ALLARD
Christophe

3ème année
Licence informatique

RAPPORT DE STAGE

Association Tetaneutral.net

Stage du 15 avril au 7 juin

2012 / 2013

Remerciements

Je tiens à remercier M. Laurent GUERBY de m'avoir chaleureusement accueilli au sein de son association pendant la durée de mon stage mais a l'association Tetaneutral.net.

Je tiens à remercier dans un premier temps, toute l’équipe pédagogique de la licence informatique de l'Université Paul Sabatier et les intervenants professionnels responsables de la formation, pour avoir assuré la partie théorique de celle-ci.

Sujet du stage

Définition d'un protocole de test et développement des outils associés pour évaluer la performance en charge réseau de routeurs logiciels libres sur plates-formes physiques et virtuelles. L'objectif est de connaître les limites de ces technologies en particulier lors d'attaques par déni de service pour pouvoir les détecter et réagir.

Sommaire

Introduction

L'internet héberge un grand nombre de services dont le plus connu est le World Wide Web. Ces services sont des logiciels fonctionnant sur des ordinateurs connectés entre eux par Internet, du plus modeste au domicile d'un internaute aux plus puissantes machines regroupées par milliers dans des centres de traitement de données. Mais quelque soit la structure, des failles et des problèmes sur le réseau peuvent survenir et provoquer un dysfonctionnement. Tetaneutral.net n'échappe pas à cette règle et doit se protéger de ces défaillances, volontaires ou non.

PARTIE I

Présentation de l'association et analyse de l'existant

1) Tetaneutral.net

Les fournisseurs d'accès à internet, les hébergeurs et les opérateurs relient les utilisateurs du réseau entre eux, et avec des fournisseurs de services. Ces intermédiaires ont techniquement la capacité de discriminer les informations transmises par et vers l'internaute en fonction de la source, de la destination ou du contenu de l'information partagée sur le réseau. Or pour le bon fonctionnement de l'internet il est important que ces acteurs respectent la "neutralité du réseau" et s'interdisent de telles discriminations. De puissants intérêts commerciaux et politiques sont en jeu derrière cette simple notion. C'est en quoi veille Tetaneutral.net.

Tetaneutral.net est une association qui relève de la loi du 1er juillet 1901 et du décret du 16 août 1901 (association à but non lucratif). Elle est membre fondatrice de la Fédération FDN (French Data Network) qui regroupe des fournisseurs d'accès à internet associatifs ayant pour valeurs communes bénévolat, solidarité, fonctionement démocratique ainsi que défense et promotion de la neutralité du net.

Tetaneutral.net ainsi que la FFDN agissent pour défendre des valeurs telles que la liberté d'expression sur internet, la gestion participative et coopérative du réseau ainsi que la diffusion de la connaissance sur le fonctionnement du Web. Ces structures comptent, non pas sur le nombre d'abonnés à Internet, mais bien sur leur nombre d'adhérents afin de communiquer leur message.

Avec le dévelopement d'internet et la quasi-nécessité de son utilisation nous sommes obligés aujourd'hui d'y avoir accès quelque soit la démarche que l'on souhaite effectuer. Les moyens que Tetaneutral.net détient permettent de fournir une connexion internet gratuitement aux personnes dans le besoin comme le CREA de Toulouse et leur Centre Social Autogéré. Il permet aussi d'apporter un accès à internet dans les zones blanches ADSL comme autour de Saint-Gaudens et Monès. Le principe de tarification appliqué par l'association se base sur une participation libre mais nécessaire et sur une grille tarifaire basée sur les moyens de ses membres.

L'objectif de l'association est d'obtenir ainsi le statut de LIR (Local Internet Registry).

2) Analyse du réseau

Tetaneutral.net utilise tous les moyens à sa disposition afin d'offir un accès à internet à ses membres. Via FDN, l'association peut offrir un accès ADSL en dégroupage partiel.

La structure du réseau est quelque peu complexe car celui-ci est composé d'un ensemble de plus petits réseaux afin de créer un maillage pour recouvrir le plus de territoire possible.

Entre le local de l'association et le centre réseau voisin, une fibre optique a été mise en place. Les utilisateurs accèdent alors au local via une liaison radio en réseau redondant déployé à travers Toulouse et ses alentours, et basé sur les travaux de Toulouse Sans Fil qui avait déjà déployé un réseau hertzien en utilisant la technologie AirMax d'Ubiquity. Ces antennes sont pour la plupart des antennes de transmission basées sur une fréquence de 5 Ghz avec un débit pouvant aller jusqu'a 100 Mbit/s. Chaque antenne directionnelle peut desservir jusqu'à 8 km sans perte importante de données (voir annexe 1 et 2).

Dans le cadre d'une évolution future, l'association regarde et étudie la question du déploiement de la fibre optique chez ses membres. Actuellement, elle leur permet de louer leur matériel d'accès au réseau radio sans fil.

Tetaneutral.net développe aussi son réseau en investissant dans des datacenter permettant un débit bien meilleur et de la redondance sur le réseau en fibre optique.

