Groupe de travail Réseau
Request for Comments : 1542
Met à jour la RFC 951
Rend obsolète la RFC 1532
Catégorie : Sur la voie de la normalisation

W. Wimer, Carnegie Mellon University

octobre 1993

Traduction Claude Brière de L'Isle

Éclaircissements et extensions au protocole Bootstrap

Statut de ce mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Certains aspects du protocole BOOTP ont été définis de façon assez lâche dans la spécification d'origine. En particulier, une description générale a seulement été fournie pour le comportement des "agents de relais BOOTP" (à l'origine appelés "agents de transmission BOOTP"). La description du comportement du client a aussi été négligée d'une certaine façon. Le présent mémoire tente de préciser et renforcer la spécification dans ces domaines. Du fait de certaines erreurs introduites dans la RFC 1532 au cours de l'édition, le présent mémoire remplace la RFC 1532.

De plus, de nouveaux problèmes sont apparus depuis la rédaction de la spécification d'origine. Le présent mémoire tente aussi de traiter certains d'entre eux.

Table des Matières

1. Introduction

Le protocole Bootstrap (BOOTP) se fonde sur UDP/IP qui permet à un hôte qui s'amorce de se configurer de façon dynamique et sans supervision de l'utilisateur. BOOTP donne le moyen de notifier à un hôte son adresse IP allouée, l'adresse IP d'un serveur d'amorçage, et le nom d'un fichier à charger en mémoire et à exécuter [RFC0951]. D'autres informations de configuration comme le gabarit de sous réseau local, le décalage de l'heure locale, les adresses des routeurs par défaut, et les adresses de divers serveurs Internet peuvent aussi être communiquées à un hôte avec BOOTP [RFC1497].

Malheureusement, la spécification BOOTP d'origine [RFC0951] a laissé certains problèmes du protocole ouverts à la discussion. Le comportement exact des agents de relais BOOTP anciennement appelés "agents de transmission BOOTP") n'a pas été clairement spécifié. Certaines parties de la spécification globale du protocole sont en fait en conflit, tandis que d'autres parties ont été mal interprétées, ce qui indique que des éclaircissements sont nécessaires. Le présent mémoire aborde ces problèmes.

Depuis l'introduction de BOOTP, le réseau en anneau à jetons IEEE 802.5 a été développé qui présente un problème unique pour le paradigme particulier de transfert de message de BOOTP. Ce mémoire suggère aussi une solution à ce problème.

Note : sauf spécification contraire de ce document ou de documents ultérieurs, les informations et exigences spécifiées ici s'appliquent aux extensions à BOOTP comme au protocole de configuration dynamique d'hôte (DHCP) [RFC1541].

1.1 Exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

1.2 Terminologie

Le présent mémoire utilise les termes suivants :

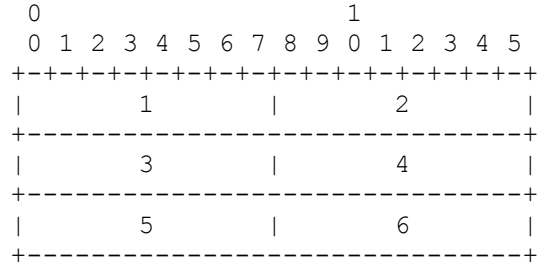
BOOTREQUEST : un message BOOTREQUEST est un message BOOTP envoyé d'un client BOOTP à un serveur BOOTP, qui demande des informations de configuration.

BOOTREPLY : un message BOOTREPLY est un message BOOTP envoyé d'un serveur BOOTP à un client BOOTP, qui fournit des informations de configuration.

éliminer en silence : le présent mémoire spécifie plusieurs cas où une entité BOOTP va "éliminer en silence" un message BOOTP reçu. Cela signifie que l'entité va éliminer le message sans autre traitement, et que l'entité ne va ensuite envoyer aucun messages d'erreur ICMP. Cependant, pour le diagnostic des problèmes, l'entité DEVRAIT fournir la capacité d'enregistrer l'erreur, incluant le contenu du message éliminé en silence, et DEVRAIT enregistrer l'événement dans un compteur de statistiques.

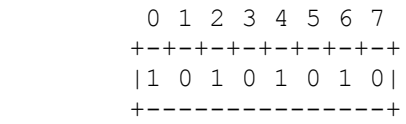
1.3 Ordre de transmission des données

L'ordre de transmission de l'en-tête et des données décrit dans le présent document est résolu au niveau de l'octet. Chaque fois qu'un diagramme montre un groupe d'octets, l'ordre de transmission de ces octets est l'ordre normal dans lequel ils sont lus en français. Par exemple, dans le diagramme suivant, les octets sont transmis dans l'ordre de leurs numéros.



Chaque fois qu'un octet représente une quantité numérique, le bit le plus à gauche dans le diagramme est le bit de poids fort

(MSB, *most significant bit*). C'est-à-dire que le bit marqué 0 est le bit de poids fort. Par exemple, le diagramme qui suit représente la valeur 170 (décimal).



De même, chaque fois qu'un champ de plusieurs octets représente une quantité numérique, le bit de gauche du champ est le bit de poids fort. Quand une quantité multi octets est transmise, l'octet de poids fort est transmis en premier.

2. Problèmes généraux

La présente section couvre les questions générales sur toutes les entités BOOTP (clients, serveurs, et agents de relais).

2.1 Traitement général de BOOTP

Les vérifications de cohérence suivantes DEVRAIENT être effectuées sur les messages BOOTP :

o La longueur totale IP et la longueur UDP DOIVENT être suffisantes pour contenir les 300 octets minimum de l'en-tête BOOTP (dans le champ Données UDP) spécifiés dans la [RFC0951].

Note : de futures extensions du protocole BOOTP PEUVEUT augmenter la taille des messages BOOTP. Donc, les messages BOOTP qui, conformément aux champs Longueur totale IP et Longueur UDP, sont supérieurs à la taille minimum spécifiée par la [RFC0951] DOIVENT aussi être acceptés.

o Le champ "op" (opcode) du message DOIT contenir le code (1) pour BOOTREQUEST ou le code (2) pour BOOTREPLY.

Les messages BOOTP qui ne passent pas ces vérifications de cohérence DOIVENT être éliminés en silence.

