RFC 6272 : Internet Protocols for the Smart Grid

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 juin 2011

Date de publication du RFC : Juin 2011

<https://www.bortzmeyer.org/6272.html>

Le titre de ce RFC ne rend pas complètement compte de son but. Ce document est en fait une présentation complète de la suite de protocoles TCP/IP, celle qui fait tourner l'Internet. Conçu pour des ingénieurs qui viennent d'une autre culture que celle des barbus Unix, récapitulant tout ce qu'il faut savoir sur les protocoles de la famille, ce texte est utilisable comme base d'un cours, ou comme introduction à TCP/IP. Il a été conçu pour le projet "*Smart Grid*".

En effet, ce projet de distribution de l'électricité en faisant appel à des équipements plus proches des ordinateurs que des traditionnels transformateurs et compteurs, équipements qui communiqueraient beaucoup entre eux, cherche une architecture réseau. Un certain nombre de consultants, souvent ignorants de ce qui s'est fait depuis trente ans sur l'Internet, prône le développement en partant de zéro de nouvelles techniques, non prouvées et coûteuses à créer, alors que TCP/IP fournit déjà tout ce qu'il faut. Ce RFC explique TCP/IP pour les débutants, et fournit une liste des protocoles, dont peut être tirée un futur **profil** "*Smart Grid*" (un profil étant un sous-ensemble adapté à une tâche donnée). Voir la section 1 à ce sujet.

Comme je ne connais pas assez "*Smart Grid*", et que le RFC ne contient quasiment rien de spécifique à ce projet (l'annexe A rassemble les informations purement "*Smart Grid*"), je vais adopter une autre perspective en utilisant ce RFC comme moyen de présenter les protocoles d'Internet, à un ingénieur qui ne les connaîtrait pas. Ces protocoles sont désignés dans le RFC sous le nom d'**IPS**, "*Internet Protocol Suite*". À partir d'ici, si vous connaissez déjà l'IPS, vous avez le droit de partir et d'aller voir un film (par exemple *Le chat du rabbin*) ou de lire un livre (par exemple *Le chat du rabbin*).

L'IPS est décrite dans les fameux RFC. Outre la normalisation des protocoles, certains RFC sont là pour décrire l'implémentation par exemple les RFC 1122¹ et RFC 1123 sur les machines terminales et le

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc1122.txt>

RFC 1812 sur les routeurs. (Avec une version des spécificités IPv6 dans le RFC 8504.) Une des particularités de l'IPS est qu'elle offre des choix : ainsi, il existe plusieurs protocoles de transport comme TCP (RFC 793) ou SCTP (RFC 9260) et même un « non-protocole », UDP (RFC 768), où l'application doit faire tout le travail de la couche 4. La section 2 de ce RFC va présenter l'architecture de l'IPS, la section 3 une liste de protocoles et la section 4 l'organisation non-technique de l'Internet (relations d'affaires, etc).

Donc, c'est parti, comment tous ces protocoles tiennent-ils ensemble? L'"Internet Protocol Suite" suit la classique organisation en couches (section 2.1). Si le principe est celui du modèle OSI, le découpage des couches n'est pas forcément le même, ni la terminologie. Ainsi, l'OSI parle (en anglais) de "*end system*" lorsque l'IPS dit "*host*" (que je traduis par machine terminale <<https://www.bortzmeyer.org/terminal-host.html>>) et l'OSI dit "*intermediate system*" où l'IPS dit "*router*".

En partant du haut, de ce qui est le plus proche des utilisateurs, on trouve la couche Applications (section 2.1.1). L'IPS ne fait pas de distinction forte entre les couches 5, 6 et 7, considérant plutôt une seule couche Applications.