En plus d'être un fournisseur d'accès à Internet, Tetaneutral.net propose aussi d'autres services tels que l'hébergement de machine physique quelque soit le format de la machine ou la mise à disposition de machine virtuelle hébergée sur leurs serveurs.

Aujourd'hui, la plupart des antennes de relais fonctionnent sous le système OpenWRT. Les systèmes d'exploitation des serveurs peuvent varier mais pour la plupart, ils sont sur une distribution telle que Debian ou Ubuntu.

Toute cette structure et cette organisation font la force de Tetaneutral.net, mais aussi sa faiblesse. En effet, une telle structure peut présenter des points faibles.

Par exemple, lors de la journée du 29 juin 2012, l'association à été victime d'une attaque de type déni de service (Denial of Service, DoS). Afin de pallier cette difficulté à l'avenir, mon stage porte sur l'analyse des différentes méthodes et outils de diagnostic réseau et la mise en place de protocoles de test complet de l'ensemble du réseau Tetaneutral.net.

PARTIE II

Protocole de test de performance

1) Les outils de diagnostic

Afin de s'assurer du bon fonctionnement de l'ensemble du réseau, des diagnostics doivent être effectués régulièrement. Il existe plusieurs outils qui traitent parfaitement cette tâche mais tous ne sont pas gratuits, faciles d'utilisation ou suffisament complets. Afin de proposer, dans l'optique de mon stage, des outils de diagnostic, voici quelques exemples d'utilitaires :

A) Les outils de base

L'utilisation d'outils perfectionnés est importante afin d'effectuer un diagnostic complet, mais il existe déjà des petits utilitaires se trouvant sur la plupart des systèmes d'exploitation qui permettent d'obtenir une première approche du problème :

- Ping : l'outil le plus communément utilisé pour tester rapidement une connectivité entre un hôte et une destination.

- tracert/traceroute : outil standard de détermination de route d'une machine à une autre permettant d'afficher le chemin suivi par la connexion à travers différentes passerelles.

- ipconfig/ifconfig : une bonne partie des problèmes vient principalement d'une mauvaise configuration réseau, cet utilitaire est là pour vérifier l'état des interfaces.

- nslookup : moyen rapide de résoudre un nom de domaine et ainsi voir l'état de son DNS.

- netstat : utilitaire sous Linux permettant de voir l'état des connexion réseau, on peut ainsi déterminer les ports utilisés et vérifier qu'un serveur écoute bien sur le bon port.

- route : petit outil qui vous affiche l'état de la table de routage et vous permet de les changer.

(voir dans les annexes 3 et 4 des exemples de fonctionnement des programmes)

B) IPerf

IPerf est un logiciel sous licence libre (GPL) permettant la mesure de différentes variables d'une connexion réseau IP. Développé par le National Laboratory for Applied Network Research (NLANR), il est basé sur une architecture client / serveur et est disponible sur la plupart des systèmes d'exploitation (Unix, MacOS X, Windows, etc.)

C'est un outil très simple d'utilisation qui va permettre de :

- mesurer la bande passante,
- apporter certaines informations concernant la taille des paquets, etc.,
- proposer des tests de performance aussi bien en TCP qu'en UDP.

Grâce à sa simplicité d'utilisation, on peut facilement l'inclure dans des scripts et permettre ainsi une gestion un peu plus poussée de ce programme afin de trouver les goulots d'étranglement sur un réseau complexe tel que celui de Tetaneutral.net.

(voir dans l'annexe 5 l'exemple d'utilisation de IPerf)

C) NetPerf

NetPerf est un logiciel lui aussi sous licence GPL développé par Hewlett Packard, permettant de simuler du trafic de données entre deux points d'un réseau.

Contrairement aux logiciels tel que IPerf vu précédement, NetPerf ne se limite pas à une simple mesure de la bande passante ou de la taille des paquets. Il peut en effet agir sur la taille du buffer aussi bien en réception quand émission. Cela permet de trouver le réglage optimal afin de configurer le système.

(voir dans l'annexe 6 l'exemple d'utilisation de NetPerf)

Une fois les différents diagnostics effectués et les serveurs configurés, ces derniers seront continuellement surveiller grâce à un ensemble de logiciels de monitoring.

2) Les outils de monitoring

Le monitoring permet de surveiller en temps réel l'état et le comportement des serveurs composant un réseau. La aussi, le choix des logiciels est très important afin de trouver les logiciels adaptés et de les deployer.

A) Les outils de base

Comme pour les outils de diagnostic, il existe des utilitaires simples pour surveiller l'état d'une machine. Ici nous allons présenter quelques outils disponibles sur les plateformes Linux (il existe pour la plupart leur équivalent sous les autres systèmes d'exploitations) :

- ps : cette commande permet d'afficher un image instantanée de l'état actuel des processus.

- top : ce programmes propose une vue en temps réel des processus actuels avec une possibilité de trier en fonction des paramètres demandés.

- vmstat : il apporte des informations sur les processus, la mémoire, la pagination, les entrés sorties et l'activité du processus.

- free : il affiche l'état de la mémoire en ram et de la mémoire mise en swap.

- iostat : cette commande affiche les statistiques du CPU et les statistiques d'entrée-sortie des différents matériels, des partitions, du réseau et des systèmes de fichiers.