2.2 Définition du champ "fanions"

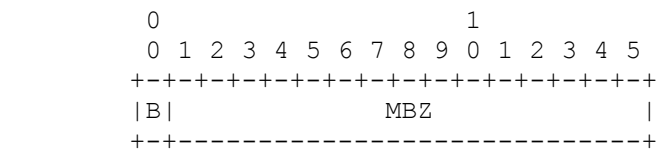
Le format standard du message BOOTP défini dans la [RFC0951] inclut un champ de deux octets situé entre le champ "secs" et le champ "ciaddr". Ce champ est simplement désigné comme "non utilisé" et son contenu est laissé non spécifié, bien que le paragraphe 7.1 de la [RFC0951] fasse la suggestion suivante : "Avant d'établir le paquet pour la première fois, c'est une bonne idée de mettre la mémoire tampon de paquet toute entière à zéro ; cela va placer tous les champs dans leur état par défaut."

Le présent mémoire désigne ce champ de deux octets comme le champ "fanions".

Le présent mémoire définit le bit de poids fort du champ "fanions" comme étant le fanion BROADCAST (B) *(diffusion)*. La sémantique de ce fanion est discutée aux paragraphes 3.1.1 et 4.1.2 du présent mémoire.

Les bits restants du champ "fanions" sont réservés pour une utilisation future. Ils DOIVENT être réglés à zéro par les clients et ignorés par les serveurs et agents de relais.

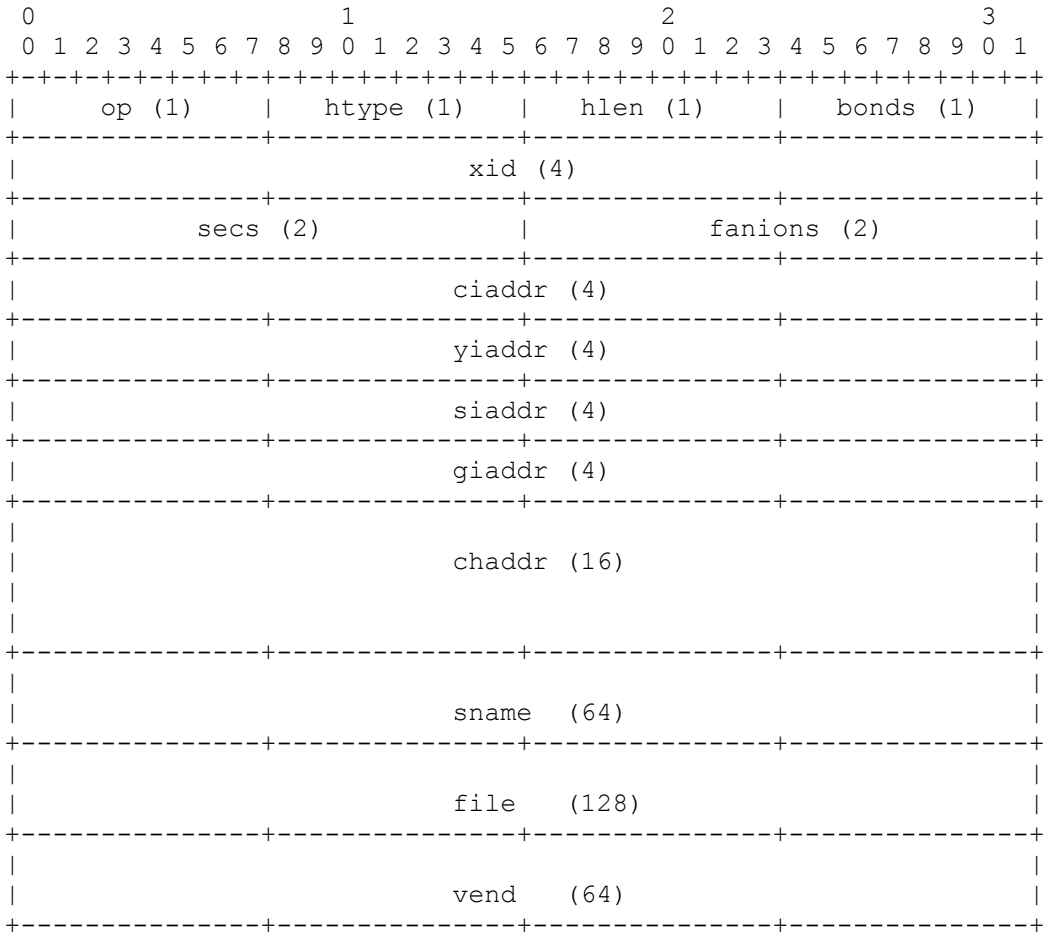
Le champ "fanions" apparaît alors comme suit :



où :
B : fanion BROADCAST (voir aux paragraphes 3.1.1 et 4.1.2)
MBZ : DOIT être à zéro (réservé pour utilisation future)

Le format d'un message BOOTP est montré ci-dessous. Les nombres entre parenthèses indiquent la taille de chaque champ

en octets.



2.3 Ordre des bits des adresses matériel

L'ordre des bits utilisé pour les adresses de matériel de niveau liaison dans le champ "chaddr" DEVRAIT être le même que celui utilisé pour le protocole ARP [RFC0826] sur le réseau niveau liaison du client (en supposant qu'ARP est défini pour ce réseau).

Le champ "chaddr" DOIT être préservé comme il a été spécifié par le client BOOTP. Un agent de relais NE DOIT PAS inverser l'ordre des bits du champ "chaddr" même si il se trouve relayer une BOOTREQUEST entre deux réseaux qui utilisent un ordre binaire différent.

Discussion : Une des principales raisons de l'existence du champ "chaddr" est de permettre aux serveurs et agents de relais BOOTP de communiquer directement avec les clients sans utiliser de diffusions. En pratique, le contenu du champ "chaddr" est souvent utilisé pour créer une entrée d'antémémoire ARP exactement de la même façon que l'aurait fait le protocole ARP normal. En clair, l'interopérabilité ne peut être réalisée que si une interprétation cohérente du champ "chaddr" est utilisée. À titre d'exemple pratique, cela signifie que l'ordre des bits utilisé pour le champ "chaddr" par un client BOOTP sur un réseau à anneau à jetons IEEE 802.5 est l'opposé de l'ordre des bits utilisé par un client BOOTP sur un réseau DIX ethernet.

