RFC 6574 : Report from the Smart Object Workshop

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 avril 2012

Date de publication du RFC : Avril 2012

http://www.bortzmeyer.org/6574.html

Traditionnellement, les engins connectés à l'Internet étaient des ordinateurs généralistes, dotés de bonnes capacités de calcul et alimentés en courant électrique à volonté. Est-ce que la famille de protocoles TCP/IP convient toujours, lorsque les machines connectées sont des petits trucs avec un processeur ridicule, et alimentés par une batterie qu'il faut faire durer le plus longtemps possible ? C'était l'objet de l'atelier "*Interconnecting Smart Objects with the Internet*" qui s'est tenu à Prague le 25 mars 2011, à l'instigation de l'IAB. Le but était d'explorer cette question, de voir si la connexion à l'Internet de ces « objets malins » était triviale, possible avec quelques adaptations des protocoles ou, pourquoi pas, irréaliste sans des changements radicaux.

L'atelier a rassemblé près de cent participants (liste complète en annexe D), la plupart ayant déjà travaillé sur l'adaptation de TCP/IP à des objets **contraints** (en mémoire, en alimentation, etc). Ces objets peuvent être des capteurs, des actionneurs, etc. À côté d'eux, même un "*smartphone*" est riche en capacités CPU et en énergie électrique.

Rappelons que l'IAB, qui a produit ce RFC, est en charge du travail d'**architecture** à l'IETF. Ce comité ne s'occupe pas des protocoles individuels (dont le développement, dans les groupes de travail IETF, est piloté par l'IESG) mais du maintien et du développement d'une architecture saine pour l'Internet, avec une insistance sur le travail à long terme.

Une de ces questions à long terme est celle du passage d'un Internet composé uniquement d'ordinateurs à un Internet auquel sont attachés, entre autres, des engins qu'on qualifie du terme générique d'« objets » et qui ont en commun de sévères restrictions matérielles, rendant parfois difficile un attachement normal à l'Internet. Pour les développements liés à ces objets, les gens du marketing ont inventé le terme ridicule d'« Internet des objets » (IoT en anglais pour « "*Internet of Things*" »). Mais, même si le terme est grotesque, il y a une vraie question technique : l'Internet est-il accessible à des objets contraints en ressources ?

Connecter ces objets présente souvent des avantages (pour des capteurs, cela permet de les interroger à distance, alors qu'auparavant il fallait qu'un humain se déplace pour les lire), et certains le sont déjà, en utilisant des protocoles privés. Mais la tendance actuelle est de migrer vers les protocoles standard de l'Internet. Cela se reflète entre autres dans les groupes de travail IETF, dont plusieurs planchent sur cet Internet des Trucs : "*Constrained RESTful Environments (CoRE)*" <http://tools.ietf.org/wg/core>, "*IPv6 over Low power WPAN (6LowPAN)*" <http://tools.ietf.org/wg/6lowpan>, "*Routing Over Low power and Lossy networks (ROLL)*" <http://tools.ietf.org/wg/roll> ou "*LightWeight Implementation Guidance (LWIG)*" <http://tools.ietf.org/wg/lwig>.

C'est que le travail d'ingéniérie n'est pas facile : il faut jongler entre des exigences contradictoires, la sécurité, le prix, l'utilisabilité, la protection de la vie privée (ces objets sont souvent liés à notre vie quotidienne), la longévité sous batterie, etc. (Le RFC cite, sur cette questions des exigences contradictoires, l'excellent article « "*Tussle in Cyberspace : Defining Tomorrow's Internet*" <http://www.bortzmeyer.org/tussle-cyberspace.html> ».)

L'atelier de l'IAB partait de la constatation que les protocoles actuels de l'Internet, d'IPv4 à IPv6, d'UDP à TCP et jusqu'à bien sûr HTTP, fonctionnent même sur les objets contraints. Il en existe déjà dans la nature. Une première question était « quel bilan tirer de ces premiers déploiements ? ». Une autre était « quels sont les problèmes pas encore résolus ? ».