B) Multi Router Traffic Grapher (MRTG)

MRTG est un programme qui permet de créer des graphiques de l'activité de l'ordinateur. Ces graphiques sont sauvegardés sous forme de fichiers HTML. Le but est de les rendre accessibles depuis l'extérieur en utilisant un serveur web, comme Apache, ce qui permet de surveiller l'activité de l'ordinateur à distance, depuis un simple navigateur web. Il utilise le protocole Simple Network Management Protocol (SNMP) pour interroger les équipements réseaux tels que des routeurs, des commutateurs, ou bien encore d'autres serveurs disposant d'une Management Information Base (MIB).

C) Network Top (NTop)

Ntop (Network Top) est un outil open-source sous licence GPL de supervision réseau. C'est une application qui produit des informations sur le trafic réseau en temps réel (comme pourrait le faire la commande top avec les processus). Il capture et analyse les trames d'une interface donnée en utilisant libpcap. Il permet d'observer une majeure partie des caractéristiques du trafic réseau en entrée et sortie à travers une interface web et un mode interactif. NTop utilise la bibliothèque libpcap.

D) TCPDump / Wireshark

TCPDump est un analyseur de paquets en ligne de commande. Wireshark est quant à lui un outil comparable à TCPDump avec une interface graphique. Ils permettent d'obtenir le détail du trafic visible depuis une interface réseau. Distribués pour la plupart des systèmes d'exploitation, ils dépendent de la bibliothèque libpcap. Ils permettent une analyse plus fine du réseau grâce à un ensemble de filtres. Combinés à d'autres utilitaires, on peut obtenir un monitoring complet de ces interfaces .

E) Bibliothèque libpcap

Pcap (pour "packet capture") est une interface de programmation permettant la capture d'un trafic réseau. Il existe un portage de cette bibliothèque sous Windows appelé WinPcap.

C'est une bibliothèque très utilisée dans les logiciels d'analyse de trafic réseau. Elle est plutôt difficile d'utilisation mais permet de programmer un ensemble de logiciels adaptés au besoin de chaque système et réseau.

Il existe aussi de nombreux logiciels utilisant libpcap (par exemple NTop, TCPDump ou bien Wireshark précédement cités). Cela peut aller du simple analyseur de trafic et de protocole à un monitoring complet des réseaux du système en passant par la génération de trafic et la détection d'intrusion.

Tous ces outils permettent de prévenir et de détecter les problèmes mais pas de les corriger pour autant. Le développement d'application et de structure spécifique est nécessaire au bon fonctionnement du réseau.

PARTIE III

Présentation d'une solution d'amélioration de performance : NetMap

1) NetMap

Lorsque qu'un serveur route diverses connexions réseaux entre elles, il va passer par un ensemble d'analyses gérées par le noyau. Ces analyses s'effectuent non pas sur la quantité de données reçues, mais sur la quantité de paquets reçus. Tetaneutral.net fut vitctime d'une attaque qui ne cherchait pas à saturer la bande passante, mais à saturer l'utilisation du processeur dans le traitement de petits paquets qui arrivaient en grande quantité.

Pour palier cette faille, une solution en cours de dévelopement est sur le point de résoudre le problème. Elle est mise au point à l'Université de Pise en Italie et se prénome NetMap. Mon travail a été de comprendre et de tester les capacités de cette solution.

NetMap est un framework de gestion rapide des paquets réseaux en entrée et sortie. Il utilise quelques techniques d'amélioration de performance qui fait de lui un outil très puissant et très rapide dans son fonctionnement.

Son principal atout réside dans le contournement du noyau lors de l'analyse des paquets (voir annexe 7). En effet, il fait la liaison directement entre le matériel et les applications. Il permet ainsi un gain de performance car l'utilisation du noyau est onéreuse en temps. Il protège aussi le système des crashs possibles car il est dans l'espace mémoire réservé aux utilisateurs et donc n'a pas accès aux ressources critiques du système.

Par la suite, le traitement des paquets se fait par une application native compilée à partir d'une API NetMap qui apporte l'interface nécessaire à l'utilisation de ses capacités.

2) Mise en pratique

L'utilisation de ce framework est plutôt simple, contrairement à libpcap par exemple. Avec un minimum de documentation, on peut coder un petit programme de générateur de paquets (exemple du site internet : voir annexe 8). Cette simplicité fait de NetMap un outil léger et puissant et lui permet de s'adapter à la plupart des situations. Une partie de mon stage a été de tester les performances de NetMap dans certaines situations.

Mais avant d'effectuer les tests, quelques étapes sont nécessaires. Il a fallu dans un premier temps recompiler un noyau (tests effectués sur un noyau FreeBSD). NetMap n'est pas qu'un simple programme et nécessite une modification du noyau.

Le noyau compilé avec NetMap va implémenter un nouveau matériel se trouvant dans /dev/netmap qui sera utilisé dans le programme à travers une API NetMap. C'est ce matériel qui va emuler et gérer tout le fonctionnement et permettre d'outre-passer le noyau dans son utilisation. Le langage de programmation utilisé est le C.