2.4 BOOTP sur réseaux à anneau à jetons IEEE 802.5

Une considération particulière des interactions client/serveur et client/agent de relais DOIT être apportée au réseau IEEE 802.5 à cause du pontage non transparent.

Le client DEVRAIT envoyer sa diffusion de BOOTREQUEST avec un champ d'informations d'acheminement (RIF, *Routing Information Field*) explorateur de tous chemins. Cela va permettre aux serveurs/agents de relais de mettre en antémémoire le chemin de retour si ils choisissent de le faire. Pour les serveurs/agents de relais qui ne peuvent pas mettre en antémémoire le chemin de retour (parce qu'ils sont sans états, par exemple) le message BOOTREPLY DEVRAIT être envoyé à l'adresse de matériel du client, telle que prise dans le message BOOTP, avec un RIF enraciné dans l'arborescence

d'expansion. Le chemin du pont réel sera enregistré par le client et le serveur/agent de relais par le code normal de traitement ARP.

Discussion : Dans le cas le plus simple, un réseau non ponté à un seul anneau, le comportement de diffusion du protocole BOOTP est identique à celui des réseaux Ethernet. Cependant, a client BOOTP ne peut pas savoir, a priori, qu'un réseau 805.2 n'est pas ponté. En fait, la probabilité est que le serveur, ou agent de relais, ne le saura pas non plus. Des quatre scénarios possibles, seuls deux sont intéressants : où l'hypothèse est que le réseau 805.2 est non ponté alors qu'il l'est, et où l'hypothèse est que le réseau est ponté alors qu'il ne l'est pas. Dans le premier cas, le RIF ne sera pas utilisé, donc, si le serveur/agent de relais est sur un autre segment de l'anneau, le client ne peut pas l'atteindre. Dans le dernier cas, le RIF sera utilisé, résultant en quelques octets étrangers sur l'anneau. Il est évident qu'une inefficacité presque imperceptible est à préférer à un échec complet de communication. Étant donné que l'hypothèse est que les RIF seront nécessaires, il faut déterminer la méthode optimale pour que le client atteigne le serveur/agent de relais, et la méthode optimale pour que la réponse soit retournée.

3. Comportement du client BOOTP

Cette section précise divers problèmes concernant le comportement du client BOOTP.

3.1 Utilisation par le client du champ "Fanions"

3.1.1 Fanion BROADCAST

Normalement, les serveurs et agents de relais BOOTP tentent de livrer les messages BOOTREPLY directement au client en utilisant la livraison en envoi individuel. L'adresse IP de destination (dans l'en-tête IP) est réglée à l'adresse BOOTP "yiaddr" et l'adresse de destination de couche liaison est réglé à l'adresse BOOTP "chaddr". Malheureusement, certaines mises en œuvre de client sont incapables de recevoir de tels datagrammes IP en envoi individuel tant qu'elles ne connaissent pas leur propre adresse IP (on a donc un problème de "la poule et l'œuf"). Souvent, cependant, elles peuvent recevoir des datagrammes IP en diffusion (ceux qui ont une adresse de diffusion IP valide comme destination IP et l'adresse de diffusion de couche liaison comme destination de couche liaison).

Si un client entre dans cette catégorie, il DEVRAIT établir (à 1) le fanion nouvellement défini BROADCAST dans le champ "fanions" des messages BOOTREQUEST qu'il génère. Cela va donner une indication aux serveurs et agents de relais BOOTP qu'ils DEVRAIENT tenter de diffuser leurs messages BOOTREPLY au client.

Si un client n'a pas cette limitation (c'est-à-dire, si il est parfaitement capable de recevoir les messages BOOTREPLY en envoi individuel) il NE DEVRAIT PAS établir le fanion BROADCAST (c'est-à-dire, il DEVRAIT mettre le fanion BROADCAST à 0).

Discussion : cet ajout au protocole est une astuce pour les vieilles mises en œuvre d'hôtes. De telles mises en œuvre DEVRAIENT être modifiées afin qu'elles puissent recevoir des messages BOOTREPLY en envoi individuel, rendant donc inutile l'utilisation de cette astuce. En général, l'utilisation de ce mécanisme est déconseillée.

3.1.2 Reste du champ "Fanions"

Les bits restants du champ "fanions" sont réservés pour une utilisation future. Un client DOIT régler ces bits à zéro dans tous les messages BOOTREQUEST qu'il génère. Un client DOIT ignorer ces bits dans tous les messages BOOTREPLY qu'il reçoit.

3.2 Définition du champ "secs"

Le champ "secs" d'un message BOOTREQUEST DEVRAIT représenter le temps écoulé, en secondes, depuis que le client a envoyé son premier message BOOTREQUEST. Noter que cela implique que le champ "secs" du premier message BOOTREQUEST DEVRAIT être réglé à zéro.

Les clients NE DEVRAIENT PAS régler le champ "secs" à une valeur constante pour tous les messages BOOTREQUEST.

Discussion : La définition originale du champ "secs" était vague. Il n'était pas clair si il représentait le temps écoulé depuis l'envoi du premier message BOOTREQUEST ou quelque autre période de temps comme celui depuis la mise sous tension de la machine client. Cela a limité son utilité comme mécanisme de contrôle de politique pour les serveurs

et agents de relais BOOTP. De plus, certaines mises en œuvre de client sont connues pour régler ce champ à une valeur constante ou utiliser un ordre incorrect des octets. Un ordre incorrect des octets le fait ordinairement apparaître comme si un client avait attendu plus longtemps qu'il ne l'a fait en réalité, de sorte qu'un agent de relais va relayer la BOOTREQUEST plus tôt que désiré (généralement immédiatement). Ces erreurs de mis en œuvre ont de plus nuit à l'utilité du champ "secs". Ces mises en œuvre incorrectes DEVRAIENT être corrigées.

