

Introduction aux Réseaux : TP1

Line POUVARET, Mickaël TURNEL

2015-2016

Compte-Rendu de TP

2.1 Mise en place du réseau

2.1.2 Configuration des machines

Configuration manuelle

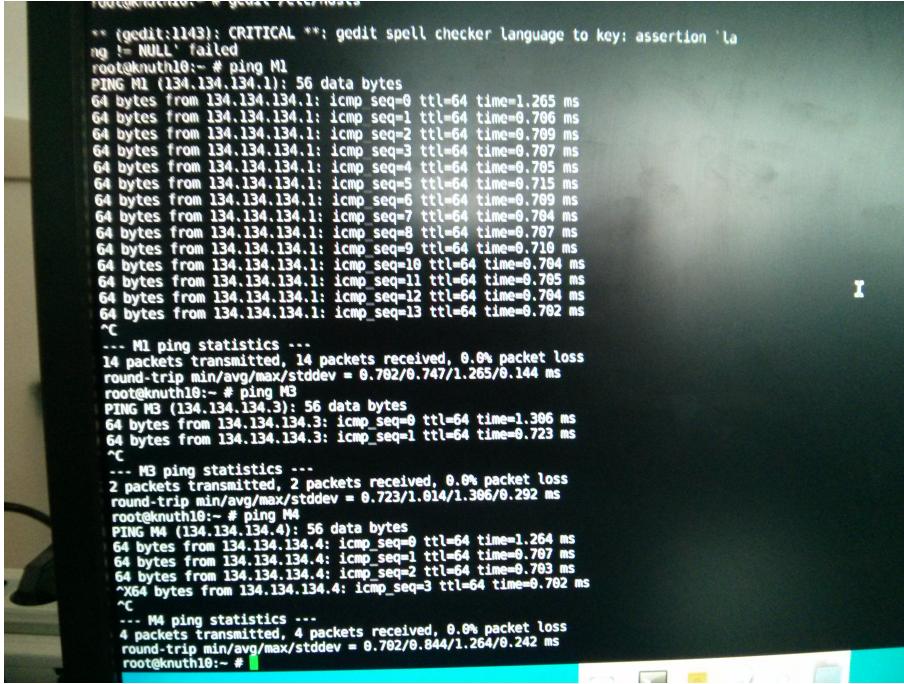
Nous avons choisi comme adresse IP Internet 134.134.134.X, X étant le numéro de machine allant de 1 à 4. Pour chaque machine, nous avons exécuté la commande : ifconfig em0 134.134.134.X up Si on exécute à nouveau ifconfig sur chaque machine, on constate bien que nous obtenons la bonne adresse configurée pour inet et que status est à active.

Nous avons modifié le fichier /etc/hosts de chaque machine de sorte à avoir :
134.134.134.1 M1 134.134.134.2 M2 134.134.134.3 M3 134.134.134.4 M4

2.1.3 Contrôle du réseau

Utilisation du ping

Nous avons effectué une commande ping sur chaque machine vers les trois autres machines du réseau. (ex sur M2 : ping M1, ping M3, ping M4)



```
** (gedit:1143): CRITICAL **: gedit spell checker language to key: assertion `la
ng != NULL' failed
root@knuth10:~ # ping M1
PING M1 (134.134.134.1): 56 data bytes
64 bytes from 134.134.134.1: icmp_seq=0 ttl=64 time=1.265 ms
64 bytes from 134.134.134.1: icmp_seq=1 ttl=64 time=0.796 ms
64 bytes from 134.134.134.1: icmp_seq=2 ttl=64 time=0.799 ms
64 bytes from 134.134.134.1: icmp_seq=3 ttl=64 time=0.797 ms
64 bytes from 134.134.134.1: icmp_seq=4 ttl=64 time=0.795 ms
64 bytes from 134.134.134.1: icmp_seq=5 ttl=64 time=0.715 ms
64 bytes from 134.134.134.1: icmp_seq=6 ttl=64 time=0.799 ms
64 bytes from 134.134.134.1: icmp_seq=7 ttl=64 time=0.794 ms
64 bytes from 134.134.134.1: icmp_seq=8 ttl=64 time=0.797 ms
64 bytes from 134.134.134.1: icmp_seq=9 ttl=64 time=0.710 ms
64 bytes from 134.134.134.1: icmp_seq=10 ttl=64 time=0.794 ms
64 bytes from 134.134.134.1: icmp_seq=11 ttl=64 time=0.795 ms
64 bytes from 134.134.134.1: icmp_seq=12 ttl=64 time=0.794 ms
64 bytes from 134.134.134.1: icmp_seq=13 ttl=64 time=0.792 ms
^C
--- M1 ping statistics ---
14 packets transmitted, 14 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.792/0.747/1.265/0.144 ms
root@knuth10:~ # ping M3
PING M3 (134.134.134.3): 56 data bytes
64 bytes from 134.134.134.3: icmp_seq=0 ttl=64 time=1.396 ms
64 bytes from 134.134.134.3: icmp_seq=1 ttl=64 time=0.723 ms
^C
--- M3 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.723/1.014/1.396/0.292 ms
root@knuth10:~ # ping M4
PING M4 (134.134.134.4): 56 data bytes
64 bytes from 134.134.134.4: icmp_seq=0 ttl=64 time=1.264 ms
64 bytes from 134.134.134.4: icmp_seq=1 ttl=64 time=0.797 ms
64 bytes from 134.134.134.4: icmp_seq=2 ttl=64 time=0.793 ms
64 bytes from 134.134.134.4: icmp_seq=3 ttl=64 time=0.792 ms
^C
--- M4 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.792/0.844/1.264/0.242 ms
root@knuth10:~ #
```