Ensuite, toujours en allant vers le bas, l'IPS a une couche de Transport (section 2.1.2). Le terme de transport est très galvaudé. Énormément d'ingénieurs l'utilisent pour dire « les couches du dessous qui ne m'intéressent pas ». Un concepteur d'un réseau optique va parler du transport pour la fibre sous-jacente, alors que le développeur d'une application Web va considérer que le protocole HTTP est le transport, pour son application. Mais dans l'IPS, « transport » a un sens bien précis, c'est la plus basse couche qui va de bout en bout. Des protocoles archi-connus comme TCP (RFC 793) ou très exotiques comme NORM (RFC 5740) mettent en œuvre ce transport. Comme plusieurs flots de données simultanés peuvent exister entre deux adresses IP, le démultiplexage se fait sur la base d'un numéro de port géré par la couche transport. Deux flots différentes auront ainsi des couples {port source, port destination} différents. Les différents protocoles de transport fournissent en outre des services tels que garantie de délivrance des données (ou pas), gestion de messages séparés (ou bien flot d'octets continu, ce que fait TCP), contrôle de la congestion.

Continuons à descendre vers les couches bases, la couche Réseau (section 2.1.3) vient ensuite. Au contraire du choix qui existe en couche Transport, il n'y a qu'un seul protocole de couche Réseau, IP (ou deux, si on considère IPv4 et IPv6 comme différents). IP fournit un service de datagrammes : chaque paquet est indépendant des autres, porte donc les adresses IP de source et de destination, et il n'y a pas besoin d'établir un circuit avant d'envoyer des données. IP fournit également un service de fragmentation, permettant de découper un paquet trop gros avant de le réassembler à la fin.

IP peut fonctionner sur d'innombrables protocoles de couche 2. L'IETF n'en normalise aucun (à l'exception possible de MPLS, RFC 4364, tout dépend comment on place ce protocole dans le modèle en couches). La plupart du temps, ces protocoles des couches basses (section 2.1.4) sont normalisés par l'IEEE : 802.3, 802.11, 802.16, etc. IP peut aussi tourner sur IP, via des tunnels comme GRE (RFC 2784).

La famille TCP/IP a souvent été critiquée pour son manque de sécurité (section 2.2). En fait, toute la sécurité dans l'Internet ne dépend pas de TCP/IP. Le hameçonnage, par exemple, est une question d'éducation et d'interface utilisateur, deux points très importants mais qui ne sont pas du ressort de l'IETF. D'autres problèmes (comme la faille BGP dite « attribut 99 <<https://www.bortzmeyer.org/bgp-attribut-99.html>> ») ne relèvent pas non plus de la sécurité du protocole puisqu'ils sont causés par une bogue dans un programme mettant en œuvre le protocole.

D'autres questions de sécurité viennent de l'organisation de l'Internet, pas de ses protocoles. Ainsi, il n'existe pas de chef de l'Internet, qui pourrait ordonner que les vieilles versions des logiciels, versions

ayant des bogues connues, soient immédiatement retirées du service. On constate qu'elles traînent souvent dans la nature. De même, les nouvelles techniques de sécurité ont souvent du mal à être déployées, puisqu'aucune autorité ne peut imposer leur emploi. Il faut au contraire convaincre chaque administrateur réseaux. (Des exemples de protocole de sécurité dont le déploiement a été faible sont SPF - RFC 4408 et IPsec - RFC 4301. Un exemple d'un protocole de sécurité qui a mis très longtemps à être déployé est DNSSEC - RFC 4033.)

Cela ne veut pas dire que l'IETF baisse les bras sur la sécurité. Chaque nouveau protocole doit inclure une analyse de sécurité, et le RFC 3552 guide les auteurs pour cette tâche.