3.3 Utilisation des champs "ciaddr" et "yiaddr"

Si un client BOOTP ne sait pas quelle adresse IP il DEVRAIT utiliser, ce client DEVRAIT régler le champ "ciaddr" à 0.0.0.0. Si le client a la capacité de se souvenir de la dernière adresse IP qui lui a été allouée, ou si il a été préconfiguré avec une adresse IP via quelque autre mécanisme, le client PEUT remplir le champ "ciaddr" avec cette adresse IP. Si le client place une adresse IP non à zéro dans le champ "ciaddr", le client DOIT être prêt à accepter les datagrammes entrants en envoi individuel adressés à cette adresse IP et aussi à répondre aux demandes ARP pour cette adresse IP (si ARP est utilisé sur ce réseau).

Le serveur BOOTP est libre d'allouer une adresse IP (dans le champ "yiaddr") différente de celle qu'a exprimé le client dans "ciaddr". Le client DEVRAIT adopter l'adresse IP spécifiée dans "yiaddr" et commencer à l'utiliser aussitôt que possible.

Discussion : Il y a diverses interprétations sur l'objet du champ "ciaddr" et, malheureusement, aucun accord sur une seule interprétation correcte. Une interprétation est que si un client veut accepter toute adresse IP que le serveur BOOTP lui alloue, le client DEVRAIT toujours placer 0.0.0.0 dans le champ "ciaddr", sans considérer si il connaît ou non l'adresse qui lui a été allouée précédemment. À l'inverse, si le client souhaite affirmer qu'il DOIT avoir une adresse IP particulière (par exemple, l'adresse IP a été configurée manuellement par l'administrateur de l'hôte et BOOTP est seulement utilisé pour obtenir un fichier d'amorçage et/ou des informations provenant du champ "vend") le client va alors remplir le champ "ciaddr" avec l'adresse IP désirée et ignorer l'adresse IP allouée par le serveur BOOTP comme indiquée dans le champ "yiaddr". Une autre interprétation dit que le client remplit toujours le champ "ciaddr" avec son adresse IP la plus récemment allouée (si elle est connue) même si cette adresse PEUT être incorrecte. Un tel client va quand même accepter et utiliser l'adresse allouée par le serveur BOOTP comme indiquée dans le champ "yiaddr". Le motif de cette interprétation est d'aider le serveur à identifier le client et/ou à délivrer la BOOTREPLY au client. Une troisième (mauvaise) interprétation permet au client d'utiliser "ciaddr" pour exprimer l'adresse IP désirée par le client, même si le client n'a jamais utilisé cette adresse auparavant ou n'utilise pas actuellement cette adresse. Cette dernière interprétation est incorrecte car elle PEUT empêcher la BOOTREPLY d'atteindre le client. Le serveur va généralement envoyer la réponse en envoi individuel à l'adresse donnée dans le "ciaddr" mais le client PEUT ne pas écouter encore sur cette adresse, ou le client PEUT être connecté à un sous réseau incorrect tel que l'acheminement IP normal achemine (correctement) la réponse à un sous réseau différent. La seconde interprétation souffre aussi du problème du "sous réseau incorrect". La première interprétation semble être la plus sûre et la plus apte à promouvoir l'interopérabilité.

3.4 Interprétation du champ "giaddr"

Le champ "giaddr" est assez mal nommé. Il existe pour faciliter le transfert des messages BOOTREQUEST provenant d'un client, à travers des agents de relais BOOTP, aux serveurs sur des réseaux différents de celui du client. De même, il facilite la livraison des messages BOOTREPLY des serveurs, à travers des agents de relais BOOTP, en retour au client. Il ne représente en aucun cas un routeur IP général à utiliser par le client. Un client BOOTP DOIT régler le champ "giaddr" à zéro (0.0.0.0) dans tous les messages BOOTREQUEST qu'il génère.

Un client BOOTP NE DOIT PAS interpréter le champ "giaddr" d'un message BOOTREPLY comme étant l'adresse IP d'un routeur >IP. Un client BOOTP DEVRAIT complètement ignorer le contenu du champ "giaddr" dans les messages BOOTREPLY.

Discussion : La sémantique du champ "giaddr" était mal définie. Le paragraphe 7.5 de la [RFC0951] déclare : "Si "giaddr" (adresse de passerelle) est non zéro, alors les paquets DEVRAIENT être transmis là en premier, afin d'arriver au serveur". Dans cette phrase, "arriver à" se réfère à la communication du client au serveur à la suite de l'échange BOOTP, comme à une session TFTP. Malheureusement, le champ "giaddr" PEUT contenir l'adresse d'un agent de relais BOOTP qui n'est pas lui-même un routeur IP (selon la [RFC0951], Section 8, cinquième alinéa) auquel cas, il sera inutile comme premier bond pour les paquets TFTP envoyés au serveur (car, par définition, les non routeurs ne transmettent pas de datagrammes à la couche IP).

Bien que ce soit maintenant interdit par le paragraphe 4.1.1 du présent document, le champ "giaddr" peut contenir une adresse de diffusion conformément à la Section 8, alinéa 6 de la [RFC0951]. Non seulement une telle adresse serait inutile comme adresse de routeur, mais elle pourrait aussi faire que le client lance ARP pour l'adresse de diffusion (car, si le client

n'a pas reçu un gabarit de sous réseau dans le message BOOTREPLY, il sera incapable de reconnaître une adresse de diffusion de sous réseau). Ceci est clairement indésirable.

Pour atteindre un serveur non local, les clients peuvent obtenir une adresse de routeur de premier bond du sous champ "Gateway" des "extensions d'informations de fabricant" [RFC1497] (si il est présent) ou via le protocole de découverte de routeur ICMP [RFC1256] ou autre mécanisme similaire.

3.5 Information de fabricant "cookie magique"

Il est RECOMMANDÉ qu'un client BOOTP remplisse toujours les quatre premiers octets du champ "vend" (informations de fabricant) d'une BOOTREQUEST avec un identifiant de quatre octets appelé un "cookie magique". Un client BOOTP DEVRAIT faire cela même si il n'a pas d'information particulière à communiquer au serveur BOOTP en utilisant le champ "vend"'. Cela aide le serveur BOOTP à déterminer quel format d'informations de fabricant il DEVRAIT utiliser dans ses messages BOOTREPLY.