Procédure de login sur une machine distante

Nous avons utilisé l'application telnet pour que la machine M2 communique avec M4 en exécutant la commande suivante : telnet -y M4 En exécutant ifconfig (toujours dans l'application telnet) sur M2, on remarque que l'adresse inet est bien 134.134.134.4 (c'est à dire l'adresse Internet de M4)

```

^C
--- M4 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.702/0.844/1.264/0.242 ms
root@knuth10:~ # telnet M4
Trying 134.134.134.4...
Connected to M4.
Escape character is '^]'.
Trying SRA secure login:
User (root): guest
Password:
[ SRA accepts you ]

FreeBSD/i386 (knuth12.e.ujf-grenoble.fr) (pts/2)

Last login: Tue Sep 29 16:32:16 from M3
FreeBSD 9.3-RELEASE (PAE-KERNELTP) #0: Fri Sep 5 09:31:56 CEST 2015
Reinstall: Mon Sep 7 14:35:29 CEST 2015
*****
ATTENTION !!
- Sur ce système, vous disposez de peu de place dans la partition /
- Pour stocker des données volumineuses, utiliser un répertoire créé
dans la partition /usr : /usr/home/mes_données par exemple
- Ces systèmes ne sont pas sauvegardés, pensez à copier vos résultats
aujourd'hui !
- En fin de TP vous pouvez rétablir la config. originale en appelant le script /var/backups
- Après ça, scp est ton ami.
- Pour démarrer X11, tapez "xinit".
> ping M2
PING M2 (134.134.134.2): 56 data bytes
64 bytes from 134.134.134.2: icmp_seq=0 ttl=64 time=0.686 ms
64 bytes from 134.134.134.2: icmp_seq=1 ttl=64 time=0.688 ms
64 bytes from 134.134.134.2: icmp_seq=2 ttl=64 time=0.688 ms
64 bytes from 134.134.134.2: icmp_seq=3 ttl=64 time=0.688 ms
^C
--- M2 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.686/0.688/0.688/0.001 ms
> ifconfig
bge0: flags=8802<POINTOPOINT,BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=C019b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HNCSUM,TS04,VLAN_HWTSO,LINKSTATE>
    ether 00:18:18:89:e9:93
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
    media: Ethernet autoselect
    status: active
    flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=4219b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HNCSUM,TS04,WOL_MAGIC,VLAN_HWTSO>
    ether 18:03:73:c7:e7:65
    inet6 fe80::1a03:73ff:fe7e:765%em0 prefixlen 64 scopeid 0x2
    inet 134.134.134.4 netmask 0xffffffff broadcast 134.134.255.255
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
    media: Ethernet autoselect (10baseT/UTP <half-duplex>)
    status: active
    flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
    ipfw0: flags=8801<UP,SIMPLEX,MULTICAST> metric 0 mtu 65536
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
    lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
    options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>