En quoi ces objets sont-ils contraints ? Que leur manque-t-il exactement, par rapport à un ordinateur classique ? La section 2 détaille leurs propriétés :

— Très limités en alimentation électrique, ces engins doivent se mettre en sommeil fréquemment. Cela fait un sérieux changement de paradigme pour les protocoles Internet, habitués aux ordinateurs toujours en marche.
— Très limités en capacité réseau, ces objets ont souvent moins de 100 kb/s disponibles et parfois seulement 10 ou 20 ! En outre, leur connectivité est fréquemment affectée par des taux de pertes de paquets importants.
— Très limités en mémoire, les appareils en question ne peuvent pas stocker du code ou des données compliquées. Par exemple, la carte Arduino n'a que 2 ko de RAM et 32 ko de mémoire flash.
— Liée au problème de la consommation électrique, il y a aussi la faible capacité des processeurs. Pour moins consommer, ils sont souvent lents.
— Peu ou pas d'interface utilisateur (encore moins qu'un routeur), et donc difficiles à configurer.
— Enfin, il y a des limitations de taille physique. Si la traditionnelle carte ATX fait 305x244 mm, la CoreExpress, conçue pour l'embarqué, ne fait que 58x65 mm.

La loi de Moore, souvent citée pour relativiser ces limites, ne suffit pas. Les gains qu'elle permet sont souvent utilisés pour réduire coûts et consommation électrique, pas pour augmenter la puissance.

Place à l'atelier lui-même, maintenant. Avec 70 papiers acceptés (la liste complète figure en annexe B du RFC), il n'est pas facile à synthétiser. La section 3 de notre RFC essaie quand même d'indiquer les grands lignes de la réflexion. Quatre grandes questions structuraient l'atelier, les questions d'architecture, les points soulevés par les nœuds dormants, la sécurité et le routage. Commençons par l'architecture, section 3.1.

Première question d'architecture, parle-t-on de connecter ces objets contraints à l'Internet (singulier, et avec une majuscule, car c'est une entité unique comme l'Atlantique ou l'Himalaya) ou bien à des réseaux utilisant TCP/IP, mais qui ne sont pas forcément reliés à l'Internet ? Après tout, les capteurs et actionneurs dans une usine n'ont pas forcément besoin de se connecter à YouTube et, même si le RFC ne le rappelle pas, du point de vue sécurité, il est irresponsable de relier un SCADA à l'Internet. (Le RFC cite un cas moins grave, celui des fontaines devant l'hôtel Bellagio, à Las Vegas, fontaines qui sont électroniquement contrôlées à distance mais qui n'ont pas besoin d'être sur le même réseau que les clients dans leurs chambres, cf. l'exposé de B. Dolin à l'atelier.) Même chose pour les "*smart grids*"

qui n'ont rien à gagner (et beaucoup à perdre) à être joignables depuis l'Internet. Bref, l'IETF doit-elle séparer clairement le cas des objets connectés à l'Internet et celui des objets reliés à TCP/IP ? Du point de vue purement économique, il est clair qu'il vaut mieux un seul réseau, comme l'a rappelé Cullen Jennings, qui prédisait que tour le monde serait connecté au même Internet, car c'était plus simple et moins coûteux.

Une autre question d'architecture fondamentale est celle des protocoles spécifiques à un domaine d'application. Prenons l'exemple d'une ampoule électrique du futur. Grâce aux normes, on la branche au réseau et elle acquiert une adresse IP (grâce à DHCP ou NDP), trouve un serveur DNS, on peut la pinguer. On peut même imaginer qu'elle ait un serveur HTTP et tout client HTTP peut alors s'y connecter. Mais pour une tâche simple pour laquelle elle est conçue, éclairer, il n'y a pas de norme. Une commande de base « allume-toi » n'est pas normalisée. Comment s'assurer que l'objet, non seulement sera connecté (ping), mais pourra interagir avec son environnement, selon ses capacités ? Les protocoles de couche 3, 4 et même 7 ne suffisent pas. Il va falloir mettre au point des modèles (des classes, dirait-on en programmation objet, dont le nom est bien adapté ici), pour exploiter ces objets « intelligents ».