Si il n'utilise pas de cookie magique spécifique d'un fabricant particulier, un client BOOTP DEVRAIT utiliser la valeur décimale séparée par des points de 99.130.83.99 comme spécifié dans la [RFC1497]. Dans ce cas, si le client n'a pas d'informations à communiquer au serveur, l'octet qui suit immédiatement le cookie magique DEVRAIT être réglé à l'étiquette "Fin" (255) et les octets restants du champ "vend" DEVRAIENT être mis à zéro.

Discussion : Parfois des systèmes d'exploitation ou des paquetages de réseautage différents sont utilisés sur la même machine à des moments différents (ou même en même temps !). Comme l'adresse de matériel placée dans le champ "chaddr" sera probablement la même, les BOOTREQUEST provenant de clients BOOTP complètement différents sur la même machine seront probablement difficiles à différencier pour un serveur BOOTP. Si le client inclut un cookie magique dans ses BOOTREQUEST, le serveur va au moins savoir à quel format s'attend et peut comprendre le client dans les messages BOOTREPLY correspondants.

4. Agents de relais BOOTP

Dans de nombreux cas, les clients BOOTP et leurs serveurs BOOTP associés ne résident pas sur le même réseau ou sous réseau IP. Dans de tels cas, une sorte d'agent tiers est requis pour transférer les messages BOOTP entre clients et serveurs. Un tel agent était à l'origine appelé un "agent de transmission BOOTP". Cependant, afin d'éviter la confusion avec la fonction de transmission IP d'un routeur IP, le nom de "agent de relais BOOTP" est adopté ici à la place.

Discussion : Un agent de relais BOOTP effectue une tâche qui est distincte de la fonction normale de transmission IP d'un routeur IP. Alors qu'un routeur commute normalement les datagrammes IP entre les réseaux de façon plus ou moins transparente, un agent de relais BOOTP PEUT être vu de façon plus appropriée comme recevant les messages BOOTP comme destination finale et ensuite génère les nouveaux messages BOOTP qui en résultent. Il est incorrect qu'une mise en œuvre d'agent de relais transmette simplement un message BOOTP "directement comme un paquet normal". Cet fonction d'agent de relais est plus convenablement située dans les routeurs qui interconnectent les clients et les serveurs, mais PEUVENT aussi être localisés dans un hôte qui est directement connecté au sous réseau du client.

Tout hôte ou routeur Internet qui fournit une capacité d'agent de relais BOOTP DOIT se conformer aux spécifications du présent mémoire.

4.1 Traitement BOOTP général pour les agents de relais

Tous les messages UDP livrés en local dont le numéro d'accès de destination UDP est BOOTPS (67) sont considérés comme susceptibles d'un traitement particulier par l'agent de relais BOOTP logique de l'hôte ou routeur.

Dans le cas d'un hôte, les datagrammes livrés en local sont simplement tous les datagrammes reçus normalement par cet hôte, c'est-à-dire, les datagrammes en diffusion et en diffusion groupée aussi bien que ceux en envoi individuel envoyés aux adresses IP de cet hôte.

Dans le cas d'un routeur, les datagrammes livrés en local sont en diffusion et en diffusion groupée ainsi qu'en envoi individuel envoyés aux adresses IP de ce routeur. Ce sont des datagrammes pour lesquels le routeur DEVRAIT être considéré comme destination finale par opposition à un mode de commutation intermédiaire. Donc un datagramme en envoi individuel avec une destination IP qui ne correspond à aucune des adresses IP du routeur n'est pas prise en compte pour le traitement par l'agent de relais logique BOOTP du routeur.

Les hôtes et routeurs sont généralement obligés d'éliminer en silence les datagrammes entrants qui contiennent des adresses IP de source illégales. C'est ce qu'on appelle généralement le "filtrage des adresses martiennes". Une de ces adresses illégales est 0.0.0.0 (ou en fait tout ce qui est sur le réseau 0). Cependant, les hôtes ou routeurs qui prennent en charge un agent de relais BOOTP DOIVENT accepter la livraison à l'agent de relais des messages BOOTREQUEST dont l'adresse IP de source est 0.0.0.0. Les messages BOOTREQUEST provenant d'adresses IP de source légales DOIVENT aussi être acceptés.

Un agent de relais DOIT éliminer en silence tout message UDP reçu dont le numéro d'accès de destination UDP est BOOTPC (68).

Discussion : Il NE DEVRAIT PAS être nécessaire qu'un agent de relais traite les messages adressés à l'accès BOOTPC. Une lecture attentive de la spécification BOOTP originale [RFC0951] le montre. Néanmoins, certaines mises en œuvre d'agent de relais relaient de tels messages de façon incorrecte.

Les vérifications de cohérence du paragraphe 2.1 DEVRAIENT être effectuées par l'agent de relais. Les messages BOOTP qui ne satisfont pas à ces vérifications de cohérence DOIVENT être éliminés en silence.

4.1.1 Messages BOOTREQUEST

Un mécanisme de configuration DOIT exister pour activer ou désactiver le relais des messages BOOTREQUEST. Le relais DOIT être désactivé par défaut.

Lorsque l'agent de relais BOOTP reçoit un message BOOTREQUEST, il PEUT utiliser la valeur du champ "secs" (secondes depuis l'amorçage du client) de la demande comme facteur de décision de relayer la demande. Si un tel mécanisme de politique est mis en œuvre, son seuil DEVRAIT être configurable.

Discussion : aujourd'hui, cette caractéristique du protocole BOOTP ne s'est pas nécessairement montrée utile. Voir une discussion au paragraphe 3.2.

L'agent de relais DOIT éliminer en silence les messages BOOTREQUEST dont le champ "bonds" excède la valeur 16. Une option de configuration DEVRAIT être fournie pour régler de seuil à une valeur inférieure si désiré par le gestionnaire du réseau. Le réglage par défaut d'un seuil configurable DEVRAIT être 4.

Si l'agent de relais décide de relayer la demande, il DOIT examiner le champ "giaddr" (adresse IP de "passerelle"). Si ce champ est zéro, l'agent de relais DOIT remplir ce champ avec l'adresse IP de l'interface sur laquelle la demande a été reçue. Si l'interface a plus d'une adresse IP qui lui est logiquement associée, l'agent de relais DEVRAIT choisir une adresse IP associée à cette interface et l'utiliser de façon cohérente pour tous les messages BOOTP qu'il relaye. Si le champ "giaddr" contient une valeur non zéro, le champ "giaddr" NE DOIT PAS être modifié. L'agent de relais NE DOIT en aucune circonstance remplir le champ "giaddr" avec une adresse de diffusion comme suggéré dans la [RFC0951] (Section 8, sixième paragraphe).

