Groupe de travail Réseau
Request for Comments : 4941
RFC rendue obsolète : 3041
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

T. Narten, IBM Corporation
R. Draves, Microsoft Research
S. Krishnan, Ericsson Research
septembre 2007

Extensions de confidentialité pour l'autoconfiguration d'adresse sans état dans IPv6

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet sur la voie de la 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 "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

(La présente traduction incorpore les errata 998, 3239, 3240, 3241, 3242, 3243, 4404, et 4763)

Résumé

Les nœuds utilisent l'autoconfiguration d'adresse IPv6 sans état pour générer des adresses en utilisant une combinaison d'informations localement disponibles et d'informations annoncées par les routeurs. Les adresses sont formées en combinant les préfixes de réseau avec un identifiant d'interface. Sur une interface qui contient un identifiant IEEE incorporé, l'identifiant d'interface en est normalement déduit. Sur les autres types d'interface, l'identifiant d'interface est généré par d'autres moyens, par exemple, via la génération de nombres aléatoires. Le présent document décrit une extension à l'autoconfiguration d'adresse IPv6 sans état pour les interfaces dont l'identifiant d'interface est déduit d'un identifiant IEEE. L'utilisation de l'extension cause la génération par les nœuds d'adresses de portée mondiale à partir des identifiants d'interface qui changent au fil du temps, même dans les cas où l'interface contient un identifiant IEEE incorporé. Changer l'identifiant d'interface (et les adresses de portée mondiale générées à partir de lui) au fil du temps rend plus difficile à un espion et autre collecteur d'informations d'identifier quand différentes adresses utilisées dans différentes transactions correspondent en fait au même nœud.

Table des matières

1. Introduction

La [RFC4862] "Auto configuration d'adresse sans état" définit comment un nœud IPv6 génère des adresses sans avoir besoin d'un serveur du protocole de configuration dynamique d'hôte pour IPv6 (DHCPv6, *Dynamic Host Configuration Protocol for IPv6*). Certains types d'interfaces réseau ont un identifiant IEEE incorporé (c'est-à-dire, une adresse MAC de couche de liaison) et dans ce cas, l'auto configuration d'adresse sans état utilise l'identifiant IEEE pour générer un identifiant d'interface de 64 bits [RFC4291]. Par conception, l'identifiant d'interface va probablement être unique au monde quand il est généré de cette façon. L'identifiant d'interface est à son tour ajouté à un préfixe pour former une adresse IPv6 de 128 bits. Noter qu'un identifiant IPv6 ne doit pas nécessairement être long de 64 bits, mais l'algorithme spécifié dans le présent document est ciblé sur des identifiants d'interface de 64 bits.

Tous les nœuds combinent les identifiants d'interface (qu'ils soient déduits d'un identifiant IEEE ou générés par une autre technique) avec le préfixe de liaison locale réservé pour générer les adresses de liaison locale pour leurs interfaces rattachées. Des adresses supplémentaires peuvent alors être créées en combinant les préfixes annoncés dans les annonces de routeur via la découverte de voisin [RFC4861] avec l'identifiant d'interface.

Tous les nœuds et interfaces ne contiennent pas d'identifiants IEEE. Dans ce cas, un identifiant d'interface est généré par d'autres moyens (par exemple, au hasard) et l'identifiant d'interface résultant peut n'être pas unique au monde et peut aussi changer dans le temps. Le présent document se concentre sur les adresses déduites des identifiants IEEE parce que le suivi des appareils individuels, qui est le sujet traité ici, n'est possible que dans les cas où l'identifiant d'interface est unique au monde et ne change pas. Le reste de ce document suppose que des identifiants IEEE sont utilisés, mais les techniques décrites peuvent aussi s'appliquer aux interfaces sur d'autres types d'identifiants uniques au monde et/ou persistants.

Le présent document discute les problèmes associés à l'incorporation d'identifiants d'interface non changeants au sein des adresses IPv6 et décrit des extensions à l'autoconfiguration d'adresse sans état qui peuvent aider à atténuer ces problèmes pour les utilisateurs individuels et dans des environnements où de tels problèmes sont significatifs. La Section 2 donne les informations de base sur la question. La Section 3 décrit une procédure pour générer des identifiants d'interface de remplacement et des adresses de portée mondiale. La Section 4 discute les implications du changement des identifiants d'interface. Le terme "adresses de portée mondiale" est utilisé dans le présent document pour se référer collectivement aux "adresses en envoi individuel mondiales" comme défini dans la [RFC4291] et aux "adresses uniques locales" comme défini dans la [RFC4193].

1.1 Conventions utilisées dans ce document

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 Position du problème

Les adresses générées en utilisant l'autoconfiguration d'adresse sans état [RFC4862] contiennent un identifiant d'interface incorporé, qui reste constant dans le temps. Chaque fois qu'un identifiant fixe est utilisé dans plusieurs contextes, il devient possible de corréler des activités sans relation entre elles en utilisant cet identifiant.

La corrélation peut être effectuée par :
- Un attaquant qui est dans le chemin entre le nœud en question et le ou les homologues avec lesquels il communique, et qui peut voir les adresses IPv6 présentes dans les datagrammes.
- Un attaquant qui peut accéder aux enregistrements de communication des homologues avec lesquels le nœud a communiqué.

Comme l'identifiant est incorporé dans l'adresse IPv6, qui est une exigence fondamentale de communication, il ne peut pas être facilement caché. Le présent document propose une solution à ce problème en générant des identifiants d'interface qui varient dans le temps.

Noter qu'un attaquant, qui est dans le chemin, peut être capable d'effectuer une corrélation significative fondée sur :
- le contenu de la charge utile des paquets dans le réseau,
- les caractéristique des paquets telles que la taille des paquets et leur rythme.

L'utilisation d'adresses temporaires ne va pas empêcher de telles corrélations fondées sur la charge utile.

2. Fondements

Cette section discute plus en détails le problème, donne le contexte pour évaluer la signification des problèmes dans des environnements spécifiques et fait des comparaisons avec les pratiques existantes.

