec netstat, déterminez les services IPv4 et IPv6 qui écoutent sur votre machine.

Lorsque vous exécutez lsof et netstat, vous devez le faire en tant que root :

```
$ sudo lsof
$ sudo netstat
```

Rappel : vous devrez spécifier des options pour déterminer les services IPv4 et IPv6 qui s'exécutent sur votre machine.

2. tcpdump et tshark

Vous devez d'abord installer ces deux programmes :

```
$ sudo apt-get install tcpdump tshark
```

Utilisez tcpdump comme suit :

```
$ sudo tcpdump -i lo -A -s1500 -w /tmp/tcpdump.log
```

Vous pouvez maintenant initier du trafic sur votre interface de bouclage avec un autre terminal. C'est-à-dire ouvrir une autre session ssh sur votre PC/machine virtuelle.

Par exemple :

```
$ ping localhost
$ ssh localhost
```

etc. Appuyez ensuite sur CTRL-C pour mettre fin à la session tcpdump.

Remarque : ssh donne des résultats beaucoup plus "intéressants". Lisons maintenant avec shark les résultats de tcpdump :

```
$ sudo tshark -r /tmp/tcpdump.log | less
```

Que voyez-vous ? Pouvez-vous suivre la session SSH que vous avez lancée plus haut ?

Nous allons ensuite utiliser ftp. Vous devez d'abord installer un client ftp :

```
$ sudo apt-get install ftp
```

Essayez maintenant ceci :

```
$ sudo rm /tmp/tcpdump.log
$ sudo tcpdump -i eth0 -A -s1500 -w /tmp/tcpdump.log
```

Sur un autre terminal :

```
$ ftp limestone.uoregon.edu

Connected to limestone.uoregon.edu.
220 FTP Server ready.
Name (limestone.uoregon.edu:sysadmin): anonymous
Password: <anything you want>
ftp> exit
```

Mettez fin à la session tcpdump sur l'autre terminal (CTRL-C) et observez maintenant le contenu du fichier log :

```
$ sudo tshark -r /tmp/tcpdump.log | less
```

Voyez-vous votre mot de passe ? En présence d'un gros trafic sur votre réseau, il se peut que le fichier tcpdump.log soit volumineux. Vous pouvez rechercher votre session FTP en tapant :

"/FTP"

dans l'écran de résultat. Étant donné que vous avez redirigé le flux de votre commande shark sur la commande "less" avec "/", la recherche de chaînes fonctionne. Appuyez maintenant sur la touche "n" pour "next" (suivant) afin de suivre la session FTP. Vous devriez voir une ligne comportant la chaîne suivante :

"FTP Request: PASS PasswordYouTypedIn"

Ce type d'outil permet de détecter très facilement des mots de passe cryptés sur des réseaux LAN sans fil.

N'oubliez pas de nettoyer après votre passage :

$ rm /tmp/tcpdump.log

3. tcpdump partie II

Vous pouvez également utiliser tcpdump comme un outil d'investigation informatique en temps réel. Il faudrait plusieurs heures de classe pour traiter totalement tcpdump, mais nous allons commencer avec un autre exemple concret.

Nous allons examiner une requête dhcp partant de votre PC et les réponses qu'elle reçoit.

Connectez-vous d'abord à votre image PC et devenez utilisateur root :

```
$ sudo -s
```

Ensuite, nous allons utiliser un utilitaire du nom de "screen" :

```
# apt-get install screen
```

Maintenant, exécutez screen :

```
# screen
```

À ce stade, vous pouvez avoir plusieurs sessions terminal ouvertes dans une seule fenêtre ssh. Lancez le processus tcpdump d'écoute de requêtes dhcp :

```
# tcpdump -s0 -ni eth0 port 67 or port 68
```

Avec screen, ouvrez maintenant un autre "écran" dans votre fenêtre terminal.

Faites ctrl-a c

Pour comprendre l'action de "-s0", "-n" et "-i", vous pouvez vous référer à la page de manuel tcpdump :

man tcpdump

Recherchez "-s" en tapant "/" et ensuite "-s", puis appuyez sur ENTRÉE. Appuyez sur "n" pour voir l'occurrence suivante de la chaîne "-s".

Maintenant lancez sur votre machine une requête dhcp portant sur une nouvelle adresse eth0 :

dhcpclient

Revenez à l'écran précédent pour voir ce qu'affiche tcpdump :

Appuyez sur "ctrl-a p" ("p" pour Précédent, "n" pour "next" (suivant) pour faire défiler les écrans)

Vous devriez obtenir un résultat comme celui-ci :

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
18:03:05.003190 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request
from 52:54:4a:5e:68:77, length 300
18:03:05.004349 IP 10.10.0.254.67 > 10.10.0.250.68: BOOTP/DHCP, Reply,
length 300
```

Pour arrêter la session tcpdump tapez "ctrl-c"

Savez-vous ce que cela signifie ? Pourquoi avons-nous spécifié d'écouter les ports 67 et 68 ? Si vous ouvrez le fichier /etc/services, vous trouverez les définitions suivantes pour les ports 67 et 68

```
bootps          67/udp     # Bootstrap Protocol Server
bootps          67/tcp     # Bootstrap Protocol Server
bootpc          68/udp     # Bootstrap Protocol Client
bootpc          68/tcp     # Bootstrap Protocol Client
```

Vous pouvez revenir à l'écran où vous avez exécuté dhcpclient et quittez l'écran si vous le souhaitez :

ctrl-a-n

Puis tapez :

exit

Si l'utilitaire screen et son fonctionnement vous intéressent, visitez :

http://www.howtoforge.com/linux_screen

ou posez des questions à votre instructeur.

4. Utilisation d'iperf

Installez tout d'abord iperf :

```
$ sudo apt-get install iperf
```

Utilisez "man iperf" ou "iperf -h" pour obtenir de l'aide.

Demandez à votre voisin d'exécuter :

```
$ iperf -s
```

Connectez-vous à la machine de votre voisin avec :

```
$ iperf -c ipNeighbor
```

Si vous ne connaissez pas l'adresse IP de la machine de votre voisin, demandez-lui de faire :

```
$ ifconfig eth0
```

et de vous indiquer quelle adresse IP utilise sa machine.

Quel est le débit entre vos machines ? Vous pouvez répéter cet exercice avec n'importe quelle machine distante où iperf est installé et sur laquelle vous avez un compte. C'est un moyen rapide de voir comment se présente la bande passante entre deux points.

Pour arrêter le serveur iperf sur lequel vous avez exécuté "iperf-s", appuyez sur CTRL-c.

Si vous en avez le temps, continuez d'explorer les options iperf. Si vous disposez d'un PC distant fonctionnant sous UNIX ou Linux, vous pouvez essayer d'installer iperf et de tester la connexion entre le laboratoire de l'atelier et votre machine distante.

Autres aspects à explorer...

* Testez TCP dans différentes tailles de fenêtres (-2).

* Vérifiez TCP MSS (-m). Quel en est l'incidence sur le débit ? Qu'entend-on par découverte du MTU d'un chemin ?

* Testez avec deux threads parallèles (-P) et comparez les totaux. Constatez-vous une différence ? Pourquoi ?

* Testez avec différentes tailles de paquets et l'option TCP_NODELAY (-N).