La valeur du champ "bonds" DOIT être incrémentée.

Tous les autres champs BOOTP DOIVENT rester intacts.

À ce point, la demande est relayée à sa ou ses nouvelles destinations. Cette destination DOIT être configurable. De plus, cette configuration de destination DEVRAIT être indépendante de la configuration de destination pour tout autre "transmetteur de diffusion" (par exemple, pour les protocoles TFTP fondés sur UDP, DNS, Heure du réseau, etc.).

Discussion : Le gestionnaire de réseau PEUT souhaiter que la destination relayeuse soit en envoi individuel IP, en diffusion groupée, en diffusion, ou leur combinaison. Une liste configurable des adresses IP de destination donne une bonne souplesse. Les schémas de configuration les plus flexibles sont encouragés. Par exemple, il PEUT être souhaitable d'envoyer à l'adresse de diffusion limitée (255.255.255.255) sur des interfaces physiques spécifiques. Cependant, si le message BOOTREQUEST a été reçu comme diffusion, l'agent de relais NE DOIT PAS rediffuser la BOOTREQUEST sur l'interface physique de laquelle il est venu.

Un agent de relais DOIT utiliser la même destination (ou ensembles de destinations) pour tous les messages BOOTREQUEST qu'il relaye d'un certain client.

Discussion : Au moins une mise en œuvre connue d'agent de relais utilise un schéma round-robin pour assurer l'équilibrage de charge à travers les divers serveurs BOOTP. Chaque fois qu'il reçoit un nouveau message BOOTREQUEST, il

relaye le message au prochain serveur BOOTP de sa liste des serveurs. Donc, avec cet agent de relais, plusieurs messages BOOTREQUEST consécutifs d'un certain client seront livrés à des serveurs différents.

Malheureusement ce schéma bien intentionné réagit mal avec DHCP [RFC1541] et peut-être d'autres variations du protocole BOOTP qui dépendent de multiples échanges de messages BOOTREQUEST et BOOTREPLY entre clients et serveurs. Donc, tous les messages BOOTREQUEST provenant d'un client donné DOIVENT être relayés à la même destination (ou ensemble de destinations).

Une façon de satisfaire cette exigence tout en apportant un avantage d'équilibrage de charge est de hacher l'adresse de couche liaison du client (ou quelque autre information fiable d'identification du client) et d'utiliser la valeur du hachage résultant pour choisir la destination (ou ensemble de destinations) de relais appropriée. La solution la plus simple, bien sûr, est de ne pas utiliser de schéma d'équilibrage de charge et de juste relayer TOUS les messages BOOTREQUEST reçus à la même destination (ou ensemble de destinations).

Lorsque il transmet la demande à sa prochaine destination, l'agent de relais PEUT régler le champ IP Durée de vie (TTL) à la valeur par défaut pour les nouveaux datagrammes générés par l'agent de relais, ou au TTL de la BOOTREQUEST d'origine décrémenté de (au moins) un.

Discussion : Comme précaution supplémentaire contre les boucles de BOOTREQUEST, il est préférable d'utiliser le TTL décrémenté provenant de la BOOTREQUEST d'origine. Malheureusement, cela PEUT être difficile à faire dans certaines mises en œuvres.

Si la BOOTREQUEST a une somme de contrôle UDP (c'est-à-dire, si la somme de contrôle UDP n'est pas à zéro) elle DOIT être recalculée avant de transmettre la demande.

4.1.2 Messages BOOTREPLY

Les agents de relais BOOTP ne relayent les messages BOOTREPLY qu'aux clients BOOTP. Il est de la responsabilité des serveurs BOOTP d'envoyer les messages BOOTREPLY directement à l'agent de relais identifié dans le champ "giaddr". Donc, un agent de relais PEUT supposer que tous les messages BOOTREPLY qu'il reçoit sont destinés aux clients BOOTP sur ses réseaux directement connectés.

Lorsque un agent de relais reçoit un message BOOTREPLY, il DEVRAIT examiner les champs BOOTP "giaddr", "yiaddr", "chaddr", "htype", et "hlen". Ces champs DEVRAIENT fournir des informations adéquates pour que l'agent de relais livre le message BOOTREPLY au client.

Le champ "giaddr" peut être utilisé pour identifier l'interface logique à partir de laquelle la réponse DOIT être envoyée (c'est-à-dire, l'interface d'hôte ou de routeur connectée au même réseau que le client BOOTP). Si le contenu du champ "giaddr" ne correspond pas à une des interfaces logiques directement connectées de l'agent de relais, le message BOOTREPLY DOIT être éliminé en silence.

Les champs "htype", "hlen", et "chaddr" fournissent le type de matériel de couche liaison, la longueur de l'adresse de matériel, et l'adresse de matériel du client comme défini dans le protocole ARP [RFC0826] et le document des numéros alloués [RFC1340]. Le champ "yiaddr" est l'adresse IP du client, comme allouée par le serveur BOOTP.

L'agent de relais DEVRAIT examiner le fanion BROADCAST nouvellement défini (voir plus d'informations aux paragraphes 2.2 et 3.1.1). Si ce fanion est réglé à 1, la réponse DEVRAIT être envoyée comme diffusion IP en utilisant l'adresse IP de diffusion limitée de 255.255.255.255 comme adresse IP de destination et l'adresse de diffusion de couche liaison comme adresse de destination de couche liaison. Si le fanion BROADCAST est à zéro, la réponse DEVRAIT être envoyée en envoi individuel IP à l'adresse IP spécifiée par le champ "yiaddr" et l'adresse de couche liaison spécifiée dans le champ "chaddr". Si l'envoi individuel n'est pas possible, la réponse PEUT être envoyée comme diffusion, auquel cas elle DEVRAIT être envoyée à l'adresse de diffusion de couche liaison en utilisant l'adresse de diffusion IP limitée de 255.255.255.255 comme adresse IP de destination.