Un des principes de base de l'Internet, qui est défié par l'arrivée massive des objets connectés, est celui comme quoi l'intelligence est uniquement aux extrémités et le réseau fournit uniquement un service de base (faire passer les paquets). Avec des objets assez limités en ressources, on pourrait imaginer de revisiter ce principe et de mettre de l'« intelligence » dans le réseau. Parmi les services pour lesquels ce serait un bon endroit, on peut imaginer :

— La localisation, à savoir indiquer à l'objet où il se trouve. Il ne s'agit pas uniquement des coordonnées GPS mais aussi de l'environnement (« tu es dans un hôpital », « tu es dehors »).
— Routage fondé sur les noms, où les couches basses sauraient utiliser directement le nom d'une ressource (par exemple « Lampe »), sans passer par des résolutions en identificateurs de plus bas niveau. C'est relativement simple à faire pour un petit réseau local, où un contrôleur peut diffuser à tous « Lampe, allume-toi », et où la lampe se reconnait et s'exécute, mais bien plus complexe dans les grands réseaux. La question n'a pas fait l'objet d'un consensus à l'atelier, loin de là.

Deuxième grande question après l'architecture, le sommeil des objets (section 3.2). Pour économiser l'énergie, beaucoup de ces objets contraints s'endorment souvent. Pour qu'une pile AAA tienne des mois, il faut que l'objet dorme pendant 99, voire 99,9 % du temps. Chaque bit, chaque aller-retour sur le réseau, et chaque milli-seconde d'activité radio (une activité qui consomme beaucoup de courant) est précieux. Or, la plupart des protocoles TCP/IP sont conçus autour de l'idée que les machines sont allumées en permanence, qu'elles peuvent garder un état et répondre à tout moment à un paquet, même non sollicité. Mais, lorsque la simple attente d'un éventuel message consomme du courant, cette idée n'est plus valable : les objets contraints sont plus souvent endormis qu'éveillés, contrairement aux ordinateurs classiques.

Pire, lorsqu'un nœud se réveille après une période de sommeil, son adresse IP a pu être prise par un autre. Il va donc devoir se lancer dans une procédure coûteuse pour obtenir une adresse. Il peut toutefois économiser des efforts s'il met en œuvre les méthodes DNS ("*Detecting Network Attachment*") des RFC 4436 [1] et RFC 6059. Autre piège, si l'objet est mobile, après son réveil, il a pu bouger pour un tout autre endroit.

Pour gérer ces machines « Belle au bois dormant », les solutions envisagées lors de l'atelier étaient :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4436.txt`

— Ne plus considérer comme acquis que les machines sont toujours allumées, lorsqu'on conçoit un protocole. Prévoir une réduction des fonctions du protocole, ou bien des intermédiaires, toujours allumés, qui reçoivent les messages et les stockent (on nomme cela un "*sleep proxy*", comme dans la norme ECMA-393 <http://www.ecma-international.org/publications/standards/Ecma-393.htm>). Autre solution, que les machines communiquent leurs heures de réveil à leurs pairs, comme dans 802.11.
— Réduire le prix d'une connexion initiale au réseau. Actuellement, l'arrivée sur un nouveau réseau (et un nœud mobile, lorsqu'il se réveille, doit toujours supposer qu'il est sur un nouveau réseau) met en jeu des protocoles coûteux en énergie comme DHCP ou NDP. Peut-on réduire ce coût, surtout pour les machines qui savent qu'elles ne sont pas mobiles ?
— Réduire le coût des échanges de messages. Préferer les protocoles qui limitent le bla-bla (plus le dialogue est long, plus il faut rester allumé longtemps), les encodages qui ne consomment pas trop d'octets (donc, attention à XML),
— Penser au coût total en énergie : le RFC cite l'exemple de l'absurdité de la domotique où des ampoules allumées et éteintes par un contrôleur consomment soi-disant moins d'énergie... alors que le bilan global est négatif, à cause de la consommation du contrôleur et du réseau.