```

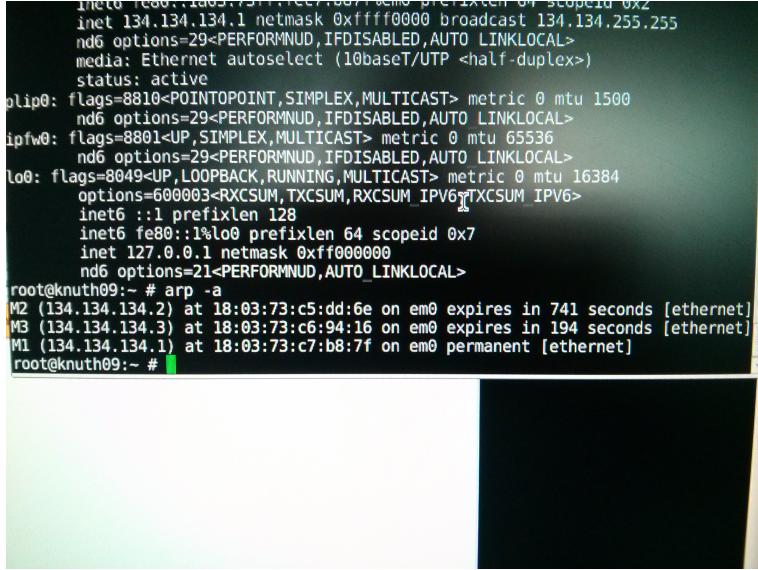
2.2 Observation de l'activité du réseau

Nous avons lancé Wireshark sur M2 (134.134.134.2) et exécuté une commande ping de M1 vers M3.

Time	Source IP	Destination IP	Type	Description
39.102.6021933000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=12/3072, ttl=64 (request in 37)
40.103.6025480000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=13/3328, ttl=64 (reply in 40)
41.103.6031880000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=13/3328, ttl=64 (request in 39)
42.103.6035460000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=14/3584, ttl=64 (reply in 42)
43.164.8824652000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=14/3584, ttl=64 (request in 41)
44.164.8838270000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=0/0, ttl=64 (reply in 44)
45.165.8838980000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=0/0, ttl=64 (request in 43)
46.185.8842498000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=1/256, ttl=64 (reply in 46)
47.166.8848979000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=1/256, ttl=64 (request in 45)
48.166.8852498000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=2/512, ttl=64 (reply in 48)
49.167.8858930000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=2/512, ttl=64 (request in 47)
50.168.8862450000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=3/768, ttl=64 (reply in 50)
51.168.8868660000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=3/768, ttl=64 (request in 49)
52.168.8872379000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=4/1024, ttl=64 (reply in 52)
53.169.8878330000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=4/1024, ttl=64 (request in 51)
54.169.8882320000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=5/1280, ttl=64 (reply in 54)
55.170.8888760000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=5/1280, ttl=64 (request in 53)
56.170.8892260000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=6/1536, ttl=64 (reply in 56)
57.171.8898770000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) reply id=0xa04 seq=7/1792, ttl=64 (reply in 58)
58.171.8902260000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) request id=0xa04 seq=7/1792, ttl=64 (request in 57)
59.180.4499180000	18:03:73:c7:b8:7f	Broadcast	ARP	60 Who has 134.134.134.3? Tell 134.134.134.1
60.180.4502130000	18:03:73:c6:94:16	18:03:73:c7:b8:7f	ARP	60 134.134.134.3 is at 18:03:73:c6:94:16
61.180.4505530000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=0/0, ttl=64 (reply in 62)
62.180.4509220000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=0/0, ttl=64 (request in 61)
63.181.4508260000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=1/256, ttl=64 (reply in 64)
64.181.4511780000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=1/256, ttl=64 (request in 63)
65.182.4518210000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=2/512, ttl=64 (reply in 66)
66.182.4521750000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=2/512, ttl=64 (request in 65)
67.183.452814000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=3/768, ttl=64 (reply in 68)
68.183.453178000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=3/768, ttl=64 (request in 67)
69.184.453813800	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=4/1024, ttl=64 (reply in 70)
70.184.4541650000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=4/1024, ttl=64 (request in 69)
71.185.454807800	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=5/1280, ttl=64 (reply in 72)
72.185.455164000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=5/1280, ttl=64 (request in 71)
73.186.455799000	134.134.134.1	134.134.134.3	ICMP	98 Echo (ping) request id=0xa04 seq=6/1536, ttl=64 (reply in 74)
74.186.456156000	134.134.134.3	134.134.134.1	ICMP	98 Echo (ping) reply id=0xa04 seq=6/1536, ttl=64 (request in 73)
0009 18:03:73:c6:94:16 18:03:73:c7:b8:7f 08:08:45:08:...:E...				
0010 09:54:09:85:00:00:40:01:61:13:86:86:86:01:86:86:T....@.a...				
0020 09:03:08:00:10:87:a4:00:00:56:56:0d:28:35:00:d0:.....:V.(..)				
0030 d4:19:08:09:08:00:0c:0e:0f:10:11:12:13:14:15:.....:%				
0040 16:17:18:19:1a:1b:1c:1d:1e:1f:20:21:22:23:24:25:.....:#%				
0050 em0: <live capture in progress> File: ... Packets: 74 - Displayed: 74 (100.0%)				Profile: Default

