Quelles machines pinguer pour vérifier sa connectivité Internet ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 septembre 2012. Dernière mise à jour le 7 septembre 2012

https://www.bortzmeyer.org/que-pinguer.html

Lorsqu'on met en place une infrastructure de surveillance de serveurs Internet, il est agaçant de recevoir autant d'alarmes que de serveurs surveillés, alors que c'était simplement la connectivité Internet de la sonde de surveillance qui était en panne. Tous les grands logiciels de surveillance de réseau ont une option pour éviter cela, qui permet de dire que le test d'un serveur dépend du succès d'un test sur un autre composant, par exemple le routeur de sortie (si ce dernier est en panne, il n'y a pas besoin de tester les serveurs et d'alerter si ça rate). Mais quel composant tester pour déterminer qu'on a un accès Internet qui marche ?

Avec les logiciels de la famille Nagios comme Icinga, l'option qui permet d'indiquer la dépendance d'une machine envers une autre se nomme `parents` . Si on écrit :

```
define host {
        host_name               freebox
        address                 192.168.1.1
}

define host {
        host_name    remote-host
        parents freebox
...
}
```

alors on déclare que la connectivité de `remote-host` dépend de celle de `freebox`. Si on est connecté à l'Internet par ce seul routeur, c'est logique. Si `freebox` est injoignable, le reste de l'Internet l'est aussi.

Mais si `freebox` fonctionne mais ne route plus <http://bugs.freeplayer.org/task/10258> ? Ou bien si quelque chose déconne dans le réseau de Free bloquant tout accès à l'extérieur ? C'est là qu'il est utile de tester autre chose que le premier routeur. On voudrait en fait tester si on a toujours un accès Internet et pouvoir écrire :

```
define host {
        host_name    remote-host
        parents Internet
...
}
```

(Les experts Nagios/Icinga auront noté qu'il faudrait plutôt utiliser `hostdependency` ou `servicedependency`, pour ne même pas effectuer les tests en cas de panne de la connectivité. Mais c'est un détail ici.) Mais comment faire ce test ? En pinguant des machines distantes, bien évidemment. Il « suffit » de sélectionner des machines stables, qui répondent aux paquets ICMP d'écho, et qui n'ont elles-mêmes que très peu de pannes. Mais lesquelles choisir ?

Il faut bien voir que ces machines sur l'Internet, ces amers ou mires <https://www.bortzmeyer.org/amer-mire.html> ("*target*" en anglais) ne nous appartiennent pas. Si chaque petit réseau local, pour tester sa connectivité, pingue `192.0.2.1` toutes les trois minutes, et qu'il y a dix millions de petits réseaux qui le font dans le monde, `192.0.2.1` recevra 50 000 paquets/seconde, ce qui est une charge sérieuse même pour un gros serveur. (En débit, cela ne fera pas grand'chose car les paquets de test seront de petite taille. Mais pour les routeurs et les serveurs, ce n'est pas que le nombre de bits par seconde qui compte, celui des paquets par seconde est également important, chaque paquet nécessitant un traitement, indépendemment de sa taille.) Utiliser le premier serveur qu'on a choisi pour des tests répétitifs, c'est comme jeter une canette vide par terre : si une seule personne le fait, c'est juste un peu agaçant, si tout le monde le fait, on détruit l'environnement. Ce n'est pas parce qu'un service est publiquement accessible qu'on peut le bombarder de requêtes pour son usage personnel. Dans le passé, il est déjà arrivé qu'un constructeur configure (bêtement) ses machines pour utiliser un serveur public sans autorisation, l'écroulant ainsi sous la charge <http://slashdot.org/story/06/04/07/130209/d-link-firmware-abuses-open-ntp-servers>.

Qu'est-ce que cela nous laisse comme possibilités ? En posant la question, on obtient des réponses (plus ou moins dans l'ordre décroissant du nombre de suggestions) :