Les problèmes de sécurité de la couche physique se traitent en général par des moyens physiques (enterrer et protéger les câbles, par exemple). Pour les autres couches, le problème est posé en termes de trois services : Confidentialité, Intégrité et Disponibilité. On souhaite que les conversations privées restent privées (Confidentialité), on souhaite que les données soient authentifiées au départ, puis arrivent sans modifications (Intégrité) et on souhaite que l'Internet marche (Disponibilité). Voici deux exemples :

- Une attaque classique contre TCP est d'envoyer des faux paquets RST ("*ReSeT*"), faux car utilisant une adresse IP source fabriquée, pour casser la connexion en cours. Il existe plusieurs protections, comme l'authentification TCP (RFC 5925), ou bien une sécurité au niveau en dessous comme IPsec. (Le RFC laisse entendre que TLS est une autre solution mais il ne protège pas du tout contre ces attaques.)
- Une autre attaque contre TCP est le "*SYN flooding*" où l'attaquant tente d'ouvrir suffisamment de connexions TCP pour épuiser les ressources de l'attaqué (la liste des connexions est typiquement de taille finie). Il existe aussi des protections comme les "*SYN cookies*" du RFC 4987.
- Dans les deux cas précédents, l'attaque portait sur la Disponibilité. Un exemple d'attaque contre la Confidentialité serait un accès non autorisé au réseau.

Dans beaucoup de cas, une première étape vers la sécurisation est la création d'un canal sûr entre la source et la destination (TLS ou IPsec le permettent.) Mais attention, ce n'est pas suffisant de sécuriser la communication, car le pair peut lui-même être malveillant, délibérément ou à son insu.

Enfin, l'infrastructure de l'Internet compte également des protocoles qui sont officiellement dans les applications mais sont en pratique nécessaires au bon fonctionnement. La section 2.3.1 mentionne le DNS, système de correspondance entre un nom de domaine et des données (typiquement des adresses IP). Bien sûr, l'Internet peut fonctionner sans le DNS. Mais, en pratique, cela serait vite insupportable (contrairement à ce qu'on lit parfois, la plupart des sites Web ne sont pas accessibles via leur adresse IP, voir section 5.4 du RFC 7230). Et la section 2.3.2 mentionne les outils de gestion de réseau comme les deux protocoles de gestion SNMP et Netconf (RFC 4741). (Dans le projet français PREMIO <http://www.smartgrid.epri.com/doc/10_Premio%20Project%20Progress.pdf>, SNMP est utilisé.)

La section 3 du RFC attaque ensuite les protocoles spécifiques. L'idée, là encore, est d'utiliser pour la "*Smart Grid*" des protocoles bien conçus, éprouvés, et disposant de mises en œuvre nombreuses et de qualité, souvent en logiciel libre.

Cette section commence avec la sécurité, La section 3.1 se penche d'abord sur l'AAA. Dans le contexte de l'Internet, cela concerne les protocoles qui autorisent l'accès au réseau, Radius (RFC 2865 et Diameter (RFC 6733). Ces protocoles utilisent souvent un mécanisme d'authentification nommé EAP (RFC 4017; opinion personnelle, EAP est d'une grande complexité), qui permet à son tour plusieurs méthodes (RFC 5216, RFC 5433).

Une fois qu'on a accès au réseau, reste à protéger les communications. Au niveau 3, une solution générale est IPsec (RFC 4301), qui « cryptographie » tout le trafic IP, fournit intégrité et authentification, aussi bien que confidentialité (IPsec est une architecture, pas un protocole, donc la liste des services peut dépendre du protocole). Pour l'échange des clés cryptographiques, cela peut se faire à la main (sans

doute la technique la plus courante aujourd'hui) ou bien via le protocole IKE (RFC 5996). À noter que notre RFC 6272, bien que se voulant pratique, mentionne rarement de choix lorsque plusieurs protocoles existent. Toutefois, pour IPsec, il note que ESP (RFC 4303) est de très loin le plus utilisé et qu'on peut oublier les autres.

La sécurité peut-elle être assurée dans la couche 4? Oui, grâce au très connu protocole TLS (RFC 5246). TLS chiffre toute la communication de l'application. Il ne protège pas contre les attaques situées plus bas, mais est bien plus simple à déployer qu'IPsec.

