Déboguer les problèmes réseau : la taille compte

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 juin 2011

http://www.bortzmeyer.org/ping-taille-compte.html

Tout administrateur réseaux le sait : lorsqu'il y a un problème de connectivité IP, le premier outil à dégainer est ping. C'est en général un bon réflexe. Mais malheureusement beaucoup de ces administrateurs ne dépassent pas le stade de `ping nom-machine` et ne regardent jamais `lemanuel`. Ils se privent ainsi d'options intéressantes, comme celle qui permet de faire varier la **taille** des paquets de test. Ils ont tort car cette taille a une influence importante sur le test.

Prenons un exemple réel, sur un réseau CPL <http://www.bortzmeyer.org/cpl-maison.html> qui marchait mal (adaptateur CPL défaillant, a été changé par la suite). Le Web est peu utilisable (trop lent, certaines images ne chargent pas du tout). Pourtant, ping ne montre aucun problème :

```
% ping 208.75.84.80
PING 208.75.84.80 (208.75.84.80) 56(84) bytes of data.
64 bytes from 208.75.84.80: icmp_seq=1 ttl=46 time=132 ms
64 bytes from 208.75.84.80: icmp_seq=2 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=3 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=4 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=5 ttl=46 time=132 ms
64 bytes from 208.75.84.80: icmp_seq=6 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=7 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=8 ttl=46 time=131 ms
64 bytes from 208.75.84.80: icmp_seq=9 ttl=46 time=130 ms
64 bytes from 208.75.84.80: icmp_seq=10 ttl=46 time=131 ms
^C
--- 208.75.84.80 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9033ms
rtt min/avg/max/mdev = 130.773/131.526/132.131/0.521 ms
```

L'excellent logiciel mtr ne montre rien de plus. tcpdump montre peu de trafic (donc la ligne n'est pas surchargée) et de longues attentes (puis ça repart). Mais si on fait varier la taille des paquets avec l'option `-s` de ping, on voit bien le problème :

```
% ping -s 1450 208.75.84.80
PING 208.75.84.80 (208.75.84.80) 1450(1478) bytes of data.
1458 bytes from 208.75.84.80: icmp_seq=1 ttl=46 time=168 ms
1458 bytes from 208.75.84.80: icmp_seq=5 ttl=46 time=167 ms
1458 bytes from 208.75.84.80: icmp_seq=6 ttl=46 time=167 ms
1458 bytes from 208.75.84.80: icmp_seq=9 ttl=46 time=169 ms
1458 bytes from 208.75.84.80: icmp_seq=10 ttl=46 time=167 ms
1458 bytes from 208.75.84.80: icmp_seq=13 ttl=46 time=168 ms
1458 bytes from 208.75.84.80: icmp_seq=15 ttl=46 time=168 ms
1458 bytes from 208.75.84.80: icmp_seq=18 ttl=46 time=167 ms
^C
--- 208.75.84.80 ping statistics ---
19 packets transmitted, 8 received, 57% packet loss, time 18013ms
rtt min/avg/max/mdev = 167.407/168.034/169.066/0.639 ms
```

Avec des paquets de 1450 octets (la taille par défaut utilisée par ping est de 56 octets), le taux de perte, qui était nul, passe à plus de 50 %. On comprend pourquoi le Web avait des problèmes : si les requêtes HTTP sont souvent petites, les réponses, elles, ont en général la taille maximum permise par le réseau (1500 octets pour Ethernet). Si un problème frappe spécifiquement les gros paquets, un ping naïf marchera alors qu'une navigation Web échouera.

Mais pourquoi est-ce que les gros paquets auraient plus de mal à passer que les petits ? Une raison possible est un parasite aléatoire. Si le signal parasite frappe au hasard, les gros paquets auront une probabilité plus élevée d'être touchés (sur Ethernet, en raison de la somme de contrôle des paquets, si un seul bit est modifié, le paquet entier est jeté). Ce genre de choses arrive rarement sur les réseaux filaires mais est bien plus fréquent en WiFi.

Autre exemple de comportement différent selon la taille, les problèmes du point d'échange Sfinx en juin 2011 (ticket Sfinx n° 2215418). Un mtr ordinaire montre un certain pourcentage de pertes, dont la cause est inconnue :

```
% mtr --report --report-cycles 100 -4 f.root-servers.net
                                Loss%   Snt   Last   Avg  Best  Wrst StDev
  ...
  3. vl387-te2-6-paris1-rtr-021.n  0.0%   100    1.4   2.2   1.4  33.1   4.4
  4. te0-1-0-3-paris1-rtr-001.noc  0.0%   100   94.2  18.5   1.5  98.7  24.5
  5. isc-f-root-server-2.sfinx.tm  7.0%   100    1.9   4.7   1.8 179.3  20.6
  6. f.root-servers.net            6.0%   100    1.9   2.7   1.8  10.0   1.8
```

En augmentant la taille des paquets grâce à l'option `--psize`, le taux de pertes augmente nettement :

```
% mtr --report --report-cycles 100 --psize 1000 -4 f.root-servers.net
                                Loss%   Snt   Last   Avg  Best  Wrst StDev
...
  3. vl387-te2-6-paris1-rtr-021.n  0.0%   100    1.6   1.7   1.5  19.6   1.8
  4. te0-1-0-3-paris1-rtr-001.noc  0.0%   100    2.0  19.3   1.7 130.3  24.7
  5. isc-f-root-server-2.sfinx.tm 17.0%   100    2.1   5.3   2.1 179.4  21.2
  6. f.root-servers.net           17.0%   100    2.7   3.5   2.6  10.5   1.9
```

Cela indique clairement un problème situé dans les couches 1 ou 2, par exemple une mauvaise adaptation Ethernet ("*half-duplex*" au lieu de "*full-duplex*"). Merci à Johan Remy pour son aide sur ce point.