Observation de la commande ping

- Ping envoie une requête à la machine M3 (dans notre exemple), la source de la requête a l'adresse 134.134.134.1 et le destinataire a l'adresse 134.134.134.3 et la machine M3 envoie une réponse à M1 (destinataire vers la source). Ici les paquets envoyés sont visibles par toutes les machines (car M2 peut observer tous les paquets qui transitent par le hub) mais seulement le destinataire du ping (M3) répond (en envoyant un paquet).
- Quand la source du ping ne connaît pas l'adresse Ethernet du destinataire (d'adresse Internet connue 134.134.134.3), il envoie à tout le monde un paquet de type ARP afin de demander aux autres machines quelle est l'adresse Ethernet de 134.134.134.3 et la machine concernée répond à la source en lui donnant son adresse Ethernet via un paquet de type ARP.
- La table ARP est une table de correspondance entre une adresse Internet avec son adresse Ethernet des machines du réseau. Chaque entrée est conservée un certain temps (il y a une date d'expiration pour chaque entrée), ainsi les machines n'ont pas à chaque fois besoin de renvoyer des paquets de type ARP pour redemander les adresses Internet.



```

inet0: flags=110<NOFORWDIGIT,BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        inet 134.134.134.1 netmask 0xffffffff broadcast 134.134.255.255
                nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
                media: Ethernet autoselect (10baseT/UTP <half-duplex>)
                status: active
glip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
ipfw0: flags=8801<UP,SIMPLEX,MULTICAST> metric 0 mtu 65536
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=600003<RXCSUM,TXCSUM,RXCSUM IPV6 TXCSUM IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x7
        inet 127.0.0.1 netmask 0xff000000
                nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@knuth09:~ # arp -a
M2 (134.134.134.2) at 18:03:73:c5:dd:6e on em0 expires in 741 seconds [ethernet]
M3 (134.134.134.3) at 18:03:73:c6:94:16 on em0 expires in 194 seconds [ethernet]
M1 (134.134.134.1) at 18:03:73:c7:b8:7f on em0 permanent [ethernet]
root@knuth09:~ #

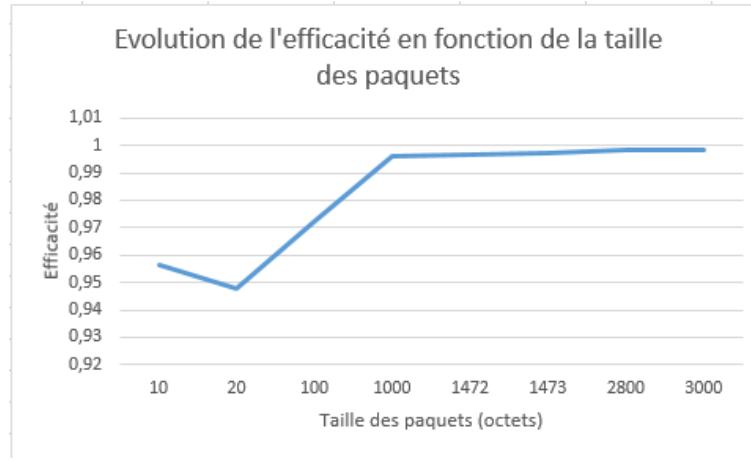
```

2.3 Observation du protocole CSMA/CD

- Nous avons exécuté la commande "udpmpt -s 2000 -p 55000 192.168.0.4" sur M2 (192.168.0.3) et au préalable nous avions exécuté "udptarget -p 55000" sur la machine d'adresse 192.168.0.4.
- On remarque qu'il n'y a aucune collision puisque qu'il n'y a qu'une seule machine qui émet vers une autre.
- En exécutant udpmpt sur une troisième à destination d'une quatrième machine, on remarque que des collisions apparaissent dans netstat.
- Le nombre de collisions augmente au fur et à mesure.
- Plus les paquets sont petits, plus on observe de collisions. Plus les paquets sont petits plus la carte réseau envoie un nombre important de paquets augmentant ainsi la probabilité de collisions.
 - Le protocole CSMA permet de détecter si le support est libre et ainsi d'émettre que si celui-ci est libre.
 - Le protocole CD permet de détecter une collision en écoutant le support pendant qu'une machine émet. S'il y a une différence entre ce qu'elle émet et ce que le support reçoit c'est qu'il y a eu une collision.
- Taille des paquets : 1472 o, Débit : 5617 kb/s

