RFC 7326 : Energy Management Framework

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 6 septembre 2014

Date de publication du RFC : Septembre 2014

https://www.bortzmeyer.org/7326.html

Ce nouveau RFC présente un cadre général pour la gestion de l'énergie dans les protocoles IETF. La préoccupation est assez récente. Pendant longtemps, les ordinateurs et routeurs connectés à l'Internet consommaient autant de courant qu'ils voulaient et personne ne regardait la facture. La montée des préoccupations écologiques, le prix de plus en plus élevé de l'électricité dans les centres de données, et le désir d'avoir des machines sans fil à la patte, alimentées uniquement par une batterie, font que la gestion de l'énergie est maintenant devenue une préoccupation à part entière de l'IETF. Ce cadre modélise les consommateurs d'électricité comme des objets énergétiques ("*energy objects*") qui vont pouvoir être supervisés et contrôlés à distance. On pourra, par exemple, par des protocoles normalisés, suivre la consommation électrique d'un serveur, ou bien la quantité d'énergie restante dans une batterie mais aussi faire passer une machine dans un état à consommation énergétique réduite.

La gestion de réseaux informatiques est divisée par la norme X.700 en cinq parties, Panne, Configuration, Comptabilité, Performance et Sécurité. La gestion de l'énergie n'y figure pas. La norme ISO 50001 ajoute cette préoccupation aux textes. L'IETF a créé un groupe de travail sur la gestion de l'énergie, EMAN <https://tools.ietf.org/wg/eman>, qui avait déjà produit un premier RFC, le cahier des charges, le RFC 6988 [1].

Notre nouveau RFC introduit le concept d'interface énergie, en s'inspirant de celui bien connu d'interface réseau. C'est par une interface énergie que la machine fournit ou consomme de l'énergie. Une machine ne sait pas forcément mesurer ce qu'elle consomme et c'est donc parfois en interrogeant le dispositif de fourniture d'énergie qu'on aura cette information. (L'interface énergie de sortie de l'un étant l'interface énergie d'entrée de l'autre.)

Autres termes définis par ce RFC (section 2) :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6988.txt

— Entrée (”*power inlet*” ou simplement ”*inlet*”) : l'interface énergie par laquelle le courant arrive.
— Sortie (”*outlet*”) : l'interface énergie par laquelle le courant sort.
— Énergie : le nombre de kWh consommés ou produits, ou bien qu'on peut consommer ou produire.
— Puissance : énergie divisée par le temps (en watts ou en joules/seconde). Si l'énergie est une distance, la puissance est la vitesse.
— Demande : puissance moyennée sur un intervalle. Ces trois dernières définitions sont tirées de IEEE 100.
— Attributs du courant : la tension, la phase, la fréquence, etc.
— Qualité du courant : ses attributs par rapport à une référence. Comme le précédent, ce terme vient de IEC 60050.

La section 3 du RFC définit la notion de cible (”*target device*”). Ce sont tous les engins qui peuvent être supervisés ou contrôlés dans le cadre défini dans ce RFC : ordinateurs de bureau, serveurs, routeurs, imprimantes, commutateurs, points d'accès WiFi, batteries (lorsqu'elles sont autonomes et non pas enfermés dans un ordinateur), générateurs, fournisseurs PoE, compteurs, capteurs divers... Plusieurs de ces engins ne parleront pas IP et devront être interrogés / commandés via un relais.

Le cadre général défini par ce RFC concerne l'énergie, mais il ne couvre pas toutes les applications liées à l'énergie. La section 5 liste ce qui n'est pas l'objet du travail EMAN en cours : les engins non-électriques (si vous avez un routeur ”*steampunk*”, propulsé par la vapeur, EMAN ne va pas vous aider) et la production électrique (seules sa distribution et sa consommation sont prises en compte).