Enfin, la sécurité peut être appliquée au niveau des applications, où on a le maximum de souplesse. La famille TCP/IP a plusieurs mécanismes dans sa boîte à outils pour cela :

- CMS (RFC 5652) est une syntaxe pour avoir des messages authentifiés et/ou confidentiels, en utilisant la cryptographie. Cette syntaxe peut ensuite être utilisée dans des applications comme les mises à jour de logiciels (RFC 4108) ou bien le courrier électronique (RFC 5751 sur S/MIME, à noter qu'il existe une syntaxe concurrent, celle d'OpenPGP, RFC 4880, non mentionnée par le RFC).
- Si on préfère le format XML, le RFC 3275 décrit un moyen de signer XML.
- L'authentification peut utiliser OAuth (RFC 5849).

Toujours dans la boîte à outils, la famille TCP/IP a aussi un protocole de connexion sécurisée à distance, SSH (RFC 4253), qui peut aussi servir de mécanisme pour faire des tunnels sûrs (en concurrent d'IPsec, donc).

Toute solution fondée sur la cryptographie a besoin d'un mécanisme de gestion des clés. L'IETF en a au moins deux, PKIX (RFC 5280), fondé sur X.509, et Kerberos (RFC 4120), le second étant plutôt limité au cas où tout les participants sont dans le même domaine administratif.

Après ces considérations sur la sécurité, place à la description des protocoles de couche 3. L'Internet en a actuellement deux, IPv4 et IPv6 (section 3.2). IPv4 est décrit en détail dans la section 3.2.2. Je passe rapidement, pour arriver à IPv6 en section 3.2.3. IPv6 est normalisé dans le RFC 2460. À noter que, pour le cas de la "*Smart Grid*", dont certains équipements peuvent avoir des capacités et des connexions peu performantes, les RFC comme le RFC 4919 peuvent être également utiles.

Les adresses IPv6 sont décrites dans le RFC 4291. Elles peuvent être distribuées aux machines terminales via le protocole DHCP (RFC 8415) ou bien sans serveur par un système d'auto-configuration (RFC 4862). Deux machines IPv6 dans le même réseau se trouvent par le protocole NDP (RFC 4861).

Le routage des paquets IPv6 se fait d'abord par la distribution des informations entre routeurs, puis par l'utilisation de ces routes lors de la transmission de chaque paquet. La route la plus spécifique (le préfixe le plus long) est utilisée. Pour distribuer les routes, les routeurs peuvent utiliser divers protocoles comme OSPF (RFC 5340 et qui peut router IPv4 comme IPv6, grâce au RFC 5838) et BGP (RFC 2545). OSPF est utilisé à l'intérieur d'un domaine administratif et fait partie des protocoles « à état des liens » dont l'avantage est que chaque routeur a une vue complète du réseau (au lieu de router par « on-dits »). BGP est utilisé entre domaines administratifs et c'est donc lui qui lie les routeurs de l'Internet.

Il y a aussi des protocoles de routage spécifiques aux réseaux de capteurs, les équipements peu puissants dont je parlais plus haut. OSPF, où chaque routeur doit garder en mémoire tous les liens de tout le réseau, ne convient pas forcément pour cette tâche. Parmi les concurrents plus adaptés, on trouve par exemple AODV (RFC 3561) ou d'autres protocoles en cours de définition (voir le cahier des charges dans les RFC 5548, RFC 5673, RFC 5826 et RFC 5867).