3) Résultats et performances

Suite à de nombreux essais sur différentes machines (physiques ou virtuelles) mises à ma disposition, j'ai pu constater le gain de performance de NetMap comparé à l'utilisation standard d'un noyau. J'ai effectué deux tests différents :

- transfert de gros fichiers fragmentés en petits paquets,
- routage de données entre deux interfaces.

La différence de performances va du simple au quadruple pour la plupart de mes tests (voir annexe 9). Les résultats annoncés par les différents tests trouvés sur internet montrent des variations tout aussi intéressantes (voir annexes 10, 11 et 12). Il est même annoncé que NetMap peut générer un trafic de plus de 14 millions de paquets par seconde sur du 10 Gbit/s en éthernet juste avec un processeur simple coeur cadencé à 900 Mhz.

Ces chiffres nous offrent beaucoup d’espoir quant à ce logiciel qui pourrait bien être une future évolution de nos systèmes d’exploitation en étant proposé dans les versions standard de Linux.

Conclusion

Le stage à l'association Tetaneutral.net a été très instructif : j'ai pu approché un réseau complexe, j'ai pu touché à quelque chose qui se rapprochait d'internet. Il m'a permis de remettre en question ma vision du fonctionnement d'un réseau informatique en mettant l'accent sur la sécurité, la surveillance des machines et le développement d'applications ciblées sur une structure définie.

Ce stage a satisfait ma curiosité envers le fonctionnement d'un fournisseur d'accès à internet et m'a permis d'appréhender de nouvelles problématiques dans le domaine des réseaux à l'échelle mondiale. J'ai beaucoup apprécié le caractère presque humanitaire de l'association qui a pour but de donner accès à internet à tous. Mon seul regret a été que le stage ne dure pas plus longtemps : je n'ai pu qu'effleurer cette immense architecture qu'est Internet.

Comme j'ai pu m'en apercevoir durant mon stage, les nouvelles technologies sont très prometteuses grâce à des outils de diagnostic et de monitoring de plus en plus perfectionnés qui nous permettent d'améliorer la connectivité à travers le monde.

Sources et références

Association Toulouse Sans Fil : http://www.toulouse-sans-fil.net/

Association Tetaneutral.net : http://tetaneutral.net/

Technologie AirMax d'Ubiquity : http://www.ubnt.com/airmax

Site d'IPerf : http://iperf.fr/

Site de NetPerf : http://www.netperf.org/netperf/

Site de MRTG : http://www.mrtg.com/

Site de NTop : http://www.ntop.org/

Site de TCPDump / Libpcap : http://www.tcpdump.org/

Site de Wireshark : http://www.wireshark.org/

Site de WinPCap : http://www.winpcap.org/

Projet NetMap : http://info.iet.unipi.it/~luigi/netmap/

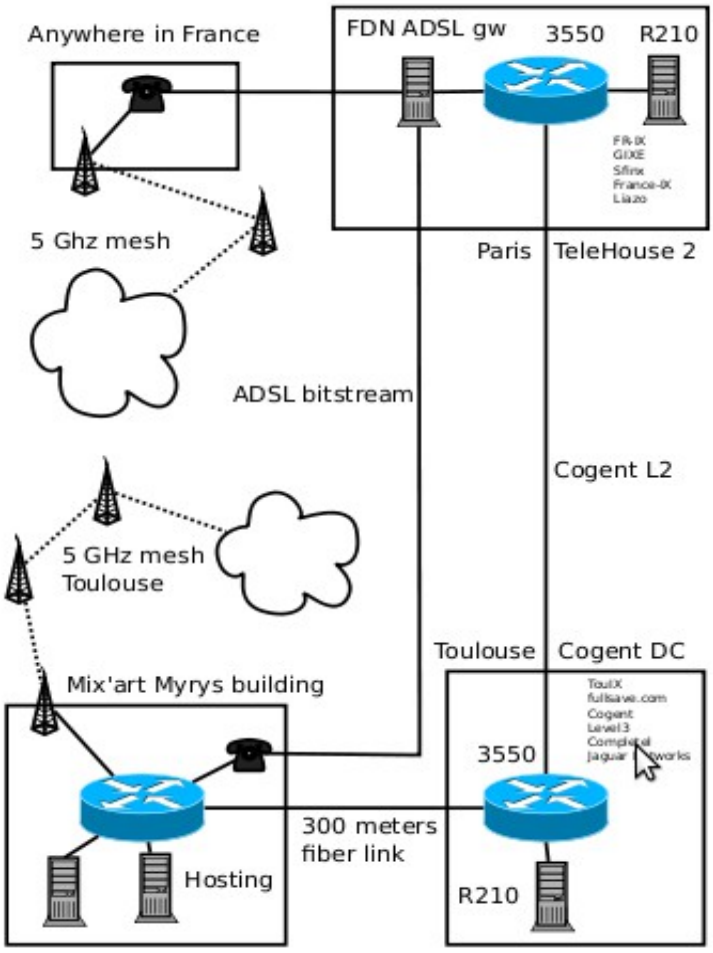
Article d'explication sur NetMap : http://queue.acm.org/detail.cfm?id=2103536