— `8.8.8.8` (ou, à la rigueur, `8.8.4.4`, qui est sans doute moins sollicité), à savoir Google Public DNS <https://www.bortzmeyer.org/google-dns.html>. Un service très fiable, qui a très peu de chances de tomber en panne. Mais peut-on l'utiliser ainsi, de manière automatique, dans des tests répétés ? Cela ne figure certainement pas dans les conditions d'utilisation de ce service, et Google pourrait donc décider de couper les réponses ICMP écho du jour au lendemain (ou de mettre en place une limitation de débit).
— `www.facebook.com` (ou `www.google.com`) avec l'argument « Si Facebook est en panne, de toute façon, tout l'Internet est fichu ». Cela pose un peu les mêmes problèmes que précédemment.
— Pinguer un des serveurs DNS de la racine. Ils marchent en permanence (c'est sans doute un des composants les plus robustes de l'Internet), répondent à l'ICMP écho et, n'étant pas gérés dans un but lucratif, on n'est pas à la merci d'un changement soudain de politique commerciale. Mais ces serveurs ne sont pas prévus pour un tel test automatisé. Ils y résisteront, bien sûr, mais est-ce un usage légitime ? Je ne pense pas et les opérateurs des serveurs racine, interrogés, sont également de cet avis. Il faut aussi se rappeler qu'il s'agit d'un service critique : toute perturbation est à éviter.
— Pinguer un des serveurs de l'AS112 (cf. RFC 7534 [1]). Après tout, ils sont **censés** recevoir du n'importe quoi tout le temps. Un peu de ping ne les dérangera pas. Par contre, ils ne sont pas toujours fiables et, par la magie de l'"*anycast*", ils peuvent être plus près que vous ne croyez, ce qui en fait de mauvaises mires pour tester le grand Internet.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc7534.txt`

— Utiliser un ensemble de sites bien connus et qui sont probablement assez stables. Si chacun se fait sa propre liste, cela diminuera la charge sur chacun de ces sites. Mais cela reste un usage non autorisé.
— Pinguer des équipements du FAI par exemple son serveur Web ou bien les routeurs sur le trajet (après un traceroute pour les repérer). Notez que cela ne détecte pas le cas où la(es) liaison(s) du FAI avec l'Internet sont en panne. Mais la plupart des pannes (comme celle de la Freebox citée plus haut) concernent « le dernier kilomètre » donc c'est peut-être supportable. L'avantage est que tout le monde ne teste pas le même service (les abonnés d'un FAI sont les seuls à tester leur FAI) et que c'est un service qu'on paie. En tirant un peu sur la ficelle, on peut considérer que l'abonnement inclut le droit de pinguer `www.mon-FAI.net` chaque minute. Les FAI ne sont pas forcément d'accord avec cette analyse, et peuvent faire remarquer que les routeurs sont là pour router, pas pour répondre aux pings. Le mieux serait que les FAI fournissent à leurs abonnés une mire à tester, comme le fait OVH avec sa machine `ping.ovh.net` (un service qui ne semble pas vraiment documenté) ou Free avec `test-debit.free.fr` (pas plus documenté).
— Utiliser une mire publique, prévue pour cela et qu'on a l'autorisation d'utiliser. (Certaines entreprises fournissent peut-être un service de mires privées payantes mais je n'en connais pas. Il existe aussi des mires spécifiques à un système d'exploitation particulier comme `www.msftncsi.com` qu'utilise Windows.) Je ne connais actuellement que trois mires publiques de ce genre, gérées par Team Cymru <`http://www.cymru.com/`> (la machine `fap.cymru.com`, en HTTP, elle donne plein de renseignements rigolos), par Rezopole <`http://www.rezopole.net`> (`ping.rezopole.net`) et par Demarcq <`http://www.demarcq.net/`> (`ping.demarcq.net`). Elles sont peu documentées, peut-être par manque de temps et peut-être pour éviter d'attirer plein de trafic. Faudrait-il créer des services associatifs pour cet usage, comme pool.ntp.org <`http://www.pool.ntp.org/`> le fait pour NTP ? Ou procéder à des échanges avec d'autres acteurs de l'Internet, comme on le fait couramment pour les serveurs DNS secondaires ?

Bref, le débat n'est pas simple. On peut encore le compliquer avec des questions comme « Vaut-il mieux utiliser les paquets ICMP echo de ping ou bien une application comme HTTP ou DNS ? ».

Merci à WBrown pour ses bonnes suggestions sur le pinguage du FAI, à John Kristoff pour m'avoir rappelé le travail que fait Team Cymru, à Jean-Philippe Camguilhem pour les explications sur les dépendances dans Icinga et à Fabien Vincent pour m'avoir appris l'existence de `ping.ovh.net`. À noter qu'il n'existe pas en France de « service public » de mires accessibles pour ce genre de tests mais que le RIPE-NCC a un service similaire, les Ancres <`https://atlas.ripe.net/about/anchors/`>. L'une des ancres a un nom simple à retenir, `ping.ripe.net`.

J'ai aussi fait un article <`https://www.bortzmeyer.org/icinga.html`> qui décrit en détail la configuration d'Icinga pour ces mires publiques et l'usage que j'en fait.