2.1 Utilisation étendue du même identifiant

L'utilisation d'un identifiant d'interface non changeant pour former des adresses est une instance spécifique du cas plus général où un identifiant constant est réutilisé sur une longue période et dans plusieurs activités indépendantes. Chaque fois que le même identifiant est utilisé dans plusieurs contextes, il devient possible que cet identifiant soit utilisé pour corréler des activités apparemment sans relation. Par exemple, un renifleur de réseau placé stratégiquement sur une liaison à travers laquelle passe tout le trafic de/vers un hôte particulier pourrait garder trace des destinations avec lesquelles un nœud a communiqué et à quels moments. De telles informations peuvent dans certains cas être utilisées pour déduire des choses, comme les heures d'activité d'un employé, quand quelqu'un est chez lui, etc. Bien qu'il puisse paraître que changer régulièrement une adresse dans de tels environnements serait souhaitable pour diminuer les problèmes de confidentialité, on devrait noter que la portion préfixe réseau d'une adresse sert aussi comme identifiant constant. Tous les nœuds, chez eux, vont avoir le même préfixe réseau, qui identifie la localisation topologique de ces nœuds. Cela a des implications pour la confidentialité, bien que pas à la même granularité que les problèmes que traite le présent document. Spécifiquement, tous les nœuds dans un rattachement pourraient être groupés pour les besoins de la collecte d'informations. Si le réseau contient un très petit nombre de nœuds, disons, juste un, changer juste l'identifiant d'interface ne va pas du tout améliorer la confidentialité, car le préfixe sert d'identifiant constant.

Une des exigences pour corréler des activités apparemment sans relation est l'utilisation (et la réutilisation) d'un identifiant qui soit reconnaissable au fil du temps dans différents contextes. Les adresses IP en donnent un exemple évident, mais il y en a d'autres. De nombreux nœuds ont aussi des noms de DNS associés à leurs adresses, et dans ce cas, le nom DNS sert d'identifiant similaire. Bien que le nom DNS associé à une adresse donne plus de travail à obtenir (il peut exiger une interrogation du DNS) l'information est souvent directement disponible. Dans ce cas, changer l'adresse sur une machine au fil du temps aurait peu d'incidence pour traiter les problèmes soulevés dans le présent document, sauf si le nom DNS est aussi changé (voir la Section 4).

Les navigateurs et serveurs de la Toile échangent normalement des "cookies" *(mouchards)* les uns avec les autres [RFC2965]. Les mouchards permettent aux serveurs de la Toile de corréler une activité actuelle avec une activité précédente. Une utilisation courante est de renvoyer une annonce ciblée à un utilisateur en utilisant le mouchard fourni par le navigateur pour identifier les interrogations qui ont été faites antérieurement (par exemple, pour quel type d'information). Sur la base des interrogations antérieures, les annonces peuvent être ciblées pour correspondre aux intérêts (supposés) de l'utilisateur final.

L'utilisation d'un identifiant constant dans une adresse est un souci particulier parce que les adresses sont une exigence fondamentale de communication et ne peuvent pas être facilement cachées aux espions et aux autres tiers. Même quand les couches supérieures chiffrent leur charge utile, les adresses dans les en-têtes de paquet apparaissent en clair. Par conséquent, si un hôte mobile (par exemple, une tablette) accède au réseau à partir de plusieurs localisations différentes, un espion pourrait être capable de retracer le mouvement de cet hôte mobile de lieu en lieu, même si les charges utiles de couche supérieure étaient chiffrées.

2.2 Usage d'adresse dans IPv4 aujourd'hui

Les adresses utilisées dans l'Internet d'aujourd'hui sont souvent non changeantes en pratique pendant de longues périodes. Dans un nombre croissant de sites, les adresses sont allouées statiquement et ne changent normalement que peu fréquemment. Ces quelques dernières années, les sites ont commencé à s'éloigner de l'allocation statique au profit d'une allocation dynamique via DHCP [RFC2131]. En théorie, l'adresse qu'un client obtient via DHCP peut changer dans le temps, mais en pratique les serveurs retournent souvent la même adresse au même client (sauf si les adresses sont si rares qu'elles sont réutilisées immédiatement par un nœud différent quand elles se libèrent). Donc, même dans les sites qui utilisent DHCP, les clients finissent fréquemment par utiliser la même adresse pendant des semaines ou des mois.

Pour les utilisateurs domestiques qui accèdent à l'Internet sur des lignes à numérotation, la situation est généralement différente. De tels utilisateurs n'ont pas de connexions permanentes et des adresses temporaires leur sont souvent allouées chaque fois qu'ils se connectent à leur fournisseur d'accès Internet (FAI). Par conséquent, les adresses qu'ils utilisent

changent fréquemment et sont partagées entre un certain nombre d'utilisateurs différents. Donc, une adresse n'identifie pas de façon fiable un appareil particulier sur un espace de temps de plus de quelques minutes.

Un cas plus intéressant concerne les connexions toujours actives (par exemple, modems câble, RNIS, DSL, etc.) qui résultent en un site qui utilise la même adresse pour de longues périodes. C'est un scénario qui commence juste à devenir courant dans IPv4 et promet de devenir un problème lorsque la connexité permanente à l'Internet devient largement disponible.

Finalement, on devrait noter que les nœuds qui ont besoin d'un nom DNS (non changeant) ont généralement des adresses statiques qui leur sont allouées pour simplifier la configuration des serveurs DNS. Bien que le DNS dynamique [RFC3136] puisse être utilisé pour mettre dynamiquement à jour le DNS, il peut n'être pas toujours disponible selon la politique administrative. De plus, changer une adresse mais garder le même nom DNS ne règle pas réellement le problème sous-jacent, car le nom DNS devient un identifiant qui ne change pas. Les serveurs exigent généralement un nom DNS (afin que les clients puissent s'y connecter) et les clients le font souvent aussi (par exemple, certains serveurs refusent de parler à un client dont l'adresse ne peut pas être transposée en un nom DNS qui se retranspose aussi en la même adresse). La Section 4 décrit une approche de ce problème.

2.3 Le problème des adresses IPv6

La division des adresses IPv6 en portions distinctes de topologie et d'identifiant d'interface soulève un nouveau problème pour IPv6 en ce que une portion fixe d'une adresse IPv6 (c'est-à-dire, l'identifiant d'interface) peut contenir un identifiant qui reste constant même quand la portion topologique d'une adresse change (par exemple, par suite de la connexion à une partie différente de l'Internet). Dans IPv4, quand une adresse change, l'adresse entière (y compris la partie locale de l'adresse) change généralement. C'est ce nouveau problème que traite le présent document.