Fonctionnement de NetMap :

http://info.iet.unipi.it/~luigi/papers/20110815-sigcomm-poster.pdf

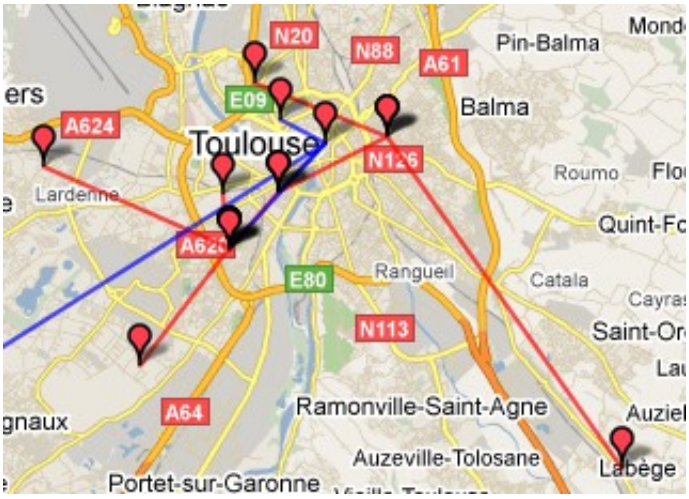
Autre recherche sur Wikipedia : http://fr.wikipedia.org

Annexes



Annexe 1

Schéma de représentation du réseau Tetaneutral.net



Annexe 2

Carte du maillage des antennes principales de Tetaneutral.net

```
user@serveur:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=48 time=6.48 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=48 time=6.59 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=48 time=7.03 ms
```

```
user@serveur:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.211 ms  0.174 ms  0.219 ms
 2  4k1-33.vty.named.com (x.x.x.1)  0.670 ms  0.791 ms  0.876 ms
 3  88.191.1.245 (88.191.1.245)  0.988 ms  1.139 ms  2.468 ms
 4  a9k1-1013.dc3.online.net (88.191.1.132)  0.889 ms  1.032 ms  1.015 ms
 5  a9k1-1012.dc1.online.net (88.191.1.130)  1.731 ms  1.728 ms  1.789 ms
 6  bzn-crs16-1-be1500-p.intf.routers.proxad.net (212.27.50.161)  1.967 ms  2.157 ms  2.101 ms
 7  cbv-9k-1-be1001.intf.routers.proxad.net (212.27.59.5)  1.243 ms  1.478 ms  1.539 ms
 8  72.14.216.98 (72.14.216.98)  2.510 ms google-pni-3.routers.proxad.net (212.27.40.102)  1.083 ms  1.163 ms
 9  72.14.238.228 (72.14.238.228)  1.347 ms
10  72.14.235.169 (72.14.235.169)  1.630 ms 72.14.235.173 (72.14.235.173)  1.615 ms 72.14.235.169 (72.14.235.169)  1.587 ms
11  216.239.43.233 (216.239.43.233)  6.436 ms  6.724 ms  6.543 ms
12  72.14.238.43 (72.14.238.43)  6.663 ms 72.14.236.191 (72.14.236.191)  6.800 ms 72.14.238.215 (72.14.238.215)  6.630 ms
13  * * *
14  google-public-dns-a.google.com (8.8.8.8)  6.524 ms  6.475 ms  6.596 ms
```

```
user@serveur:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:3e:21:e1:23
          inet addr:x.x.x.x  Bcast:88.190.255.255  Mask:255.255.0.0
          inet6 addr: fe80::216:3eff:fe21:e123/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18080 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3014 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1319712 (1.2 MiB)  TX bytes:441241 (430.8 KiB)
          Interrupt:245

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:168 (168.0 B)  TX bytes:168 (168.0 B)
```

```
user@serveur:~$ nslookup google.fr
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.fr
Address: 173.194.34.24
Name:   google.fr
Address: 173.194.34.31
Name:   google.fr
Address: 173.194.34.23
```

