RFC 7787 : Distributed Node Consensus Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 avril 2016

Date de publication du RFC : Avril 2016

https://www.bortzmeyer.org/7787.html

Ce nouveau protocole DNCP ("*Distributed Node Consensus Protocol*") est prévu notamment pour les réseaux non organisés comportant beaucoup de petites machines, comme on peut en trouver en domotique (Internet des Trucs et tout ça). Il s'appuie sur l'algorithme Trickle (RFC 6206 [1]) pour distribuer dans le réseau des informations quelconques. Aucune machine ne joue un rôle particulier dans DNCP : pas de maître ou de racine. DNCP est un algorithme générique, et son utilisation effective nécessite la définition d'un profil, comme HNCP ("*Home Networking Control Protocol*", RFC 7788) dans le cas du réseau « Homenet ».

L'information distribuée par DNCP est sous forme de TLV. Chaque tuple TLV fait 64 ko au maximum et, une fois que l'algorithme a convergé, tous les nœuds du réseau ont la même base de tuples : le réseau est synchronisé. L'état d'un nœud (l'ensemble des tuples qu'il connait) est représenté par un condensat des données (nommé "*network state hash*"). Au début, chaque nœud ne connait que ses propres tuples et son état reflète cela. Il diffuse cette information avec Trickle et chaque nœud apprend alors que la synchro n'est pas complète (puisqu'il existe des nœuds avec un condensat différent). Puis chaque nœud modifie l'état (le condensat) en intégrant les tuples des autres nœuds. À la fin, tous les nœuds ont le même état et trouvent donc le même condensat. Ils sont alors heureux et synchronisés. En situation stable, DNCP ne transmet que les condensats (pour que les nœuds puissent vérifier qu'il n'y a pas eu de changement), pas les données, ce qui le rend assez frugal.

DNCP est donc utile pour tous les cas complètement répartis, où des machines sans chef veulent se coordonner. Cela peut servir, par exemple, pour l'affectation automatique des préfixes IP dans un réseau non géré (RFC 7695). DNCP est à l'origine issu d'un projet moins général, le projet « Homenet <https://tools.ietf.org/wg/homenet> » de l'IETF ; le groupe Homenet s'est aperçu que le protocole de synchronisation pouvait avoir son utilité en dehors de la domotique et a créé DNCP (et un protocole concret pour Homenet, HNCP).

Comme indiqué plus haut, DNCP est abstrait : plusieurs choix techniques essentiels sont laissés à de futurs profils (comme HNCP, décrit dans le RFC 7788). Parmi ces choix :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6206.txt

— le mécanisme de transport (TCP, UDP, SCTP),
— la sécurité (utiliser TLS ou pas),
— le fait de ne fonctionner que sur un seul lien réseau, ou bien d'être capable de passer par des routeurs.

DNCP est très chouette mais le RFC précise qu'il ne peut pas servir à tout. Quelques exemples de cas où DNCP n'est pas adapté :

— Si on a des données de grande taille (DNCP les limite à 64 ko),
— Si on a beaucoup de voisins, et que la diffusion n'est pas possible,
— Si les données changent tout le temps (DNCP est optimisé pour être économe dans l'état stable),
— Si beaucoup de nœuds ont des contraintes de mémoire (DNCP nécessite de stocker toutes les données).

Si on a des données qui changent très souvent, il peut être préférable d'utiliser DNCP pour publier l'adresse d'un serveur où récupérer ces données.

DNCP est résumé en section 3 du RFC. Les nœuds sont découverts automatiquement (la méthode exacte dépend du profil) ou manuellement, et leur joignabilité est vérifiée. Les TLV peuvent être des demandes d'information ou bien les données elles-mêmes.

Chaque nœud commence par calculer le condensat des données (la fonction utilisée depend du profil) qu'il veut publier. Il l'annonce ensuite à ses voisins. Quand un voisin annonce un condensat différent, le nœud sait que le réseau n'est pas encore synchronisé. Il demande alors au voisin toutes ses données, les ajoute à sa base, met à jour son condensat et recommence.

Pour jouer son rôle, le nœud DNCP a besoin de quelques informations (section 5). Il a un identificateur unique (la façon doit il est choisi dépend du profil), ses données à publier, une ou plusieurs instances Trickle (le terme ne vient pas du RFC 6206 mais est défini ici : c'est un état Trickle autonome, avec les valeurs des paramètres Trickle), etc. Le nœud a également en mémoire un ensemble de voisins avec, pour chacun, son identificateur, son adresse IP, etc.

Les TLV et leur format sont décrits en section 7 : un type de deux octets, une longueur de deux octets et la valeur. Des TLV peuvent être inclus dans les TLV. Quelques exemples de types : `REQ-NETWORK-STATE` (type 1) sert à demander l'état du réseau (le condensat de tous les tuples), `NETWORK-STATE` (type 4) sert pour les réponses à ces requêtes et donne l'état du réseau, etc. L'ensemble des valeurs possibles figure dans un registre IANA `<https://www.iana.org/assignments/dncp-registry/dncp-registry.xml#tlv-types>`.

La section 8 discute de la sécurité de DNCP. Le traitement de celle-ci dépend essentiellement du profil, qui peut décider, par exemple, d'utiliser TLS ou DTLS.

La section 9 discute en détail ce qu'on attend des profils, ce que doivent définir les protocoles concrets qui réalisent DNCP :

— S'ils utilisent l'"*unicast*" ou le "*multicast*".
— Le protocole de transport, UDP ou TCP.
— S'ils utilisent TLS pour la sécurité.
— Quels TLV sont acceptés ou refusés.
— Quelles valeurs ont les paramètres Trickle.
— Quelle est la fonction de condensation.
— Etc.

Des indications sur les choix à faire sont présentées dans l'annexe B. Par exemple, si on utilise UDP, il est recommandé que les données restent d'une taille inférieure à une MTU typique, notamment pour éviter la fragmentation. D'un autre côté, UDP donne davantage de contrôle sur l'envoi des données (mais pas forcément sur leur réception).

Un exemple de profil figure dans l'annexe A. On peut aussi regarder un vrai profil, dans le RFC 7788, qui normalise HNCP. Le profil d'exemple est pour un protocole fictif nommé SHSP (il a existé un projet de protocole nommé "*Simple Home Status Protocol*" mais qui semble abandonné, et sans lien avec ce SHSP fictif), un protocole pour l'automatisation de la maison. Ses choix sont :

— Uniquement IPv6,
— "*Unicast*" en TCP et "*multicast*" en UDP,
— Zéro sécurité (ne doit être utilisé que sur des liens sûrs),
— L'identificateur d'un nœud est un nombre de 32 bits choisi aléatoirement (en cas de collision, les deux machines en choisissent un autre),
— Les paramètres quantitatifs de Trickle font que, dans l'état stable, au moins un paquet "*multicast*" est émis toutes les 25 secondes,
— La fonction de condensation est une classique SHA-256,
— Etc.