Si les adresses sont générées à partir d'un identifiant d'interface, l'adresse d'un utilisateur résidentiel pourrait contenir un identifiant d'interface qui reste le même d'une session avec numérotage à la suivante, même si le reste de l'adresse change. À la différence des clients de LAN, la façon dont PPP est utilisé aujourd'hui ne permet cependant pas au client de changer librement et unilatéralement son propre identifiant d'interface. Les serveurs PPP n'informent pas non plus unilatéralement le client de quelle adresse de 128 bits utiliser. De la même manière que pour les clients connectés à un LAN (sans PPP) un client PPP génère son adresse IPv6 de portée mondiale à partir de la partie Identifiant d'interface et de la partie Préfixe. Le client obtient un préfixe du serveur PPP de la même façon que le font les clients de LAN, c'est-à-dire, via le protocole de découverte de voisin. Cependant, un identifiant d'interface de client PPP DOIT être négocié avec le serveur PPP via le protocole IPv6CP (option Identifiant d'interface). Seulement alors, après que IPv6CP a été négocié avec succès entre le client PPP et le serveur, le client PPP peut générer et allouer son adresse individuelle IPv6 mondiale composée à partir de l'identifiant d'interface négocié et le préfixe obtenu. Afin de changer plus tard l'ancien identifiant d'interface négocié dans la même session PPP, le client DOIT initier une nouvelle négociation IPv6CP avec le serveur PPP, proposant un nouvel identifiant d'interface du client.

Un cas plus troublant concerne les appareils mobiles (par exemple, tablettes, PDA, etc.) qui se déplacent topologiquement au sein de l'Internet. Chaque fois qu'ils bougent, ils forment de nouvelles adresses pour leur point de rattachement topologique actuel. Son type aujourd'hui est le "routard" qui a la connexité Internet à la fois chez lui et au bureau. Alors que l'adresse du nœud change quand il bouge, l'identifiant d'interface contenu dans l'adresse reste le même (quand il est déduit d'un identifiant IEEE). Dans ce cas, l'identifiant d'interface peut être utilisé pour retracer le mouvement et l'usage d'une machine particulière. Par exemple, un serveur qui enregistre les informations d'usage avec les adresses de source, enregistre aussi l'identifiant d'interface parce qu'il est incorporé à l'adresse. Par conséquent, toute technique d'extraction de données qui corrèle l'activité fondée sur les adresses pourrait facilement être étendue à faire de même en utilisant l'identifiant d'interface. Ceci est particulièrement un problème avec la prolifération attendue d'appareils connectés au réseau de prochaine génération (par exemple, les PDA, téléphones cellulaires, etc.) dans lesquels de grands nombres d'appareils sont, en pratique, associés à un utilisateur individuel (c'est-à-dire, non partagés). Donc, l'identifiant d'interface incorporé dans une adresse pourrait être utilisé pour retracer les activités d'un individu, même lorsque il se déplace topologiquement dans l'Internet.

En résumé, les adresses IPv6 sur une interface générée via l'autoconfiguration sans état contiennent le même identifiant d'interface, sans considération du lieu dans l'Internet où l'appareil se connecte. Cela facilite le traçage des appareils individuels (et donc, potentiellement, des utilisateurs). L'objet du présent document est de définir des mécanismes qui éliminent ce problème dans les situations où c'est un problème.

2.4 Approches possibles

Une façon d'éviter d'avoir une adresse statique non changeante est d'utiliser DHCPv6 [RFC3315] pour obtenir les adresses. La Section 12 de la [RFC3315] discute l'utilisation de DHCPv6 pour l'allocation et la gestion des "adresses temporaires", qui ne sont jamais renouvelées et fournissent les mêmes propriétés que les adresses temporaires décrites dans ce document à l'égard de la confidentialité.

Une autre approche, compatible avec l' architecture d'autoconfiguration d'adresse sans état, serait de changer la portion "identifiant d'interface" d'une adresse au bout d'un moment et de générer de nouvelles adresses à partir de l'identifiant d'interface pour certaines portées d'adresses. Changer l'identifiant d'interface peut rendre plus difficile de chercher les adresses IP dans des transactions indépendantes et d'identifier lesquelles correspondent réellement au même nœud, dans le cas où la portion préfixe d'acheminement d'une adresse change et dans celui où elle ne change pas.

De nombreuses machines fonctionnent à la fois comme clients et comme serveurs. Dans ce cas, la machine va avoir besoin d'un nom DNS pour son utilisation comme serveur. Que l'adresse reste fixe ou change a peu d'implications sur la confidentialité car le nom DNS reste constant et sert d'identifiant constant. Cependant, quand elle agit comme client (par exemple, initiant la communication) une telle machine peut vouloir varier les adresses qu'elle utilise. Dans un tel environnement, on peut avoir besoin de plusieurs adresses : une adresse de serveur "publique" (c'est-à-dire, non secrète) enregistrée dans le DNS, qui est utilisée pour accepter les demandes de connexion entrantes provenant des autres machines, et une adresse "temporaire" utilisée pour masquer l'identité du client quand il initie la communication. Ces deux cas sont en gros analogues aux numéros de téléphone et identifiants d'appelant, où un utilisateur peut mentionner son numéro de téléphone dans l'annuaire public, mais désactiver l'affichage de son numéro via l'identifiant d'appelant quand il initie des appels.

Pour rendre difficile de deviner intelligemment si deux identifiants d'interface différents appartiennent au même nœud, l'algorithme pour générer des identifiants de remplacement doit inclure des entrées qui ont un composant imprévisible du point de vue des entités extérieures qui collectent des informations. Prendre des identifiants à partir d'une séquence pseudo-aléatoire suffit, pour autant que la séquence spécifique ne puisse pas être déterminée par un extérieur qui examine les informations directement disponibles ou facilement déterminables (par exemple, en examinant le contenu des paquets). Le présent document propose la génération d'une séquence pseudo-aléatoire d'identifiants d'interface via un hachage MD5. Périodiquement, le prochain identifiant d'interface dans la séquence est généré, un nouvel ensemble d'adresses temporaires est créé, et les précédentes adresses temporaires sont déconseillées pour dissuader de continuer leur utilisation. La séquence pseudo-aléatoire précise dépend du composant aléatoire et de l'identifiant d'interface unique au monde (quand il est disponible) pour augmenter la probabilité que différents nœuds génèrent des séquences différentes.

3. Description du protocole

Le but de cette section est de définir des procédures qui :

1. Ne résultent en aucun changement du comportement de base des adresses générées via l'autoconfiguration d'adresse sans état [RFC4862].

2. Créent des adresses supplémentaires fondées sur un identifiant d'interface aléatoire pour initier des sessions sortantes. Ces adresses "aléatoires" ou temporaires vont être utilisées pour une courte période (heures ou jours) et vont ensuite être déconseillées. Les adresses déconseillées peuvent continuer d'être utilisées pour les connexions déjà établies, mais ne sont pas utilisées pour initier de nouvelles connexions. De nouvelles adresses temporaires sont générées périodiquement pour remplacer les adresses temporaires qui arrivent à expiration, le temps exact entre les générations d'adresses étant une affaire de politique locale.