$$T_{emis} = \frac{\left(\frac{1472 * 8}{1024} \right)}{5617} \approx 0,00205s$$

$$T_{prop} = 1.5 / 1080000 = 1.4 * 10^{-6} s$$



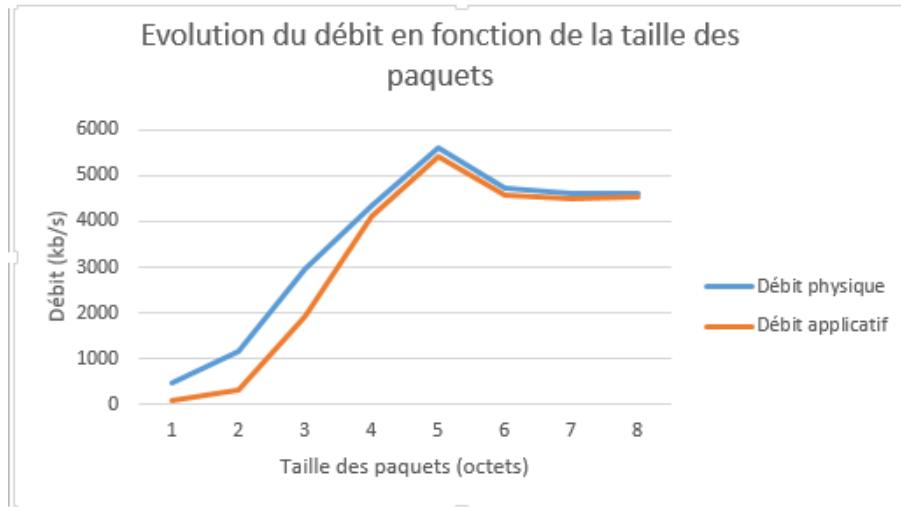
- On constate qu'à partir d'une certaine taille de paquets (ici à peu près 1000 octets) l'efficacité augmente peu et reste assez stable.

2.4 Analyse des performances du réseau

2.4.1 Mesure du débit applicatif

Taille des paquets (en octets)	10	20	100	1000	1472	1473	2800	3000
Débit physique (kb/s)	477	1145	2945	4339	5617	4733	4595	4604
Débit applicatif théorique <= à (kb/s)	74	309,5	1912,3	4116,7	5418,2	4565,6	4508,1	4522,6

(477 * $\frac{10}{64}$)



Plus la taille des données augmente, plus le débit applicatif approche du débit physique.

packets	(bge0)				output				colls
	input	errs	idrops	bytes	packets	errs	bytes	colls	
0	0	0	0	0	5502	0	6881940	718	49635.580
0	0	0	0	0	5390	0	5293800	742	49637.070
0	0	0	0	0	6661	0	6881940	728	49637.380
0	0	0	0	0	7147	0	7411329	777	49638.466
0	0	0	0	0	5643	0	5823188	769	49640.730
0	0	0	0	0	33924	0	1829672	331	49642.391
0	0	0	0	0	6422	0	5823238	762	49641.595
0	0	0	0	0	74580	0	3898440	781	49643.543
0	0	0	0	0	135798	0	7054326	155	49644.628
0	0	0	0	0	66963	0	3474120	824	49645.385
0	0	0	0	0	48215	0	2095088	785	49646.747
0	0	0	0	0	75655	0	3924966	801	49647.511
0	0	0	0	0	70976	0	3659766	829	49648.403
0	0	0	0	0	80473	0	4190166	747	49648.774
0	0	0	0	0	63713	0	3315000	785	49650.160
0	0	0	0	0	55235	0	2864166	777	49650.401
0	0	0	0	0	78865	0	4110600	802	49653.502
0	0	0	0	0	77339	0	4004520	759	49654.401
0	0	0	0	0	48215	0	2095088	785	49655.408
0	0	0	0	0	75655	0	3924966	801	49656.751
0	0	0	0	0	70976	0	3659766	829	49657.403
0	0	0	0	0	80473	0	4190166	747	49658.391
0	0	0	0	0	63713	0	3315000	785	49659.484
0	0	0	0	0	76678	0	3978800	771	49660.874
0	0	0	0	0	66241	0	3129366	811	49661.206
0	0	0	0	0	78983	0	3686238	749	49662.165
0	0	0	0	0	133242	0	6921728	93	49663.097
0	0	0	0	0	64362	0	3356756	797	49663.386
0	0	0	0	0	58928	0	3000000	755	49664.402
0	0	0	0	0	76983	0	4004520	758	49665.399
0	0	0	0	0	82297	0	4269730	791	49666.569
0	0	0	0	0	74585	0	3871920	813	49667.403
0	0	0	0	0	73178	0	3792360	795	49668.401
1	0	0	0	60	59832	0	3076320	796	49669.607
0	0	0	0	0	68446	0	3553680	775	49670.399
0	0	0	0	0	72996	0	3792350	799	49671.406
1	0	0	0	60	61244	0	4243200	754	49672.391
0	0	0	0	0	70965	0	3686238	777	49673.396
0	0	0	0	0	75402	0	3924966	771	49675.886
0	0	0	0	0	92825	0	4826649	751	49676.165
0	0	0	0	0	68268	0	3539224	795	49678.487
0	0	0	0	0	96219	0	5012880	789	49679.399
0	0	0	0	0	96351	0	5012880	789	49680.386
0	0	0	0	0	63514	0	3208920	783	49682.273
0	0	0	0	0	69341	0	3606720	749	49682.411
0	0	0	0	0	76217	0	3951480	788	49683.402
0	0	0	0	0	99133	0	5171400	771	49684.396
0	0	0	0	0	58852	0	3649800	791	49685.393