Depuis l'épuisement des adresses IPv4 <<https://www.bortzmeyer.org/epuisement-adresses-ipv4.html>>, la migration vers IPv6 est nécessaire mais elle n'est pas instantanée et il faut donc gérer la co-existence entre les deux protocoles. Il existe plusieurs mécanismes pour cela mais tous ne sont pas recommandés (cf. RFC 6180). Le plus simple et le plus conseillé est la « double-pile » : le réseau et les machines ont à la fois IPv4 et IPv6, jusqu'à ce qu'on puisse éteindre la dernière adresse v4 (RFC 4213). Si on ne peut pas disposer d'une connectivité IPv6 native, l'approche recommandée est de tunneler le trafic v6 à travers l'Internet, par exemple par les techniques du RFC 5569. Enfin, si une des deux machines participant à la communication est purement v4 ou purement v6, il faut utiliser les techniques de traduction. Celles-ci recelant des pièges inattendus, le mieux est de préférer la traduction faite au niveau des applications (par des relais applicatifs). Sinon, l'IETF a normalisé un cadre général de traduction v4- ζ v6 dans le RFC 6144, avec des déclinaisons dans des protocoles comme celui du RFC 6146.

Continuant à monter vers les couches hautes, la section 3.3 présente les protocoles de transport de l'IPS ("*Internet Protocol Suite*"). Le plus connu est TCP, responsable sans doute de la très grande majorité du trafic de l'Internet. TCP garantit aux applications qui l'utilisent la délivrance des octets, dans l'ordre et sans perte. Si le RFC qui le normalise date de trente ans, TCP a subi pas mal de travaux et de modifications et il vaut donc mieux commencer par lire le RFC 7414 pour tout savoir de ce protocole. Par exemple, pour la sécurité de TCP, le RFC 4987 est une bonne lecture.

Autre protocole de transport, UDP (RFC 768), un protocole minimal qui n'assure presque aucune fonction : l'application qui l'utilise doit tout prévoir, le contrôle de congestion (ne pas envoyer trop de paquets pour ne pas surcharger le réseau), la détection des pertes (et réessayer si nécessaire), etc. Le RFC 8085 guide les applications à ce sujet.

Il existe d'autres protocoles de transport qui n'ont guère eu de succès comme SCTP (RFC 9260) ou DCCP (RFC 4340), une sorte d'UDP avec contrôle de congestion.

Pour les protocoles d'infrastructure, la section 3.4 mentionne :

- Le DNS : des noms hiérarchiques (`www.example.com` est un sous-domaine de `example.com`, lui-même sous-domaine de `.com`) et un protocole de résolution de noms en données (RFC 1034). Il existe un mécanisme de sécurisation, décrit par le RFC 4033.
- DHCP (RFC 2131 en IPv4 et RFC 8415 en IPv6) permet d'attribuer des adresses IP (et de distribuer d'autres informations comme l'adresse du serveur de temps) aux différentes machines d'un réseau local depuis un point central. Il facilite donc beaucoup la vie de l'administrateur réseaux. Sans lui, lorsqu'on voudrait renuméroter les machines, il faudrait se connecter à chaque machine pour changer sa configuration. (Regardez comment on faisait en 1992 <http://groups.google.com/group/comp.protocols.tcp-ip.domains/browse_thread/thread/e47fca41f772c48e/6df6904950955a87>.)
- Je viens de parler du serveur de temps. Le protocole NTP (RFC 5905) permet justement de mettre à l'heure les horloges de toutes les machines de l'Internet et s'assurer qu'elles soient d'accord. C'est une tâche plus compliquée qu'elle n'en a l'air.

Et la gestion de réseaux (section 3.5)? Les deux protocoles concurrents sont SNMP (RFC 3411 et RFC 3418) et Netconf (RFC 4741). Lequel choisir? Notre RFC 6272 ne fournit pas d'indications à ce sujet... La seule information de comparaison donnée est que SNMP utilise l'encodage ASN.1 alors que Netconf utilise XML.

Une machine qui démarre a souvent besoin d'informations sur son environnement, par exemple les coordonnées d'une imprimante ou bien d'un serveur de courrier. C'est le domaine des protocoles de découverte (section 3.6). C'est depuis longtemps une faiblesse de la famille TCP/IP. Le problème n'est pas complètement résolu de manière standard. Le RFC cite un protocole comme SLP (RFC 2608, qui n'a jamais été très utilisé) ou comme le protocole d'Apple, Bonjour. D'autres protocoles sont en cours

de déploiement comme COAP ("*Constrained Application Protocol*", RFC 7252), qui utilise HTTP pour demander à un serveur ces informations, en utilisant les adresses « bien connues » du RFC 8615.