3. Produisent une séquence d'adresses temporaires de portée mondiale à partir d'une séquence d'identifiants d'interface qui paraisse être aléatoire au sens où il est difficile à un observateur extérieur de prédire une adresse (ou identifiant) future sur la base de l'actuelle, et il est difficile de déterminer les adresses (ou identifiants) précédents en connaissant la présente.

4. Par défaut, génèrent un ensemble d'adresses à partir du même identifiant d'interface (aléatoire) une adresse pour chaque préfixe pour lequel une adresse mondiale a été générée via l'autoconfiguration d'adresse sans état. Utiliser le même identifiant d'interface pour générer un ensemble d'adresses temporaires réduit le nombre de groupes de diffusion groupée IP auxquels un hôte doit se joindre. Les nœuds joignent l'adresse de diffusion groupée de nœud sollicité pour chaque adresse d'envoi individuel qu'ils prennent en charge, et les adresses de nœud sollicité dépendent seulement des

bits de moindre poids de l'adresse correspondante. Ce comportement par défaut est destiné à répondre au problème qu'un nœud qui se joint à un grand nombre de groupes de diffusion groupée peut être obligé de mettre son interface en mode promiscuité, résultant en une possible réduction des performances.

Un nœud très concerné par la confidentialité PEUT utiliser des identifiants d'interface différents sur les différents préfixes, résultant en un ensemble d'adresses mondiales qui ne peuvent pas être facilement reliées les unes aux autres. Par exemple un nœud PEUT créer différents identifiants d'interface I1, I2, et I3 à utiliser avec différents préfixes P1, P2, et P3 sur la même interface.

3.1 Hypothèses

L'algorithme suivant suppose que chaque interface maintient un identifiant d'interface aléatoire associé. Quand des adresses temporaires sont générées, la valeur courante de l'identifiant d'interface aléatoire associé est utilisée. Alors que le même identifiant peut être utilisé pour créer plus d'une adresse temporaire, la valeur DEVRAIT changer avec le temps comme décrit au paragraphe 3.5.

L'algorithme suppose aussi que, pour une adresse temporaire donnée, une mise en œuvre peut déterminer le préfixe à partir duquel il a été généré. Quand une adresse temporaire est déconseillée, une nouvelle adresse temporaire est générée. Les durées de vie spécifiques valides et préférées pour la nouvelle adresse dépendent des valeurs correspondantes de durée de vie établies pour le préfixe à partir duquel elle a été générée.

Finalement, le présent document suppose que quand un nœud initie une communication sortante, les adresses temporaires peuvent avoir la préférence sur des adresses publiques quand l'appareil est configuré à le faire. La [RFC3484] oblige les mises en œuvre à fournir un tel mécanisme, qui permet à une application de configurer sa préférence pour les adresses temporaires plutôt que les adresses publiques. Elle permet aussi à une mise en œuvre de préférer les adresses temporaires par défaut, afin que les connexions initiées par le nœud puissent utiliser des adresses temporaires sans exiger une activation spécifique de l'application. Le présent document suppose aussi qu'une API existera pour permettre aux applications individuelles d'indiquer si elles préfèrent utiliser des adresses temporaires ou publiques et outrepasser les valeurs par défaut du système.

3.2 Génération d'identifiants d'interface aléatoires

On décrit deux approches pour la génération et la maintenance de l'identifiant d'interface aléatoire. La première suppose la présence d'une mémorisation stable qui peut être utilisée pour enregistrer l'historique d'état à utiliser comme entrée dans la prochaine itération de l'algorithme à travers les redémarrages du système. Une seconde approche vise le cas où une mémorisation stable n'est pas disponible et où il est besoin de générer des identifiants d'interface aléatoires sans l'état précédent.

L'algorithme de génération d'identifiant d'interface aléatoire, comme décrit dans ce document, utilise MD5 comme algorithme de hachage. Le nœud PEUT utiliser un autre algorithme à la place de MD5 pour produire l'identifiant d'interface aléatoire.

3.2.1 Quand une mémorisation stable est présente

L'algorithme suivant suppose la présence d'une "valeur d'historique" de 64 bits qui est utilisée comme entrée pour générer un identifiant d'interface aléatoire. La première fois que le système s'amorce (c'est-à-dire, sort de la boîte) une valeur aléatoire DEVRAIT être générée en utilisant des techniques qui aident à s'assurer que la valeur initiale est difficile à deviner [RFC4086]. Chaque fois qu'un nouvel identifiant d'interface est généré, une valeur générée par le calcul est sauvegardée dans la valeur d'historique pour la prochaine itération de l'algorithme.

Un identifiant d'interface aléatoire est créé comme suit :

1. Prendre la valeur d'historique de la précédente itération de cet algorithme (ou une valeur aléatoire si il n'y a pas de valeur précédente) et l'ajouter à l'identifiant d'interface généré comme décrit dans la [RFC4291].

2. Calculer le résumé de message MD5 [RFC1321] sur la quantité créée à l'étape précédente.

3. Prendre les 64 bits de gauche du résumé MD5 et régler le bit 6 (le bit de gauche porte le numéro 0) à zéro. Cela crée un identifiant d'interface avec le bit universel/local qui indique la signification seulement locale.

4. Comparer l'identifiant généré à une liste d'identifiants d'interface réservés et à ceux déjà alloués à une adresse sur l'appareil local. Au cas où un identifiant inacceptable aurait été généré, le nœud DOIT recommencer le processus à l'étape 1, en utilisant les 64 bits de droite du résumé MD5 obtenu à l'étape 2 à la place de la valeur d'historique de l'étape 1.

5. Sauvegarder l'identifiant généré comme identifiant d'interface aléatoire associé.

6. Prendre les 64 bits de droite du résumé MD5 calculé à l'étape 2 et les sauvegarder dans une mémorisation stable comme valeur d'historique à utiliser dans la prochaine itération de l'algorithme.

MD5 a été choisi par convention, et parce que ses propriétés particulières sont adéquates pour produire le niveau désiré d'aléa. Le nœud PEUT utiliser un autre algorithme à la place de MD5 pour produire l'identifiant d'interface aléatoire.