Discussion : l'ajout du fanion BROADCAST au protocole est une astuce pour aider à promouvoir l'interopérabilité avec certaines mises en œuvre de client.

Noter que comme le champ "fanions" était précédemment défini dans la [RFC0951] simplement comme un champ "non utilisé", il est possible que de vieilles mises en œuvre de client ou de serveur puissent accidentellement et involontairement établir le nouveau fanion BROADCAST. On pense qu'en fait de telles mises en œuvre seront rares (la plupart des mises en œuvre semblent mettre ce champ à zéro) mais des interactions avec de telles mises en œuvre DOIVENT néanmoins être envisagées. Si un vieux client ou serveur établit le fanion BROADCAST à 1 par erreur, les agents de relais conformes vont

générer des messages BOOTREPLY en diffusion au client correspondant. Les messages BOOTREPLY DEVRAIENT quand même atteindre correctement le client, au prix d'une diffusion supplémentaire (par ailleurs inutile). Ceci n'est cependant pas pire qu'un serveur ou agent de relais qui diffuse toujours ses messages BOOTREPLY.

Les plus anciennes mises en œuvre de client ou serveur qui établissent accidentellement le fanion BROADCAST DEVRAIENT être corrigées pour se conformer correctement à la présente spécification.

Tous les champs BOOTP DOIT être préservés intacts. L'agent de relais NE DOIT PAS modifier de champ BOOTP de message BOOTREPLY quand il le relaye au client.

La réponse DOIT avoir son accès de destination UDP réglé à BOOTPC (68).

Si la BOOTREPLY a une somme de contrôle UDP (c'est-à-dire, si la somme de contrôle UDP n'est pas à zéro) la somme de contrôle DOIT être recalculée avant de transmettre la réponse.

5. Comportement du serveur BOOTP

Cette section apporte des éclaircissements sur le comportement des serveurs BOOTP.

5.1 Réception des messages BOOTREQUEST

Tous les messages UDP reçus dont le numéro d'accès UDP de destination est BOOTPS (67) sont considérés pour le traitement par le serveur BOOTP.

Les hôtes et routeurs sont généralement obligés d'éliminer en silence les datagrammes entrants qui contiennent des adresses IP de source illégales. C'est généralement appelé le "filtrage des adresses martiennes". Une de ces adresses illégales est 0.0.0.0 (ou en fait tout de qui est sur le réseau 0). Cependant, les hôtes ou routeurs qui prennent en charge un serveur BOOTP DOIVENT accepter la livraison locale au serveur des messages BOOTREQUEST dont l'adresse IP de source est 0.0.0.0. Les messages BOOTREQUEST provenant d'adresses IP de source légales DOIVENT aussi être acceptés.

Un serveur BOOTP DOIT éliminer en silence tout message UDP reçu dont le numéro d'accès de destination UDP est BOOTPC (68).

Discussion : Il NE DEVRAIT PAS être nécessaire qu'un serveur BOOTP traite les messages adressés à l'accès BOOTPC. Un lecture attentive de la spécification BOOTP originale [RFC0951] le montre. Les vérifications de cohérence spécifiées au paragraphe 2.1 DEVRAIENT être effectuées par le serveur BOOTP. Les messages BOOTP qui ne satisfont pas à ces vérifications de cohérence DOIVENT être éliminés en silence.

5.2 Utilisation du champ "secs"

Lorsque le serveur BOOTP reçoit un message BOOTREQUEST, il PEUT utiliser la valeur du champ "secs" (secondes depuis l'amorçage du client) de la demande comme facteur de décision de répondre à la demande, et comment le faire.

Discussion : Aujourd'hui, cette caractéristique du protocole BOOTP ne s'est pas nécessairement montrée utile. Voir la discussion du paragraphe 3.2.

5.3 Utilisation du champ "ciaddr"

Il y a eu diverses interprétations du champ "ciaddr" par le client qui sont consultables au paragraphe 3.3. Un serveur BOOTP DEVRAIT être prêt à faire face à ces diverses interprétations. En général, le champ "ciaddr" NE DEVRAIT PAS être compris comme la seule clé pour identifier un client ; le contenu des champs "ciaddr", "chaddr", "htype", et "hlen", et probablement d'autres informations (peut-être dans les champs "file" et "vend") DEVRAIT être considéré globalement pour décider comment répondre à un client donné.

Les serveurs BOOTP DEVRAIENT préserver le contenu du champ "ciaddr" dans les messages BOOTREPLY; le contenu de "ciaddr" dans un message BOOTREPLY DEVRAIT correspondre exactement au contenu du "ciaddr" dans le message BOOTREQUEST correspondant.

Discussion : Il a été suggéré qu'un client puisse souhaiter utiliser le contenu de "ciaddr" pour vérifier qu'un message

BOOTREPLY particulier lui était bien destiné.

5.4 Stratégie de livraison des messages BOOTREPLY

Une fois que le serveur BOOTP a créé un message BOOTREPLY approprié, ce message BOOTREPLY DOIT être correctement livré au client.

Le serveur DEVRAIT d'abord vérifier le champ "ciaddr". Si le champ "ciaddr" n'est pas à zéro, le message BOOTREPLY DEVRAIT être envoyé comme envoi individuel IP à l'adresse IP identifiée dans le champ "ciaddr". L'accès UDP de destination DOIT être réglé à BOOTPC (68). Cependant, le serveur DOIT être conscient des problèmes identifiés au paragraphe 3.3. Le serveur PEUT choisir d'ignorer le champ "ciaddr" et d'agir comme si le champ "ciaddr" contenait 0.0.0.0 (et donc continuer avec le reste de l'algorithme de livraison ci-dessous).

Le serveur DEVRAIT ensuite vérifier le champ "giaddr". Si ce champ n'est pas zéro, le serveur DEVRAIT envoyer la BOOTREPLY comme envoi individuel IP à l'adresse IP identifiée dans le champ "giaddr". L'accès de destination UDP DOIT être réglé à BOOTPS (67). Cette action va livrer le message BOOTREPLY directement à l'agent de relais BOOTP le plus proche du client ; l'agent de relais va ensuite effectuer la livraison finale au client. Si le serveur BOOTP sait par avance qu'un client particulier ne peut pas recevoir de messages BOOTREPLY en envoi individuel (par exemple, le gestionnaire de réseau a explicitement configuré le serveur avec de tels éléments) le serveur PEUT régler le fanion BROADCAST qu'on vient de définir à indiquer que les agents de relais DEVRAIENT diffuser le message BOOTREPLY au client. Autrement, le serveur DOIT conserver l'état du fanion BROADCAST afin que l'agent de relais puisse agir correctement sur lui.