Enfin, le RFC 6272 cite également (section 3.7) un certain nombre d'applications, qui ne font pas partie de l'IPS à proprement parler, mais qui peuvent être utiles à des déploiements sur TCP/IP. Par exemple, la famille TCP/IP a un protocole standard d'établissement de sessions multimédia, SIP (RFC 3261 et beaucoup d'autres). SIP est surtout utilisé pour la téléphonie sur IP mais a d'autres usages. SIP ne fait que la signalisation (appeler, raccrocher, etc) et la description de la session (voix, vidéo, codecs utilisés, est faite avec le format SDP (RFC 4566). En pratique, SIP a pas mal de problèmes sur l'Internet, face à la fermeture croissante du réseau par des mécanismes comme le NAT (le RFC 2993 décrit le problème et le RFC 5626 propose des solutions).

Autre application qui peut servir, XMPP (RFC 6120), un protocole d'échange de données XML en temps réel dont l'utilisation la plus connue est pour la messagerie instantanée (RFC 6121). XMPP est très riche et dispose de nombreuses extensions (par exemple pour du dialogue à plusieurs, dans des « pièces » virtuelles).

Après cette longue présentation des différents protocoles de la famille, le RFC expose l'architecture non-technique de l'Internet, c'est-à-dire son organisation sociale. Rien de nouveau pour les professionnels de l'Internet mais rappelez-vous que ce RFC est conçu pour des gens d'une culture très différente. La section 4 explique donc que l'Internet est un réseau de réseaux (c'est en cela que son ancêtre, l'Arpanet, était très différent : il était mono-réseau). Ces réseaux s'interconnectent entre eux (sinon, un abonné d'un réseau ne pourrait pas joindre un abonné d'un autre, ce qui était courant avec les technologies pré-Internet). Cette interconnexion est le cœur de l'architecture sociale de l'Internet dont le but est que tout le monde puisse parler à tout le monde (note personnelle au passage : cette interconnexion généralisée semble une banalité aujourd'hui. Mais, dans les années 1990, il était fréquent d'entendre des messieurs sérieux expliquer qu'il n'était pas nécessaire qu'une entreprise doive se connecter à d'autres que ses sous-traitants.)

On peut présenter les différents réseaux qui composent l'Internet en disant qu'il y a une différence entre les réseaux de transit et les autres. Les réseaux de transit forment le cœur de l'Internet. Leur seule activité est de connecter les autres réseaux. Ils vendent ce service aux réseaux d'accès et réseaux d'organisations. Les réseaux d'accès sont ceux du FAI typique, qui connecte M. Toutlemonde (particuliers et petites organisations) moyennant un paiement de trente € par mois. L'accès se fait via ADSL, RTC, 3G ou bien d'autres techniques. Les réseaux d'organisation sont ceux des entreprises, associations, collectivités locales ou universités, qui fournissent de l'accès à leurs employés et étudiants.

