

RFC 7228 : Terminology for Constrained Node Networks

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 mai 2014

Date de publication du RFC : Mai 2014

<https://www.bortzmeyer.org/7228.html>

Il existe désormais de nombreux protocoles et donc de nombreux documents (par exemple des RFC) pour les objets limités (en puissance de calcul, en énergie, en capacité réseau...). L'Internet, conçu d'abord pour de gros ordinateurs alimentés en permanence, s'adapte de plus en plus à ces objets limités. D'où l'importance d'une terminologie standard : ce nouveau RFC définit un certain nombre de termes utilisés dans la normalisation de protocoles pour ces petits objets.

Des exemples de tels objets ? Des capteurs mis dans la nature, un actionneur dans une usine, des Arduino au fond d'un placard, et bien d'autres encore. Ils ont en commun de manquer sérieusement d'une ou plusieurs ressources, qui étaient considérées comme virtuellement illimitées : énergie électrique, capacité réseau <<https://www.bortzmeyer.org/capacite.html>>, capacité de calcul... Tous ensemble, ces objets contribuent à ce qu'on appelle (un terme médiatique sans définition rigoureuse) « l'Internet des Objets ».

Adapter les protocoles Internet à cet Internet des Objets nécessite deux adaptations quantitatives :

- « *scaling up* », s'adapter à un réseau comptant **beaucoup** plus de machines, peut-être des dizaines de milliards <<http://www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf>>. Le modèle dominant, aux débuts de l'Internet, était d'un ordinateur par entreprise. Il est passé à un par département (l'« informatique départementale » des années 70 et 80) puis à un par personne avec la micro-informatique. Aujourd'hui, le modèle est plutôt de plusieurs ordinateurs par personne, même si on ne pense pas forcément à ces objets comme étant des ordinateurs.
- « *scaling down* », s'adapter à des machines ayant des capacités considérées aujourd'hui comme très faibles (tout en étant parfois supérieures à celles des machines des débuts de l'Arpanet).

La section 2 de notre RFC définit la terminologie utilisée. Le RFC est en anglais et les traductions sont de moi, donc n'ont pas de caractère normatif, contrairement au texte anglais. Premier terme défini, celui de « **nœud contraint** » ("*constrained node*"). Ce sont nos objets. Ils sont définis négativement, par le fait qu'il n'ont **pas** certaines caractéristiques que l'on associe d'habitude automatiquement à un ordinateur moderne. Par exemple, certains objets n'ont pas d'alimentation électrique permanente et doivent donc économiser la batterie. Certains objets n'ont pas de MMU et ne peuvent donc pas gérer de mémoire virtuelle, ce qui implique de faire très attention à la consommation mémoire des programmes. Certains objets n'ont aucune interface utilisateur (pas d'écran, même limité), ce qui rend impossible certaines opérations (« pour sécuriser l'Internet des Objets, on va utiliser des mots de passe, qu'il faudra rentrer dans chaque objet »). Les objets ont aussi des contraintes quantitatives, comme une ROM très limitée, qui impose un code de petite taille. Un nœud contraint est donc très différent d'un PC de bureau (où le programmeur de Gnome ou de Windows ne réfléchit, ni à la consommation électrique, ni à la consommation de mémoire...) mais même également d'un "*smartphone*", engin finalement très perfectionné et qui peut faire tourner un Unix peu modifié. On voit avec cet exemple que la définition de nœud contraint n'est pas complètement rigoureuse (après tout, le "*smartphone*" est très contraint en énergie électrique).

La section 3 du RFC tente de quantifier cette notion de nœud contraint, en introduisant des classes de nœuds, selon leurs caractéristiques. (Petit rappel : un kibioctet, noté Kio, c'est 1 024 octets.) La classe 0 (aussi notée C0), la plus contrainte, a une mémoire vive bien en dessous de 10 Kio, et une mémoire morte très inférieure à 100 Kio. La classe 1 a des mémoires aux alentours, respectivement, de 10 et 100 Kio. Et la classe 2, la plus riche, a dans les 50 Kio de RAM et 250 de ROM. Ne comptez pas sur la loi de Moore pour tirer ces chiffres vers le haut dans le futur : dans le domaines des objets, les progrès sont utilisés pour baisser les prix et/ou la consommation électrique, et équiper davantage d'objets, pas pour augmenter les capacités. Pour comparer, un Raspberry Pi <<https://www.bortzmeyer.org/raspberry-pi.html>> typique a 512 Mio (oui, je sais, c'est souvent écrit Mo mais c'est une erreur, il faut dire que la situation est compliquée). Il n'est donc pas considéré comme objet contraint. En revanche, un Arduino typique peut n'avoir que 2 à 8 Kio, ce qui les met dans la classe 0.