Annexe 3

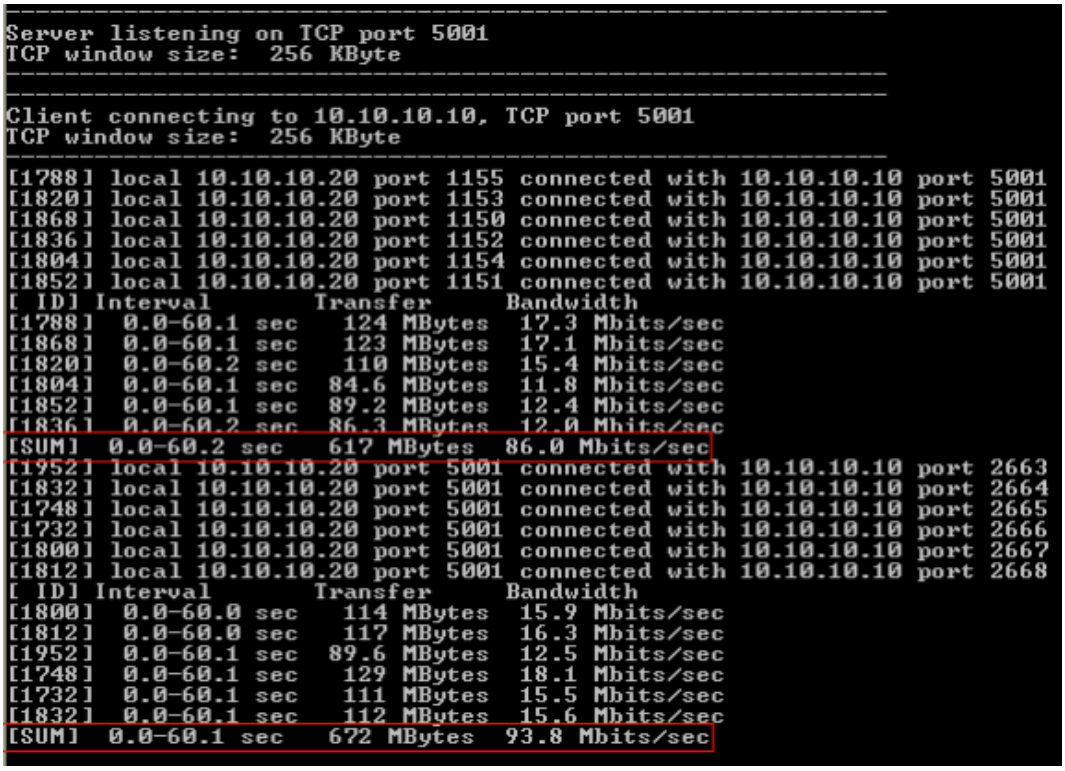
Exemple d'utilisation de Ping, Traceroute, Ifconfig et Nslookup

```
user@serveur:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0     52 server:ssh    x.x.x.x:57456 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  18     [ ]         DGRAM                    3865     /dev/log
unix  2      [ ]         DGRAM                    6035     @
unix  2      [ ]         DGRAM                    3867     /var/spool/postfix/dev/log
unix  3      [ ]         STREAM     CONNECTED     10619    /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     10618
unix  2      [ ]         DGRAM                    10615
unix  2      [ ]         DGRAM                    10253
unix  3      [ ]         STREAM     CONNECTED     10067
unix  3      [ ]         STREAM     CONNECTED     10066
unix  3      [ ]         STREAM     CONNECTED     9991     /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9990
unix  3      [ ]         STREAM     CONNECTED     9981     /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9980
unix  3      [ ]         STREAM     CONNECTED     9958     /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9957
unix  2      [ ]         DGRAM                    9947
unix  3      [ ]         STREAM     CONNECTED     9945     /var/run/dbus/system_bus_socket
```

```
user@serveur:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
x.x.x.0         *               255.255.0.0     U     0      0        0 eth0
default         x.x.x.1         0.0.0.0         UG    0      0        0 eth0
```