Troisième grande question explorée lors de l'atelier, la sécurité (section 3.3). Il faut évidemment la prendre en compte dès le début (RFC 3552 et RFC 4101). L'idéal est que la sécurité soit intégrée dans le début, au lieu d'être une option qu'on oublie d'activer. Mais les défis pour les objets contraints sont énormes. Les calculs qu'impose la cryptographie ne vont pas dans le sens de l'économie d'énergie. D'autre part, l'absence d'une interface utilisateur commode limite sérieusement la configuration qui peut être faite (pas moyen de demander à l'administrateur d'entrer un mot de passe compliqué). Les protocoles de sécurité de l'IETF permettent de gérer tous les cas, mais ils peuvent être trop riches et on peut défendre l'idée de profils, de restrictions ne gardant, par exemple, que certains algorithmes de cryptographie. Le choix ne sera pas évident car il existe des exigences contradictoires. Par exemple, les algorithmes à base de courbes elliptiques (RFC 6090) sont souvent moins consommateurs de ressources matérielles mais sont bien plus encombrés de brevets.

Enfin, les objets communiquants ayant vocation à être partout (à l'usine, au bureau, mais aussi à la maison et, pourquoi pas, dans notre propre corps), les questions de protection de la vie privée sont cruciales. Pas question qu'un "*sniffer*" puisse espionner la communication entre le frigo et le supermarché, apprenant ainsi mes habitudes, mes heures de présence, etc. (Dans ce cas, le principal danger pour la vie privée vient probablement du supermarché et, là, le chiffrement des communications n'aide pas.)

Mais l'algorithme de cryptographie n'est pas tout : l'expérience de l'usage de la cryptographie dans l'Internet a montré que les problèmes de gestion des clés cryptographiques étaient souvent bien pires.

Enfin, dernière grande question analysée à Prague, le routage (section 3.4). Les objets communiquants peuvent être trop loin les uns des autres pour se parler directement et il peut être nécessaire de passer par un routeur. Comment les objets et les routeurs vont-ils apprendre les routes disponibles ? Il existe deux approches, "*mesh-under*" et "*route-over*". La première n'est pas vraiment du routage, elle consiste à résoudre le problème au niveau 2. Les objets ont ensuite tous l'impression d'être sur le même réseau local. L'autre approche, le "*route-over*", met en place des vrais routeurs IP. Cela implique donc un protocole de routage et celui officiel à l'IETF pour cette tâche est le RPL du RFC 6550 (mais il en existe d'autres, comme le Babel du RFC 6126).

Les protocoles de routage pour les objets contraints font face à de nombreux défis. Par exemple, en filaire classique, les caractéristiques (latence, perte de paquets) sont les mêmes pour toutes les machines attachées au même lien. Ici, c'est par contre loin d'être le cas. Le groupe de travail roll <http://tools.ietf.org/wg/roll> avait été formé pour étudier ce problème (qui avait déjà été abordé par le RFC 3561) et il avait produit plusieurs RFC de débroussaillage et de définition du problème (RFC 5867, RFC 5826, RFC 5673, RFC 5548), avant de définir RPL. Notez que le problème suscite des controverses. Un RFC d'étude comparée des différents protocoles de routage avait été abandonné par le groupe

roll (manque de consensus pour avancer). Et, à l'atelier, les polémiques n'ont pas manqué, comme la défense d'AODV par Thomas Clausen dans son article.

Alors, à la fin, peut-on faire une synthèse ? La section 4 s'y essaie. Premier thème décrit, la sécurité. La consommation de temps de processeur (et donc d'électricité) par la cryptographie est une sérieuse limite à la sécurisation des objets communicants. La difficulté de la gestion des clés en est une autre. Notre RFC 6574 recommande d'y travailler sérieusement sur ce dernier point, en donnant comme exemple le mécanisme d'appariement de Bluetooth qui combine harmonieusement sécurité et simplicité. Un groupe de travail de l'IETF avait été formé sur cette question, enroll <http://tools.ietf.org/wg/enroll>, mais avait échoué. Peut-être le travail devra-t-il être repris par le nouveau groupe lwig <http://tools.ietf.org/wg/lwig>.