En théorie, générer des identifiants d'interface aléatoires successifs en utilisant un schéma d'historique comme ci-dessus ne présente pas d'avantage par rapport à une génération aléatoire. En pratique, cependant, générer des nombres vraiment aléatoires peut être délicat. L'utilisation d'une valeur d'historique est destinée à éviter le scénario particulier où deux nœuds génèrent le même identifiant d'interface aléatoire, détectent tous deux la situation via la DAD, mais procèdent ensuite à générer des identifiants d'interface aléatoires identiques via le même algorithme (fautif) de génération de nombres aléatoires. L'algorithme ci-dessus évite ce problème en utilisant l'identifiant d'interface (qui va souvent être unique au monde) dans le calcul qui génère les identifiants d'interface aléatoires suivants. Donc, si deux nœuds se trouvent générer le même identifiant d'interface aléatoire, ils devraient en générer des différents à la tentative suivante.

3.2.2 En l'absence de mémorisation stable

En l'absence de mémorisation stable, aucune valeur d'historique ne va être disponible à travers les redémarrages du système pour générer une séquence pseudo-aléatoire d'identifiants d'interface. Par conséquent, la valeur initiale d'historique utilisée ci-dessus DEVRAIT être générée au hasard. Un certain nombre de techniques pourraient être appropriées. Consulter la [RFC4086] pour des suggestions sur de bonnes sources pour obtenir des nombres aléatoires. Noter que même si les machines ne peuvent pas avoir de mémorisation stable pour mémoriser une valeur d'historique, elle vont souvent avoir des informations de configuration qui diffèrent d'une machine à l'autre (par exemple, l'identité de l'utilisateur, les clés de sécurité, les numéros de série, etc.). Une approche pour générer une valeur initiale d'historique aléatoire dans ce cas est d'utiliser les informations de configuration pour générer des bits de données (qui peuvent rester constants pour la vie de la machine, mais vont varier d'une machine à l'autre) ajouter des données aléatoires, et calculer le résumé MD5 comme avant.

3.2.3 Autres approches

Noter qu'il y a d'autres approches pour générer des identifiants d'interface aléatoires, mais avec des buts et une applicabilité différents. Un de ces approches est celle des adresses générées cryptographiquement (CGA, *Cryptographically Generated Addresses*) [RFC3972], qui génèrent un identifiant d'interface aléatoire fondé sur la clé publique du nœud. Le but des CGA est de prouver la propriété d'une adresse et d'empêcher l'usurpation d'identité et le vol des adresses IPv6 existantes. Elles sont utilisées pour sécuriser la découverte de voisin en utilisant la [RFC3971]. L'algorithme de génération d'identifiant d'interface aléatoire par CGA peut ne pas convenir pour la confidentialité des adresses à cause des propriétés suivantes :

- Il exige que le nœud ait une clé publique. Cela signifie que le nœud peut quand même être identifié par sa clé publique.
- Le processus d'identifiant d'interface aléatoire est coûteux en calcul et donc décourage les régénérations fréquentes.

3.3 Génération d'adresses temporaires

La [RFC4862] décrit les étapes de génération d'une adresse de liaison locale quand une interface devient active ainsi que les étapes pour générer des adresses pour d'autres portées. Le présent document étend la [RFC4862] comme suit. Quand il traite une annonce de routeur avec une option Informations de préfixe qui porte un préfixe de portée mondiale pour les besoins de l'autoconfiguration d'adresse (c'est-à-dire, le bit A est établi) le nœud DOIT effectuer les étapes suivantes :

1. Traiter l'option Informations de préfixe comme défini dans la [RFC4862], soit en créant une nouvelle adresse publque, soit en ajustant la durée de vie des adresses existantes, publiques et temporaires. Si une option reçue va étendre la durée de vie d'une adresse publique, la durée de vie des adresses temporaires devrait être étendue, sous réserve de la contrainte globale qu'aucune adresse temporaire ne devrait jamais rester "valide" ou "préférée" pour une durée supérieure à (TEMP_VALID_LIFETIME) ou (TEMP_PREFERRED_LIFETIME - DESYNC_FACTOR), respectivement. Les variables de configuration TEMP_VALID_LIFETIME et TEMP_PREFERRED_LIFETIME

correspondent approximativement aux durées de vie cibles pour les adresses temporaires.

2. Une façon dont une mise en œuvre peut satisfaire les contraintes ci-dessus est d'associer à chaque adresse temporaire une heure de création (appelée CREATION_TIME) qui indique l'heure à laquelle l'adresse a été créée. Quand on met à jour la durée de vie préférée d'une adresse temporaire existante, elle va être réglée à expirer à l'heure la plus proche quelle qu'elle soit : l'heure indiquée par la durée de vie reçue ou (CREATION_TIME + TEMP_PREFERRED_LIFETIME - DESYNC_FACTOR). Une approche similaire peut être utilisée avec la durée de vie valide.

3. Quand une nouvelle adresse publique est créée comme décrit dans la [RFC4862], le nœud DEVRAIT aussi créer une nouvelle adresse temporaire.

4. Quand il crée une adresse temporaire, les valeurs de durée de vie DOIVENT être déduites du préfixe correspondant comme suit :
   * Sa durée de vie valide est le plus petit de la durée de vie valide du préfixe ou de TEMP_VALID_LIFETIME.
   * Sa durée de vie préférée est le plus petit de la durée de vie préférée du préfixe ou de TEMP_PREFERRED_LIFETIME - DESYNC_FACTOR.

5. Une adresse temporaire n'est créée que si cette durée de vie préférée calculée est supérieure à REGEN_ADVANCE unités de temps. En particulier, une mise en œuvre NE DOIT PAS créer une adresse temporaire avec une durée de vie préférée de zéro.

6. Les nouvelles adresses temporaires DOIVENT être créées en ajoutant l'identifiant d'interface aléatoire de l'interface au préfixe qui a été reçu.

7. Le nœud DOIT effectuer la détection d'adresse dupliquée (DAD, *Duplicate Address Detection*) sur l'adresse temporaire générée. Si la DAD indique que l'adresse est déjà utilisée, le nœud DOIT générer un nouvel identifiant d'interface aléatoire comme décrit au paragraphe 3.2, et répéter les étapes précédentes comme approprié jusqu'à TEMP_IDGEN_RETRIES fois. Si après TEMP_IDGEN_RETRIES tentatives consécutives, aucune adresse non unique n'a été générée, le nœud DOIT enregistrer une erreur système et NE DOIT PAS tenter de générer des adresses temporaires pour cette interface. Noter que la DAD DOIT être effectuée sur chaque adresse d'envoi individuel générée à partir de cet identifiant d'interface aléatoire.