Les nœuds de la classe 0 ne peuvent pas espérer avoir de vraie communication TCP/IP. Le terme d'« Internet des Objets » est tout à fait impropre pour eux, ils ne peuvent pas se connecter à l'Internet, et surtout pas de manière sécurisée. Typiquement, leur connexion aux réseaux se fera via une passerelle. La classe 1 a plus de chance : ils n'arriveront certes pas à faire tourner HTTP + TLS + XML + XSLT mais ils peuvent utiliser des protocoles TCP/IP spécialement prévus pour eux, comme CoAP. Les nœuds de la classe 2, eux, peuvent faire tourner plus ou moins les mêmes protocoles qu'un serveur ordinaire ou qu'une machine de bureau. Bien sûr, ils restent contraints et toute optimisation est donc bonne à prendre, par exemple pour économiser la batterie. Il n'y a pas de classe 3 ou au delà : les machines ayant de meilleures capacités que la classe 2 n'ont pas de contraintes comparables et il n'y a donc pas besoin que l'IETF développe des normes ou pratiques particulières pour eux.

Deuxième terme important dans la section 2 de notre RFC, « **réseau contraint** ». Un réseau contraint ("*constrained network*", cf. RFC 7102¹) est un réseau auquel manquent certaines caractéristiques qu'on a l'habitude de tenir pour acquises sur l'Internet : le débit maximum est fortement limité, le taux de pertes de paquets très élevé, les liens sont souvent très asymétriques (ça marche dans un sens mais pas dans l'autre)... Ces contraintes peuvent venir du coût (on a serré les budgets), du milieu ambiant (pensez à un réseau sous-marin...), des contraintes légales (limitation de la puissance maximale), ou des contraintes techniques (vieilles technologies dont on ne peut pas se débarrasser).

Il ne faut pas confondre le réseau contraint avec le « **réseau de nœuds contraints** » ("*constrained node network*"). Ce dernier est un réseau composé en grande partie de nœuds contraints, ce qui réduit ses possibilités, même si le réseau lui-même est parfait.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7102.txt>

Plus grave que le réseau contraint, le « **réseau limité** » (*"challenged network"*). C'est un réseau qui a du mal à fournir des services IP de bout en bout : pas de connectivité complète, interruptions fréquentes (voir RFC 4838 pour des exemples rigolos), délais d'acheminement des paquets qui sont supérieurs au MSL (*"Maximum Segment Lifetime"*) de TCP, empêchant ainsi d'établir des connexions TCP, même si ping fonctionne (les réseaux RFC 1149 sont un bon exemple)...

Enfin, la section 2 rappelle deux termes fréquemment utilisés dans les RFC et déjà définis. Un LLN (*"Low-power Lossy Network"*), défini dans le RFC 7102, est un réseau contraint et formé de nœuds contraints. Cela oblige à des protocoles spécifiques comme RPL (RFC 6550). Et un LowPAN (*"Low-Power Wireless Personal Area Network"*, RFC 4919) était le nom d'un groupe de travail IETF et le terme est souvent utilisé comme synonyme de LLN (le *"Personal"* dans LowPAN n'ayant plus guère de signification aujourd'hui). Un dérivé, 6LowPAN, désigne le travail fait pour utiliser IPv6 sur ces réseaux LowPAN.

L'une des contraintes les plus fréquentes et les plus pénibles est évidemment la consommation électrique. La section 4 de notre RFC définit des termes pour ce domaine. Deux termes sont importants pour caractériser la consommation d'un nœud contraint : **Ps** (pour *"Power sustainable"*) indique la consommation instantanée de la machine en train de tourner (en watts) et **Et** est l'énergie totale qui sera consommée avant que la machine ne s'arrête, sa batterie à plat. « Ps » doit être indiqué pour chaque mode du nœud contraint si, comme c'est souvent le cas, il existe plusieurs modes de fonctionnement, consommant plus ou moins de courant. « Et » est infini pour une machine connectée au courant général, et se mesure en joules.

Cela permet de définir des classes selon des critères liés à l'énergie électrique disponible : E0 est la classe des machines dont la disponibilité en électricité est liée à des événements extérieurs (recharge par une action mécanique externe, par exemple), E1 celle des machines dont la consommation électrique est limitée durant une certaine période (la durée entre deux recharges, ou bien le cycle quotidien du soleil si la recharge se fait par énergie solaire), E2, celle des machines qui ont une durée de vie limitée (une batterie non remplaçable, par exemple, ce qui fait que bien des machines sont à la fois E1 et E2), et E9 la classe des machines illimitées, qui peuvent puiser du courant à volonté.

Du fait de ces limites d'énergie disponibles, les nœuds contraints doivent s'adapter. Souvent, la radio est le principal poste de consommation : il est bien plus coûteux d'envoyer un paquet que de le fabriquer, même si on a utilisé le chiffrement. Les machines peuvent être classées selon leur stratégie de consommation. P0 est la classe des machines qui sont éteintes la plupart du temps et ne sont allumées que pour une tâche spécifique. Cela veut dire que leur attachement au réseau doit recommencer à chaque allumage, une tâche coûteuse en énergie (émission d'une requête DHCP, attente...) et qu'il est intéressant d'optimiser. P1 est une classe de machines qui sont sous tension en permanence, mais dans un mode à consommation réduite, où les réactions ne sont pas forcément très rapides. Elles restent donc attachées au réseau en permanence. Enfin, P9 regroupe les machines qui sont tout le temps complètement réveillées, connectées au réseau en consommation maximale. Ce serait le cas d'un routeur, par exemple. (Le RFC 7102 ne distinguait que deux niveaux, correspondant à nos classes P0 et P9.)