Autre cas où faire varier la taille est utile, l'étude de problèmes de performances : dans certains cas, c'est le nombre de paquets par seconde qui est le facteur limitant de la performance du réseau, dans

d'autres (réseaux lents, par exemple vieux modems), c'est le nombre de bits par seconde et, dans ce cas, le test avec des gros paquets donnera une vision plus réaliste des performances réelles.

Enfin, faire varier la taille des paquets de tests est nécessaire lorsqu'on veut déboguer des problèmes de MTU <http://www.bortzmeyer.org/mtu-et-mss-sont-dans-un-reseau.html>. En raison de la fréquence des tunnels et d'une incompréhension du rôle d'ICMP par les gérants de pare-feux, c'est le plus souvent IPv6 qui est affecté par ce genre de problèmes. Voici un exemple. Tout se passe bien ?

```
% ping6 -n e.ext.nic.fr
PING e.ext.nic.fr(2a00:d78:0:102:193:176:144:6) 56 data bytes
64 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=1 ttl=59 time=22.3 ms
64 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=2 ttl=59 time=22.7 ms
64 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=3 ttl=59 time=22.2 ms
64 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=4 ttl=59 time=22.3 ms
64 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=5 ttl=59 time=22.3 ms
^C
--- e.ext.nic.fr ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 22.221/22.407/22.772/0.211 ms
```

Hélas non. Si on augmente la taille des paquets, cela se passe bien jusqu'à une certain valeur :

```
% ping6 -n -c 5 -s 1400 e.ext.nic.fr
PING e.ext.nic.fr(2a00:d78:0:102:193:176:144:6) 1400 data bytes
1408 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=1 ttl=59 time=24.8 ms
1408 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=2 ttl=59 time=24.9 ms
1408 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=3 ttl=59 time=24.8 ms
1408 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=4 ttl=59 time=27.0 ms
1408 bytes from 2a00:d78:0:102:193:176:144:6: icmp_seq=5 ttl=59 time=24.9 ms

--- e.ext.nic.fr ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 24.867/25.339/27.024/0.861 ms
```

Mais plus après :

```
% ping6 -n -c 5 -s 1470 e.ext.nic.fr
PING e.ext.nic.fr(2a00:d78:0:102:193:176:144:6) 1470 data bytes

--- e.ext.nic.fr ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4018ms
```

C'est caractéristique des problèmes de MTU. Lorsque la liaison réseau est correcte (pas de filtrage d'ICMP nulle part), la réduction de la MTU (ici, la machine cible est derrière un tunnel) ne gêne pas :

```
% ping6 -s 3000 -c 5 2001:470:1f11:3aa::1
PING 2001:470:1f11:3aa::1(2001:470:1f11:3aa::1) 3000 data bytes
From 2001:470:0:6e::2 icmp_seq=1 Packet too big: mtu=1480
3008 bytes from 2001:470:1f11:3aa::1: icmp_seq=1 ttl=56 time=102 ms
3008 bytes from 2001:470:1f11:3aa::1: icmp_seq=2 ttl=56 time=103 ms
3008 bytes from 2001:470:1f11:3aa::1: icmp_seq=3 ttl=56 time=103 ms
3008 bytes from 2001:470:1f11:3aa::1: icmp_seq=4 ttl=56 time=103 ms
3008 bytes from 2001:470:1f11:3aa::1: icmp_seq=5 ttl=56 time=102 ms

--- 2001:470:1f11:3aa::1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 102.757/102.968/103.097/0.138 ms
```

Notez le « "*Packet too big*" » qui montre que le paquet ICMP a été bien reçu.

Un point important pour tous ces tests : le routage dans l'Internet n'est pas forcément symétrique. Rien ne dit que les paquets de réponse suivront le même trajet que les paquets de requête. Cela peut rendre le déboguage très délicat : par exemple, une session HTTP où les requêtes (petites) empruntent un chemin et les réponses (grandes) un autre, sera parfois difficile à déboguer avec un `ping -s` où les paquets auront la même taille à l'aller et au retour (merci à Thomas Mangin pour la précision).

Autre point, question outils. Comme certains réseaux (ou systèmes) traitent différemment les protocoles, le test avec ICMP, que fait ping, peut ne pas être représentatif. Un outil comme hping permet le même genre de manipulations avec les autres protocoles par exemple `hping --syn -p 80 --data 1200 example.com` va tester en TCP avec des paquets de 1200 octets. hping dispose de nombreuses autres options (merci à Florian Crouzat pour la suggestion).

Bref, pour résumer, lors des entretiens d'embauche d'un administrateur réseaux, pensez à l'interroger sur le rôle de l'option `-s`. C'est un bon test !

---

http://www.bortzmeyer.org/ping-taille-compte.html