Si le champ "giaddr" est réglé à 0.0.0.0, le client réside sur un des mêmes réseaux que le serveur BOOTP. Le serveur DEVRAIT examiner le nouveau fanion BROADCAST (voir aux paragraphes 2.2, 3.1.1 et 4.1.2). Si ce fanion est à 1 ou si le serveur sait par avance que le client est incapable de recevoir des messages BOOTREPLY en envoi individuel, la réponse DEVRAIT être envoyée comme diffusion IP en utilisant l'adresse IP de diffusion limitée de 255.255.255.255 comme adresse IP de destination et l'adresse de diffusion de couche liaison comme adresse de destination de couche liaison. Si le fanion BROADCAST est à zéro (0), la réponse DEVRAIT être envoyée comme envoi individuel IP à l'adresse IP spécifiée dans le champ "yiaddr" et l'adresse de couche liaison spécifiée dans le champ "chaddr". Si l'envoi individuel n'est pas possible, la réponse PEUT être envoyée comme diffusion, auquel cas elle DEVRAIT être envoyée à l'adresse de diffusion de couche liaison en utilisant l'adresse IP de diffusion limitée de 255.255.255.255 comme adresse IP de destination. Dans tous les cas, l'accès UDP de destination DOIT être réglé à BOOTPC (68).

Discussion : l'ajout du fanion BROADCAST au protocole est une astuce pour aider à promouvoir l'interopérabilité avec certaines mises en œuvre de client.

Le tableau qui suit résume les décisions de livraison du serveur pour les messages BOOTREPLY sur la base des informations des messages BOOTREQUEST :

Champs BOOTREQUEST			**Valeurs de BOOTREPLY pour UDP, IP, couche liaison**		
"ciaddr"	"giaddr"	B	destination UDP	destination IP	dest. liaison
non zéro	X	X	BOOTPC (68)	"ciaddr"	normal
0.0.0.0	non zéro	X	BOOTPS (67)	"giaddr"	normal
0.0.0.0	0.0.0.0	0	BOOTPC (68)	"yiaddr"	"chaddr"
0.0.0.0	0.0.0.0	1	BOOTPC (68)	255.255.255.255	diffusion

B = fanion BROADCAST
X = non applicable
normal = déterminé à partir de la destination IP donné en utilisant les mécanismes d'acheminement IP normaux et/ou ARP comme pour tout autre datagramme normal.

Remerciements

L'auteur tient à remercier Gary Malkin de sa contribution à la section "BOOTP sur réseau à anneau à jetons IEEE 802.5", et Steve Deering pour ses observations sur les problèmes associés au champ "giaddr".

Ralph Droms et les nombreux membres des groupes de travail de l'IETF Configuration dynamique d'hôte et Exigences pour les routeurs ont fourni des idées pour le présent mémoire ainsi que des encouragement à l'écrire.

Philip Almquist et David Piscitello ont formulé de nombreuses suggestions utiles pour améliorer la clarté, la précision, et

l'organisation de ce mémo. Ces contributions ont été les bienvenues.

Références

[RFC0826] D. Plummer, "Protocole de résolution d'adresses Ethernet : conversion des adresses de protocole réseau en adresses Ethernet à 48 bits pour transmission sur un matériel Ethernet", STD 37, novembre 1982.

[RFC0951] B. Croft et J. Gilmore, "Protocole BOOTSTRAP (BOOTP)", septembre 1985.

[RFC1256] S. Deering, éditeur, "Messages ICMP de découverte de routeur", septembre 1991.

[RFC1340] J. Reynolds et J. Postel, "Numéros alloués", STD 2, juillet 1992. *(Historique, voir www.iana.org)*.

[RFC1497] J. Reynolds, "Extensions Informations de fabricant BOOTP", août 1993. *(Remplacée par la* RFC1533) Cette RFC est occasionnellement republiée avec un nouveau numéro. S'assurer de consulter la dernière version.

[RFC1541] R. Droms, "Protocole de configuration dynamique d'hôte", octobre 1993. *(P.S., remplaceé par* 2131*)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

Considérations sur la sécurité

De nombreux facteurs rendent BOOTP sous sa forme actuelle assez peu sûr. BOOTP est construit directement par dessus UDP et IP qui sont déjà eux-mêmes non sûrs de façon inhérente. De plus, BOOTP est généralement destiné à rendre plus facile la maintenance d'hôtes distants et/ou sans disque dur. Bien que ce ne soit peut-être pas impossible, configurer de tels hôtes avec des mots de passe ou des clés PEUT être difficile et peu pratique. Cela rend difficile de fournir une forme raisonnable d'authentification entre serveurs et clients.

Des serveurs BOOTP non autorisés PEUVENT facilement être établis. De tels serveurs peuvent alors envoyer de fausses informations potentiellement perturbatrices aux clients comme des adresses IP incorrectes ou dupliquées, des informations d'acheminement incorrectes (incluant des routeurs usurpés, etc.) des adresses incorrectes de serveur de noms de domaines (comme des serveurs de noms usurpés) et ainsi de suite. Il est clair que une fois que ce "germe" de désinformation est planté, un attaquant peut compromettre les systèmes affectés.

Des agents de relais BOOTP non autorisés PEUVENT poser les mêmes problèmes que des serveurs BOOTP non autorisés.

Des clients BOOTP malveillants pourraient se faire passer pour des clients légitimes et récupérer des informations destinées à ces clients légitimes. Lorsque l'allocation dynamique de ressources est utilisée, un client malveillant pourrait réclamer toutes les ressources pour lui-même, déniant ainsi les ressources aux clients légitimes.

Adresse de l'auteur

Walt Wimer
Network Development
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
USA
téléphone : (412) 268-6252
mél : Walter.Wimer@CMU.EDU