2.4.2 Mesure de latence

```

IPv4, packet size defaults to 1472
Packet size = 10 bytes, burst size = 10 bytes

Time      Packets   Total | Kbit/s Avg 10 Avg
52125.286 15911  15912 | 1271  1271  1271
52126.315 15300  31212 | 1190  1230  1230
52127.308 14790  46002 | 1190  1217  1217
52128.302 14790  60792 | 1190  1211  1211
52129.296 14790  75582 | 1190  1207  1207
52130.290 14790  90372 | 1190  1204  1204
^CNobufs = 5779596

----- udpmt statistics -----
102610 blocks of 10 bytes sent to 192.168.0.4 in 6.828 seconds.
Avg throughput = 1202 kbit/s
root@knuth10:~ # ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4): 56 data bytes
64 bytes from 192.168.0.4: icmp seq=0 ttl=64 time=0.685 ms
64 bytes from 192.168.0.4: icmp seq=1 ttl=64 time=0.356 ms
64 bytes from 192.168.0.4: icmp seq=2 ttl=64 time=0.510 ms
64 bytes from 192.168.0.4: icmp seq=3 ttl=64 time=0.356 ms
64 bytes from 192.168.0.4: icmp seq=4 ttl=64 time=0.508 ms
^C
--- 192.168.0.4 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.356/0.483/0.685/0.122 ms
root@knuth10:~ #

----- udpmt statistics -----
64 bytes from 192.168.0.4: ICMP seq=1 ttl=64 time=0.550 ms
64 bytes from 192.168.0.4: icmp seq=2 ttl=64 time=0.510 ms
64 bytes from 192.168.0.4: icmp seq=3 ttl=64 time=0.356 ms
64 bytes from 192.168.0.4: icmp seq=4 ttl=64 time=0.508 ms
^C
--- 192.168.0.4 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.356/0.483/0.685/0.122 ms
root@knuth10:~ # ping -s 100 192.168.0.4
PING 192.168.0.4 (192.168.0.4): 100 data bytes
108 bytes from 192.168.0.4: icmp seq=0 ttl=64 time=0.507 ms
108 bytes from 192.168.0.4: icmp seq=1 ttl=64 time=0.510 ms
108 bytes from 192.168.0.4: icmp seq=2 ttl=64 time=0.511 ms
108 bytes from 192.168.0.4: icmp seq=3 ttl=64 time=0.508 ms
108 bytes from 192.168.0.4: icmp seq=4 ttl=64 time=0.511 ms
108 bytes from 192.168.0.4: icmp seq=5 ttl=64 time=0.510 ms
108 bytes from 192.168.0.4: icmp seq=6 ttl=64 time=0.357 ms
108 bytes from 192.168.0.4: icmp seq=7 ttl=64 time=0.509 ms
^C
--- 192.168.0.4 ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.357/0.490/0.511/0.050 ms
root@knuth10:~ #

```