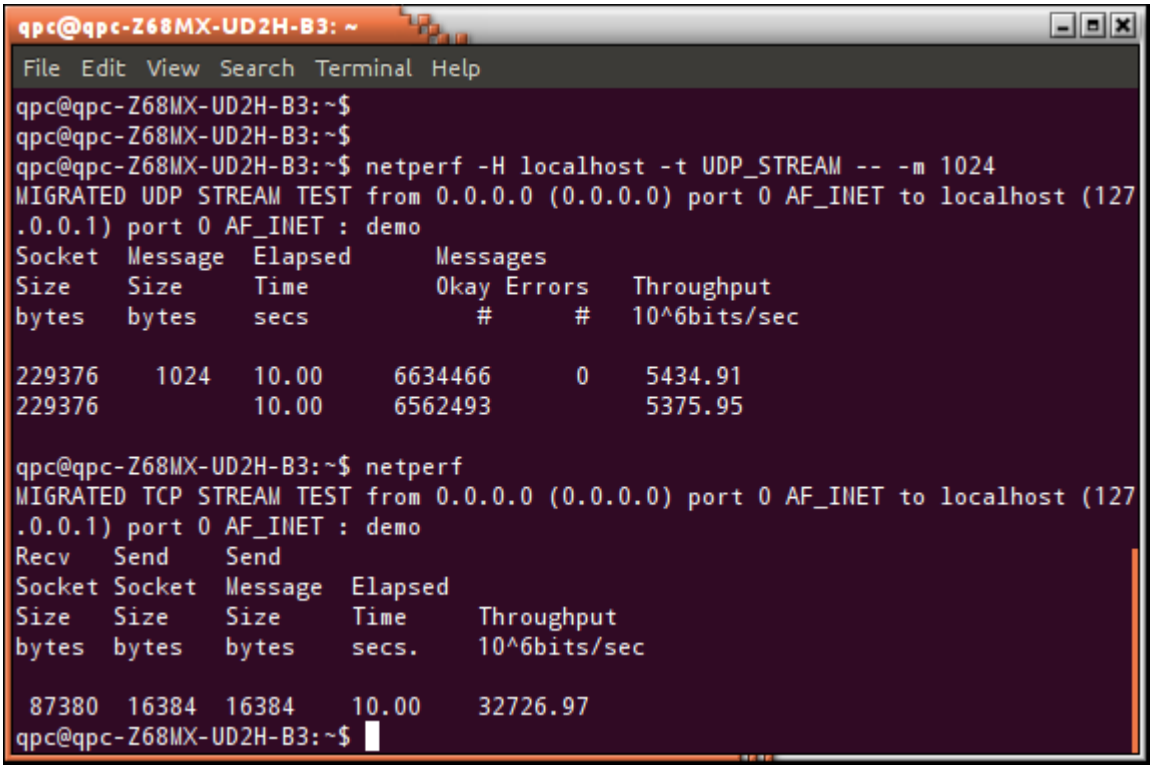
Annexe 4

Exemple d'utilisation de netstat et de route



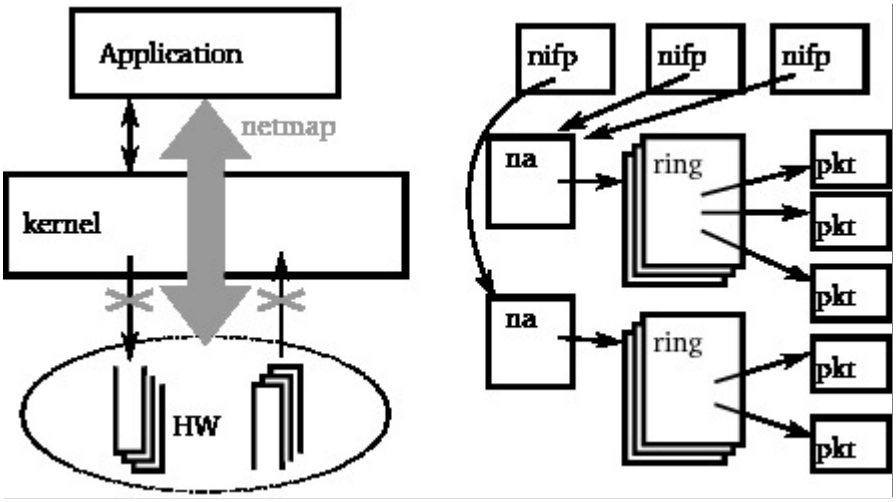
Annexe 5

Exemple d'utilisation de IPerf



Annexe 6

Exemple d'utilisation de NetPerf



Annexe 7

Schéma de l'organisation et de la structure de NetMap

Sample code for a packet generator:

```
struct netmap_if *nifp;
struct nmreq req;
char *mem;
struct pollfd fds;
bzero(&req, sizeof(req));
bzero(&fds, sizeof(fds));
fds.fd = open("/dev/netmap", O_RDWR);
strcpy(req.nm_name, "ix0");
ioctl(fds.fd, NIOCREG, &req);
mem = mmap(0, req.memsize, fds.fd);
nifp = NETMAP_IF(mem, req.offset);
fds.events = POLLOUT;
for (;;) {
        poll(fds, 1, -1);
        for (r = 0; r < req.num_queues; r++) {
                struct netmap_ring *ring = NETMAP_TXRING(nifp, r);
                while (ring->avail-- > 0) {
                        int i = ring->cur;
                        char *buf = NETMAP_BUF(ring, ring->slot[i].buf_index);
                        ... prepare packet in buf ...
                        ring->slot[i].len = ... packet length ...
                        ring->cur = NETMAP_NEXT(ring, i);
                }
        }
}
```

Annexe 8

Exemple de code utilisant NetMap

Résultat des tests de NetMap

Machine 1 : 2 core 2.88 Ghz
Machine 2 : 1 core 1.8 Ghz
Machine 3 : 2 core – 4 thread 1,8 Ghz