3.4 Expiration des adresses temporaires

Quand une adresse temporaire devient déconseillée, une nouvelle DOIT être générée. C'est fait en répétant les actions décrites au paragraphe 3.3, en commençant à l'étape 4). Noter que, sauf pour la période transitoire quand une adresse temporaire est générée à nouveau, en fonctionnement normal au plus une adresse temporaire par préfixe devrait être dans un état non déconseillée à tout moment sur une interface. Noter que si une adresse temporaire devient déconseillée par suite du traitement d'une option Informations de préfixe avec une durée de vie préférée de zéro, alors une nouvelle adresse temporaire NE DOIT PAS être générée. Pour assurer qu'une adresse temporaire préférée est toujours disponible, une nouvelle adresse temporaire DEVRAIT être générée à nouveau légèrement avant que la précédente soit déconseillée. C'est pour permettre un temps suffisant pour éviter des conditions de compétition dans les cas où la génération d'une nouvelle adresse temporaire n'est pas instantanée, comme quand la détection d'adresse dupliquée doit être effectuée. Le nœud DEVRAIT commencer le processus de regénération d'adresses REGEN_ADVANCE unités de temps avant qu'une adresse temporaire soit effectivement déconseillée.

Comme optimisation facultative, une mise en œuvre PEUT supprimer une adresse temporaire déconseillée qui n'est pas utilisée par les applications ou les couches supérieures comme précisé à la Section 6.

3.5 Régéneration des identifiants d'interface aléatoires

La fréquence à laquelle les adresses temporaires changent dépend de la façon dont un appareil est utilisé (par exemple, à quelle fréquence il initie de nouvelles communications) et les centres d'intérêt de l'utilisateur final. Le plus gros souci de confidentialité apparaît avec les adresses utilisées pour de longues périodes (des semaines, des mois ou des années). Plus fréquemment une adresse change, moins il est faisable de collecter ou coordonner les informations fondées sur les identifiants d'interface. De plus, le coût de collecte des informations et des tentatives de les corréler fondées sur les identifiants d'interface va seulement être justifié si assez d'adresses contiennent des identifiants non changeants pour en valoir la peine ; avoir un grand nombre de clients qui changent quotidiennement ou chaque semaine leur adresse est

probablement suffisant pour parer à la plupart des soucis de confidentialité.

Il y a aussi les coûts de client découlant d'un grand nombre d'adresses associées à un nœud (par exemple, en faisant des recherches d'adresse, au besoin de se joindre à un grand nombre de groupes de diffusion groupée, etc.). Donc, changer les adresses fréquemment (par exemple, toutes les quelques minutes) peut avoir des implications de performances.

Les nœuds qui suivent la présente spécification DEVRAIENT générer de nouvelles adresses temporaires de façon périodique. Cela peut se faire automatiquement en générant un nouvel identifiant d'interface aléatoire au moins une fois toutes les (TEMP_PREFERRED_LIFETIME - REGEN_ADVANCE - DESYNC_FACTOR) unités de temps. Comme décrit ci-dessus, générer une nouvelle adresse temporaire REGEN_ADVANCE unités de temps avant qu'une adresse temporaire devienne déconseillée produit des adresses avec une durée de vie préférée pas plus grande que TEMP_PREFERRED_LIFETIME. La valeur DESYNC_FACTOR est une valeur aléatoire (différente pour chaque client) qui assure que les clients ne sont pas synchronisés les uns avec les autres et ne génèrent pas de nouvelles adresses exactement au même moment. Quand la durée de vie préférée arrive à expiration, une nouvelle adresse temporaire DOIT être générée en utilisant le nouvel identifiant d'interface aléatoire.

Parce que la fréquence précise à laquelle il est approprié de générer de nouvelles adresses varie d'un environnement à l'autre, les mises en œuvre DEVRAIENT fournir aux utilisateurs finaux la capacité de changer la fréquence à laquelle les adresses sont regénérées. La valeur par défaut est donnée dans TEMP_PREFERRED_LIFETIME et est de un jour. De plus, le moment exact auquel invalider une adresse temporaire dépend de comment les applications sont utilisées par les utilisateurs finaux. Donc, la valeur par défaut suggérée de une semaine (TEMP_VALID_LIFETIME) peut n'être pas appropriée dans tous les environnements. Les mises en œuvre DEVRAIENT fournir aux utilisateurs finaux la capacité d'outrepasser ces deux valeurs par défaut.

Finalement, quand une interface se connecte à une nouvelle liaison, un nouvel identifiant d'interface aléatoire DEVRAIT être généré immédiatement avec un nouvel ensemble d'adresses temporaires. Si un appareil se déplace d'un ethernet à un autre, générer un nouvel ensemble d'adresses temporaires à partir d'un identifiant d'interface aléatoire différent assure que l'appareil utilise des identifiants d'interface aléatoires différents pour les adresses temporaires associées aux deux liaisons, rendant plus difficile de corréler les adresses provenant des deux liaisons différentes comme provenant du même nœud. Le nœud PEUT suivre tout processus disponible pour déterminer que le changement de liaison s'est produit. Un de ces processus est décrit par la détection du rattachement réseau [RFC4135].

3.6 Considérations de déploiement

Les appareils qui mettent en œuvre la présente spécification DOIVENT fournir à l'utilisateur final un moyen d'activer ou désactiver explicitement l'utilisation des adresses temporaires. De plus, un site pourrait souhaiter désactiver l'utilisation des adresses temporaires afin de simplifier le débogage et les opérations du réseau. Par conséquent, les mises en œuvre DEVRAIENT fournir un moyen pour que des administrateurs de système de confiance activent ou désactivent l'utilisation des adresses temporaires.

De plus, les sites pourraient souhaiter activer ou désactiver sélectivement l'utilisation des adresses temporaires pour certains préfixes. Par exemple, un site pourrait souhaiter désactiver la génération d'adresses temporaires pour des préfixes "uniques localement" [RFC4193] tout en générant encore des adresses temporaires pour tous les autres préfixes mondiaux. Un autre site pourrait souhaiter permettre la génération d'adresses temporaires seulement pour les préfixes 2001::/16 et 2002::/16, tout en la désactivant pour tous les autres préfixes. Pour prendre en charge ce comportement, les mises en œuvre DEVRAIENT fournir un moyen d'activer et désactiver la génération des adresses temporaires pour des sous gammes de préfixes spécifiques. Ce réglage par préfixe DEVRAIT outrepasser les réglages globaux sur le nœud par rapport aux sous gammes de préfixes spécifiées. Noter que le réglage par préfixe peut être appliqué à toute granularité, et pas nécessairement sur la base du sous réseau.