Quant à la question des algorithmes cryptographiques « développement durable » (à consommation de ressources réduite), un groupe de recherche de l'IRTF existe, cfrg <http://irtf.org/cfrg>. Par exemple, le futur SHA-3 a prévu de prendre en compte ce problème (dont l'importance n'était pas perçue pour les algorithmes précédents de la famille).

Assurer l'interopérabilité n'est pas évident lorsque des objets ont une mémoire limitée et ne peuvent pas stocker le code correspondant à tous les algorithmes existants. Le RFC rappelle que plus de cent algorithmes de cryptographie sont définis pour TLS, dont certains sont officiellement déconseillés depuis cinq ou parfois dix ans, mais toujours répandus dans la nature : une mise en œuvre de TLS qui veut être sûre de pouvoir communiquer avec tout le monde a besoin de les connaitre. Les nouveaux algorithmes à consommation d'énergie réduite pourraient encore aggraver ce problème. Faut-il créer un nouvel algorithme, avec les problèmes d'appropriation intellectuelle, avec les difficultés de déploiement, juste pour économiser 20 % de la consommation électrique ? Le RFC laisse entendre que le gain de consommation devrait être nettement plus élevé pour que le jeu en vaille la chandelle.

Un autre thème choisi pour la conclusion est justement celui de la consommation électrique. Il existe une liste de discussion active pour cela, recipe <https://www.ietf.org/mailman/listinfo/recipe>, et l'article de Margaret Wasserman à l'atelier fournit un bon point de départ pour ceux qui veulent s'engager dans ce travail.

Quant à la question du réseau « centré sur le contenu », le RFC conclut que c'est encore bien trop nébuleux et qu'il n'y a pas de travail concret de normalisation à envisager. Il s'agit de recherche pure pour l'instant <http://www.ietf.org/mail-archive/web/irtf-discuss/current/msg00041.html>.

Sur l'architecture, le RFC plaide en faveur d'un futur document de synthèse de l'IAB expliquant comment tous les protocoles Internet marchent ensemble dans ce contexte des objets contraints et communiquants. Le RFC 6272 fournit un bon exemple d'un tel travail.

Dans la lignée de l'exemple de l'ampoule électrique, le RFC insiste aussi sur l'importance de développer des modèles de données (« Une ampoule a deux états, allumé et éteint ») permettant de créer des applications (ici, de contrôle de l'ampoule). La question est « est-ce bien le travail de l'IETF, puisqu'il faut à chaque fois une expertise spécifique d'un domaine ? ».

Pour la partie « découverte » de services ou de machines, le RFC recommande de travailler avec mDNS ou équivalent.

Pour le routage, notre section de conclusion suggérait de travailler entre autres sur la question des « sous-réseaux couvrant plusieurs liens » (RFC 4903). L'essentiel du travail sur le routage pour les objets communiquants continue au sein du groupe roll <http://tools.ietf.org/wg/roll>.

Notez enfin le récent groupe de travail homenet <http://tools.ietf.org/wg/homenet> qui travaille sur les problèmes de domotique (voir son RFC 7368).

La liste des présentations à l'atelier figure dans l'annexe B. On trouve en ligne la page officielle de l'atelier <http://www.iab.org/about/workshops/smartobjects/>, les articles présentés <http://www.iab.org/about/workshops/smartobjects/papers/> (une impressionnante masse de bons documents à lire), les supports des présentations <http://www.iab.org/about/workshops/smartobjects/agenda.html>, et les notes prises pendant l'atelier <http://www.iab.org/activities/workshops/smartobjects/smartobjectworkshopmeetingminutes/>.