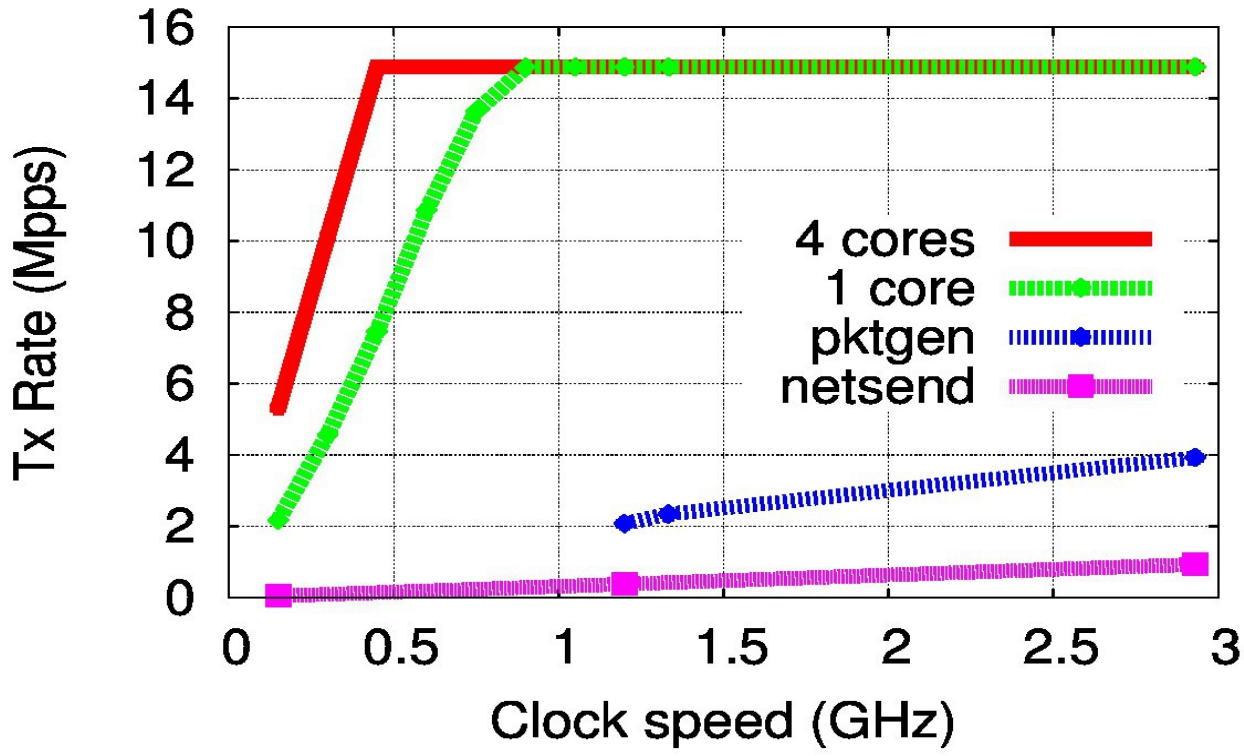
Liaison en 100Mbits/s

Transfère sans NetMap puis avec NetMap (packet 64 bytes) :
M1 => M2 : 32'564 pps => 127kpps
M2 => M1 : 39'425 pps => 135kpps
M1 => M3 : 34'213 pps => 124kpps
M3 => M1 : 38'015 pps => 135kpps
M2 => M3 : 31'956 pps => 123kpps
M3 => M2 : 32'102 pps => 125kpps

Passerelle sans NetMap puis avec NetMap (packet 64 bytes) :
M1 => M2 => M3 : 29'045 pps => 122kpps
M1 => M3 => M2 : 32'962 pps => 126kpps
M2 => M1 => M3 : 32'045 pps => 124kpps
M2 => M3 => M1 : 29'425 pps => 130kpps
M3 => M1 => M2 : 32'574 pps => 134kpps
M3 => M2 => M1 : 30'014 pps => 123kpps

Annexe 9

Résultat des tests avec et sans NetMap

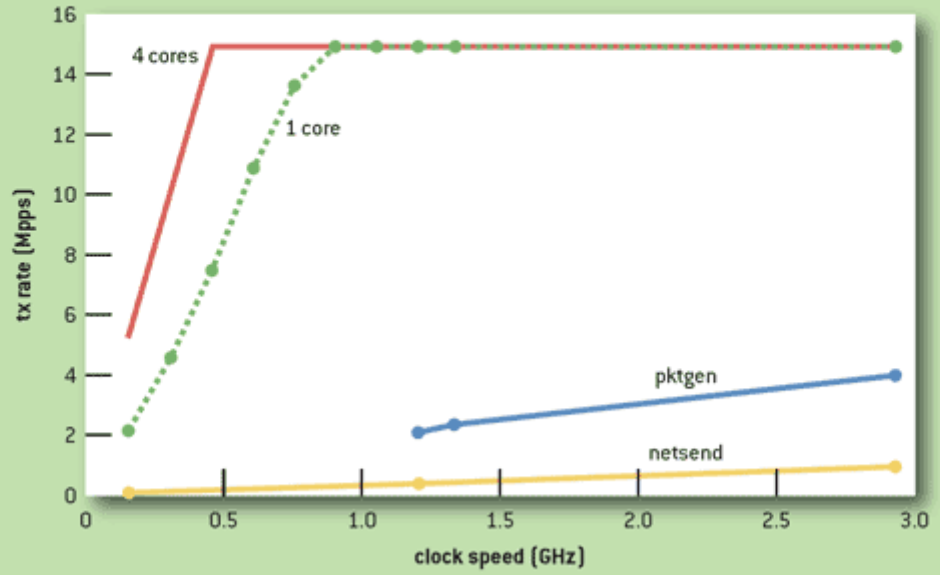


Annexe 10

Nombre de paquets traités en fonction de la vitesse du processeur et du nombre de coeur

FIGURE 4

Packet Generation Speed Using Netmap vs Traditional APIs



Annexe 11

Nombre de paquets traités en fonction de la vitesse du processeur et du nombre de coeur (2)

FIGURE 5

Performance of various applications with and without netmap

Packet forwarding	Mpps
FreeBSD bridging	0.690
netmap + libpcap emulation	7.500
netmap, native	10.660
Open vSwitch	**Mpps**
optimized, FreeBSD	0.790
optimized, FreeBSD + netmap	3.050
Click	**Mpps**
user space + libpcap	0.400
linux kernel	2.100
user space + netmap	3.950

Annexe 12

Nombre de paquets traités en fonction des logiciels utilisé