L'utilisation des adresses temporaires peut causer des difficultés inattendues avec certaines applications. Comme on le décrit ci-dessous, certains serveurs refusent d'accepter des communications provenant de clients pour lesquels ils ne peuvent pas transposer l'adresse IP en un nom DNS. De plus, certaines applications peuvent ne pas se comporter de façon robuste si des adresses temporaires sont utilisées et si une adresse expire avant que l'application soit terminée, ou si elles ouvrent plusieurs sessions, mais attendent d'elles que toutes utilisent les mêmes adresses. Par conséquent, l'utilisation d'adresses temporaires DEVRAIT être désactivée par défaut afin de minimiser les perturbations potentielles. Les applications individuelles, qui ont une connaissance spécifique de la durée normale des connexions, PEUVENT outrepasser cela comme approprié.

Si un très petit nombre de nœuds (disons, seulement un) utilise un certain préfixe pendant longtemps, changer juste la partie

identifiant d'interface de l'adresse peut n'être pas suffisant pour assurer la confidentialité, car le préfixe agit comme un identifiant constant. Les procédures décrites dans le présent document sont les plus efficaces quand le préfixe est raisonnablement non statique ou est utilisé par un très grand nombre de nœuds.

4. Implications du changement d'identifiant d'interface

L'architecture d'adressage IPv6 s'assure dans un certaine mesure que les identifiants d'interface vont probablement être uniques au monde quand il est facile de le faire. La large utilisation des adresses temporaires peut résulter en une fraction significative du trafic de l'Internet qui n'utilise pas des adresses dans lesquelles la portion identifiant d'interface est unique au monde. Par conséquent, l'usage des algorithmes du présent document peut compliquer la fourniture d'une telle souplesse future, si l'unicité globale est nécessaire.

Le désir de protéger la confidentialité individuelle peut entrer en conflit avec le désir de maintenir effectivement et déboguer un réseau. Avoir des clients qui utilisent des adresses qui changent dans le temps va rendre plus difficile de tracer et isoler les problèmes de fonctionnement. Par exemple, quand on cherche les traces d'un paquet, il pourrait devenir plus difficile de déterminer qu'un paquet montre un comportement causé par une seule machine défectueuse, ou par un certain nombre d'entre elles.

Certains serveurs refusent d'accorder l'accès à des clients pour lesquels il n'existe pas de nom DNS. C'est-à-dire, ils effectuent une interrogation PTR du DNS pour déterminer le nom DNS, et peuvent ensuite aussi effectuer une interrogation AAAA sur le nom retourné pour vérifier que le nom DNS retourné se transpose en l'adresse utilisée. Par conséquent, les clients qui ne sont pas proprement enregistrés dans le DNS peuvent être incapables d'accéder à certains services. Cependant, comme on l'a noté plus tôt, le nom DNS d'un nœud (si il ne change pas) sert d'identifiant constant. Le large déploiement de l'extension décrite dans le présent document pourrait mettre au défi la pratique de "l'authentification" fondée sur le DNS inverse qui a peu de validité, bien que largement mise en œuvre. Afin de répondre aux défis des serveurs, les nœuds pourraient enregistrer les adresses temporaires dans le DNS en utilisant des noms aléatoires (par exemple, une version sous forme de chaîne de l'adresse aléatoire elle-même).

L'utilisation des extensions définies dans le présent document peut compliquer le débogage et d'autres activités de réparation du fonctionnement. Par conséquent, la politique d'un site peut être que les adresses temporaires ne devraient pas être utilisées. En conséquence, les mises en œuvre DOIVENT fournir une méthode pour que l'utilisateur final ou un administrateur de confiance outrepasse l'utilisation d'adresses temporaires.

5. Constantes définies

Les constantes définies dans ce document incluent :

TEMP_VALID_LIFETIME -- valeur par défaut : 1 semaine. Les utilisateurs devraient être capables d'outrepasser la valeur par défaut.

TEMP_PREFERRED_LIFETIME -- valeur par défaut : 1 jour. Les utilisateurs devraient être capables d'outrepasser la valeur par défaut

REGEN_ADVANCE -- 5 secondes.

MAX_DESYNC_FACTOR -- 10 minutes. Limite supérieure de DESYNC_FACTOR.

DESYNC_FACTOR -- valeur aléatoire dans la gamme de 0 à MAX_DESYNC_FACTOR. Il est calculé une fois au démarrage du système (plutôt que chaque fois qu'il est utilisé) et ne doit jamais être supérieur à (TEMP_PREFERRED_LIFETIME - REGEN_ADVANCE).

TEMP_IDGEN_RETRIES -- valeur par défaut : 3

6. Travaux futurs

Une mise en œuvre pourrait vouloir garder trace des adresses qui sont utilisées par les couches supérieures afin d'être capable de supprimer une adresse temporaire déconseillée des structures de données internes quand aucun protocole de couche supérieure n'en utilise (mais pas avant). Ceci diffère des approches courantes où les adresses sont supprimées d'une interface quand elles deviennent invalides [RFC4862], indépendamment de si des protocoles de couche supérieure les utilisent encore ou non. Pour les connexions TCP, de telles informations sont disponibles dans les blocs de contrôle. Pour les applications fondées sur UDP, il peut se trouver que seules les applications aient connaissance des adresses actuellement utilisées. Par conséquent, une mise en œuvre va généralement avoir besoin d'utiliser une heuristique pour décider quand une adresse n'est plus utilisée.

La détermination d'utiliser des adresses publiques plutôt que temporaires peut dans certains cas seulement être faite par une application. Par exemple, certaines applications peuvent toujours vouloir utiliser des adresses temporaires, tandis que d'autres peuvent vouloir ne les utiliser que dans certaines circonstances ou pas du tout. Des extensions d'API convenables [RFC5014] vont probablement devoir être développées pour permettre à des applications individuelles d'indiquer avec une granularité suffisante leurs besoins à l'égard de l'utilisation des adresses temporaires. Des recommandations sur les pratiques du DNS pour éviter le problème décrit à la Section 4 quand les recherches inverses de DNS échouent peuvent être nécessaires. La [RFC4472] contient une discussion plus détaillée des questions relatives au DNS.

Bien que le présent document discute des façons de masquer l'adresse IP permanente d'un utilisateur, la méthode décrite est estimée être inefficace contre des formes sophistiquées d'analyse du trafic. Pour augmenter l'efficacité, on peut devoir envisager d'utiliser des techniques plus évoluées, comme l'acheminement en pelure d'oignon [ONION].