Si je fais de chez moi (abonné à Free) un traceroute vers ce blog, je passe successivement par le réseau d'accès du groupe Iliad (Free et Proxad), puis par le réseau de transit Cogent, puis j'arrive chez Datotel, le réseau d'entreprise où le serveur est hébergé :

```
% traceroute www.bortzmeyer.org
traceroute to www.bortzmeyer.org (208.75.84.80), 30 hops max, 60 byte packets
 1 freebox (192.168.2.254)  5.984 ms  7.027 ms  7.068 ms
...
 4 cha75-6k-1-v906.intf.nra.proxad.net (78.254.255.197)  35.671 ms  35.726 ms  35.766 ms
 5 bzn-crs16-1-be1502.intf.routers.proxad.net (212.27.58.65)  35.806 ms  37.060 ms  37.125 ms
 6 bzn-6k-2-po57.intf.routers.proxad.net (212.27.59.217)  38.222 ms * *
 7 th2-crs16-1-be1005.intf.routers.proxad.net (212.27.56.5)  39.893 ms  23.638 ms *
 8 te4-8.330.ccr01.par04.atlas.cogentco.com (149.6.164.221)  44.829 ms  45.136 ms  49.465 ms
 9 te0-1-0-4.mpd21.par01.atlas.cogentco.com (130.117.2.77)  55.562 ms  53.303 ms  54.986 ms
...
16 host46.datotel.com (208.82.151.46)  127.897 ms  127.465 ms  127.109 ms
17 st1-d2-g5-1.datotel.com (208.82.151.26)  128.295 ms  126.546 ms  127.575 ms
18 host125.datotel.com (208.75.82.125)  127.461 ms * *
...
```

(Les étoiles représentent des cas où un paquet a été perdu.) Notez que le routeur chez moi, la Freebox, avait une adresse privée (RFC 1918).

Le RFC ne parle pas des relations de "peering" entre réseaux. À ce sujet, un très intéressant document est le rapport de PCH <<https://www.pch.net/resources/papers/peering-survey/PCH-Peering-Survey-2011.pdf>>.

À noter que le RFC n'expose que l'architecture sociale, pas la gouvernance de l'Internet, vaste sujet mais fort loin des problèmes opérationnels...

Souvent, les traceroutes sont interrompus subitement et plus rien n'est visible. C'est souvent le résultat d'un pare-feu mis sur le trajet. Notre RFC note que ces équipements sont très fréquents dans l'Internet d'aujourd'hui, mais aussi qu'ils sont très controversés. En effet, leur efficacité n'est pas prouvée, la plupart des attaques provenant de l'intérieur (soit parce qu'une personne située dans l'organisation mène l'attaque, soit parce qu'une machine interne a été infectée par un logiciel malveillant, le pare-feu ne protégeant souvent pas contre les attaques par « charge utile »). Bref, le pare-feu ne doit être utilisé que comme composante d'une solution de sécurité, pas comme l'alpha et l'oméga de la sécurité. Il existe plusieurs RFC ayant un rapport avec les pare-feux comme le RFC 2647 (mesures de performance), RFC 2979 (règles que doivent suivre les pare-feux), RFC 5207 (un exemple des problèmes que peuvent poser les pare-feux pour un nouveau protocole), etc.

Autre source de problèmes lorsqu'on circule dans l'Internet, les NAT. Cette technique a été inventée vers 1993 pour pallier le manque d'adresses IPv4 en traduisant des multiples adresses privées du réseau interne vers la seule adresse IP publique. En IPv6, le NAT est devenu inutile, vue l'abondance d'adresses. Mais comme certaines personnes s'imaginent que le NAT joue aussi un rôle pour la sécurité, en servant de pare-feu, il reste très présent et gêne souvent les communications. Les NAT sont décrits dans les RFC 2663 et RFC 3022, leurs conséquences néfastes pour les communications sont expliquées dans le RFC 3027, des idées pour les développeurs d'applications dans un monde où il y a beaucoup de NAT ont été rassemblées dans le RFC 3235, d'autres techniques de contournement sont dans le RFC 5128, quant à la position officielle de l'IAB sur les NAT, elle est dans le RFC 3424.

Sur le projet "Smart Grid", si celui-ci n'a pas encore d'architecture réseau normalisée, il a déjà des documents. On peut consulter « *Smart Grid Architecture Committee : Conceptual Model White Paper* » <http://collaborate.nist.gov/twiki-sggrid/pub/SmartGrid/SGIPConceptualModelDevelopmentSGAC/Smart_Grid_Conceptual_Model_20100420.doc> » (document uniquement en format privé MS-Word, ce qui donne une idée de l'écart de culture avec l'Internet) et IEC 61850.