La section 6 du RFC décrit le modèle utilisé pour la gestion de l'énergie. Il suit une conception objets classique : une classe `Energy Object`, avec trois sous-classes, `Device`, `Component` et `Power Interface`. La super-classe `Energy Object` modélise tout ce qui est relié au réseau et qui l'utilise pour superviser ou commander sa gestion de l'énergie. La classe `Device` modélise les équipements physiques qui consomment, distribuent ou stockent de l'énergie, comme les ordinateurs. `Component` sert à décrire les parties d'un objet `Device`, et le nouveau concept, `Power Interface` représente les interconnexions.

Tous les objets héritent de `Energy Object` qui a comme attributs :
— Un identificateur unique, un UUID (RFC 9562),
— Un nom (relativement) lisible ; cela pourra être un nom de domaine (pour les ordinateurs et routeurs, c'est déjà le cas), ou d'autres conventions de nommage,
— Une importance, allant de 1 à 100, et qui sera utilisée pour prendre des décisions comme « les accumulateurs sont presque à plat, qui est-ce que je coupe en premier ? »,
— Une série d'étiquettes (”*tags*”) permettant de trouver et de manipuler facilement tous les objets ayant une caractéristique commune,
— Un rôle, qui indique le but principal de cet objet (routeur, lampe, réfrigérateur, etc),
— Et quelques autres attributs que je n'ai pas cité ici.

Les objets de la classe `Energy Object` ont également des méthodes permettant la mesure de certaines grandeurs : puissance, attributs du courant électrique, énergie et demande.

D'autres méthodes permettent le contrôle de l'objet et et changer son état. Plusieurs normes donnent des listes d'état possibles pour un objet. Par exemple, IEEE 1621 décrit trois états : allumé, éteint et dormant. D'autres normes, comme ACPI, vont avoir une liste différente. DMTF décrit pas moins de quinze états possibles, faisant par exemple la différence entre ”*Sleep Light*” et ”*Sleep Deep*”. Le groupe de travail EMAN, auteur de ce RFC, a produit sa propre liste, qui comprend douze états (section 6.5.4 du RFC), chacun numéroté (pour faciliter l'écriture des futures MIB, dont plusieurs sont proches de la publication en RFC). Par exemple, en cas de suspension, une machine est en `sleep(3)` et `high(11)` est au contraire l'état d'une machine complètement réveillée et qui ne cherche pas à faire des économies d'électricité. Un tableau en section 6.5.5 donne des équivalences entre ces normes. Par exemple, le `sleep(3)` d'EMAN correspond à l'état dormant de IEEE 1621, au ”G1, S3” d'ACPI, et au ”*Sleep Deep*” de DMTF. Ces états sont stockés dans un registre IANA `<https://www.iana.org/assignments/`

power-state-sets/power-state-sets.xml> (section 12 du RFC) et cette liste pourra être modifiée par la suite.

Une description formelle du modèle, en UML, figure en section 7.

Pour connaître l'état des mises en œuvre de ce cadre de gestion de l'énergie, voir le Wiki du groupe de travail <http://tools.ietf.org/wg/eman/trac/wiki/EmanImplementations> (plusieurs MIB ont déjà été définies selon ce cadre).

Maintenant, la sécurité (section 11). Sérieux sujet dès qu'on touche à un service aussi essentiel que l'alimentation électrique. Va-t-on éteindre toute l'électricité d'une ville avec son téléphone, comme dans Watch Dogs ? Non, c'est plus compliqué que cela <http://www.slate.fr/story/89411/jeu-video-watch-dogs- Néanmoins, la gestion d'énergie doit être mise en œuvre prudemment. Si on gère les machines avec SNMP, le RFC 3410 sur la sécurité de SNMP est une lecture indispensable. Les accès en écriture, permettant de modifier l'état d'unee machine, peuvent en effet permettre, s'ils ne sont pas bien sécurisés, d'éteindre un appareil critique à distance. Mais même les accès en lecture seule peuvent être dangereux, car révélant des choses qu'on aurait préféré garder pour soi.

On lire aussi avec intérêt le RFC 7603 qui précise le cadre d'utilisation.