7. Considérations sur la sécurité

Le filtrage d'entrée a été et est déployé comme moyen d'empêcher l'utilisation d'adresses de source usurpées dans des attaques de déni de service distribuées (DDoS, *Distributed Denial of Service*). Dans un réseau avec un grand nombre de nœuds, de nouvelles adresses temporaires sont créées à un rythme élevé. Cela pourrait rendre difficile aux mécanismes de filtrage d'entrée de distinguer les changements légitimes d'adresses temporaires des adresses de source usurpées, qui sont "dans le préfixe" (en utilisant un préfixe topologiquement correct et un identifiant d'interface non existant). Cela peut être traité en utilisant des mécanismes de contrôle d'accès adresse par adresse sur le point de sortie de réseau.

8. Changements significatifs depuis la RFC 3041

Cette Section résume les changements apportés par le présent document à la RFC 3041 dont une mise en œuvre de la RFC 3041 devrait avoir connaissance.

1. Exclusion de certains identifiants d'interface de la gamme des identifiants d'interface acceptables. Les identifiants d'interface comme ceux réservés pour les adresses d'envoi à la cantonnade [RFC2526], etc.

2. Ajout d'un bouton de configuration qui fournit à l'utilisateur d'extrémité un moyen pour activer ou désactiver l'utilisation des adresses temporaires sur la nbase du préfixe.

3. Ajout d'une vérification pour les attaques de déni de service en utilisant de faibles durées de validité dans les annonces de routeur.

4. La DAD fonctionne maintenant sur toutes les adresses temporaires, pas juste la première générée à partir d'un identifiant d'interface.

5. Changement à être désactivé du réglage par défaut pour l'usage des adresses temporaires.

6. Il est maintenant permis au nœud de générer différents identifiants d'interface pour différents préfixes, si il le désire.

7. L'algorithme utilisé pour générer des identifiants d'interface aléatoires n'est plus restreint au seul MD5.

8. Le nombre d'essais par défaut est réduit à 3 et une variable de configuration a été ajoutée.

9. L'algorithme de traitement des annonces de routeur (RA) n'est plus inclus dans le document, et est remplacé par une référence à la [RFC4862].

9. Remerciements

Rich Draves et Thomas Narten étaient les auteurs de la RFC 3041. Ils tiennent à remercier de leurs contributions les membres du groupe de travail ipv6 et, en particulier, Ran Atkinson, Matt Crawford, Steve Deering, Allison Mankin, et Peter Bieringer.

Suresh Krishnan était le seul auteur de cette version du document. Il tient à remercier de leurs contributions le groupe de travail ipv6 et, en particulier, Jari Arkko, Pekka Nikander, Pekka Savola, Francis Dupont, Brian Haberman, Tatuya Jinmei, et Margaret Wasserman de leur commentaires détaillés.

10. Références

10.1 Références normatives

[RFC1321] R. Rivest, "Algorithme de résumé de message MD5", avril 1992, DOI 10.17487/RFC1321, *(Information)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. DOI 10.17487/RFC2119, *(MàJ par* RFC8174)

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006, DOI 10.17487/RFC4291. (*MàJ par* 5952 et 6052, 8064) *(D.S.)*

[RFC4861] T. Narten et autres, "Découverte du voisin pour IP version 6 (IPv6)", septembre 2007. (*Remplace* RFC2461) *(D.S. ; MàJ par* RFC8028, RFC8319, RFC8425, RFC9131)

[RFC4862] S. Thomson et autres, "Auto configuration d'adresse IPv6 sans état", septembre 2007. (*Remplace* RFC2462) *(D.S.)*

10.2 Références pour information

[ONION] Reed, MGR., Syverson, PFS., and DMG. Goldschlag, "Proxies for Anonymous Routing", Proceedings of the 12th Annual Computer Security Applications Conference, San Diego, CA, décembre 1996.

[RFC2131] R. Droms, "Protocole de configuration dynamique d'hôte", mars 1997. *(DS) (Mà J par* RFC3396, RFC4361, RFC5494, et RFC6849)

[RFC2526] D. Johnson, S. Deering, "Adresses réservées d'envoi à la cantonade de sous-réseau IPv6", mars 1999. *(P.S.)*

[RFC2965] D. Kristol, L. Montulli, "Mécanisme de gestion d'état HTTP", octobre 2000. *(P.S.)*

[RFC2136] P. Vixie, S. Thomson, Y. Rekhter et J. Bound, "Mises à jour dynamiques dans le système de noms de domaine (DNS UPDATE)", avril 1997, DOI 10.17487/RFC2136.

[RFC3315] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins et M. Carney, "Protocole de configuration dynamique d'hôte pour IPv6 (DHCPv6)", juillet 2003. (*MàJ par* RFC6422 et RFC6644, RFC7227 *; rendue obsolète par* RFC8415)

[RFC3484] R. Draves, "Choix d'adresse par défaut pour le protocole Internet version 6 (IPv6)", février 2003. (*Remplacée par la* RFC6724) *(P.S.)*

[RFC3972] T. Aura, "Adresses générées cryptographiquement (CGA)", mars 2005. (*MàJ par* RFC4581, RFC4982) *(P.S.)*

[RFC3972] T. Aura, "Adresses générées cryptographiquement (CGA)", mars 2005. (*MàJ par* RFC4581, RFC4982) *(P.S.)*

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005, DOI 10.17487/RFC4086, (*Remplace* RFC1750) (BCP0106)

[RFC4135] JH. Choi, G. Daley, "Objets de la détection de rattachements réseau dans IPv6", août 2005. *(Information)*

[RFC4193] R. Hinden, B. Haberman, "Adresses IPv6 en envoi individuel uniques localement", octobre 2005. *(P.S.)*

[RFC4472] A. Durand et autres, "Considérations et problèmes de fonctionnement du DNS IPv6", avril 2006. *(Information)*

[RFC5014] E. Nordmark et autres, "API de prises IPv6 pour la sélection d'adresse de source", septembre 2007. *(Information)*

Adresse des auteurs

Thomas Narten
IBM Corporation
P.O. Box 12195
Research Triangle Park, NC
USA
mél : narten@us.ibm.com

Richard Draves
Microsoft Research
One Microsoft Way
Redmond, WA
USA
mél : richdr@microsoft.com

Suresh Krishnan
Ericsson Research
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada
mél : suresh.krishnan@ericsson.com

Déclaration complète de droits de reproduction