Groupe de travail Réseau
Request for Comments : 3404
RFC rendues obsolètes : 2915, 2168
Catégorie : En cours de normalisation

M. Mealling
VeriSign
octobre 2002
Traduction Claude Brière de L'Isle

Système de découverte dynamique de délégation (DDDS) Partie IV : Application de résolution des identifiants de ressource uniformes (URI)

Statut de ce mémoire

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

Résumé

Le présent document décrit une spécification pour localiser un serveur d'autorité pour des informations sur des identifiants de ressource uniformes (URI, *Uniform Resource Identifier*). La méthode utilisée pour localiser ce serveur d'autorité est le système de découverte dynamique de délégation.

Le présent document fait partie d'une série qui est spécifiée dans "Système de découverte dynamique de délégation (DDDS) Partie I : DDDS complet" (RFC3401). Il est important de noter qu'il est impossible de lire et comprendre un des documents de cette série sans lire les autres.

La présente traduction incorpore les errata n° 282 et 787.

Table des Matières

1. Introduction

Le système de découverte dynamique de délégation (DDDS) est utilisé pour mettre en œuvre un lien lâche de chaînes en données, afin de prendre en charge les systèmes de délégation à configuration dynamique. Le DDDS fonctionne en transposant une chaîne unique en données mémorisées au sein d'une base de données DDDS en appliquant de façon itérative les règles de transformation de chaîne jusqu'à atteindre une condition de fin.

Le présent document décrit une application DDDS pour résoudre des identifiants de ressource uniformes (URI, *Uniform*

Resource Identifier). Il ne définit pas un algorithme ou base de données DDDS. La série entière des documents qui le font est spécifiée dans "Système de découverte dynamique de délégation (DDDS) Partie I : DDDS complet" [RFC3401]. Il faut noter qu'il est impossible de lire et comprendre un document de cette série sans lire tous les autres documents.

Les identifiants de ressource uniformes (URI) ont été une avancée significative dans la restitution de ressources accessibles sur l'Internet. Cependant, leur nature fragile au fil du temps a été reconnue depuis plusieurs années. Le groupe de travail "Uniform Resource Identifier" a proposé le développement de noms de ressource uniforme (URN, *Uniform Resource Name*) [RFC2141] pour servir d'identifiant persistent, indépendant de la localisation pour les ressources de l'Internet afin de surmonter la plupart des problèmes des URI. La [RFC1737] a établi les exigences pour les URN.

Pendant la durée de vie du groupe de travail URI, un certain nombre de propositions d'URN ont été faites. Les développeurs de plusieurs de ces propositions se sont rencontrés dans une séries de réunions, d'où est résulté un compromis connu sous le nom de cadre de Knoxville. Le principe majeur qui sous-tend le cadre de Knoxville est que le système de résolution doit être séparé de la façon dont les noms sont alloués. C'est un contraste marqué avec la plupart des URI, qui identifient l'hôte à contacter et le protocole à utiliser. Le lecteurs se reportera à [Arms] pour voir les fondements du cadre de Knoxville et toutes les informations supplémentaires sur le contexte et les objectifs de cette proposition.

Séparer la façon dont les noms sont résolus de la façon dont ils sont construits procure plusieurs avantages. Cela permet de mettre plusieurs approches de dénomination et plusieurs approches de résolution en concurrence, car cela autorise l'utilisation de différents protocoles et résolveurs. Une telle séparation pose juste un problème – comment résoudre un nom lorsque il ne peut pas nous donner d'indication sur son résolveur ?

Pour le court terme, le système des noms de domaine (DNS) est le candidat évident pour le cadre de résolution, car il est largement déployé et compris. Cependant, il n'est pas approprié d'utiliser le DNS pour conserver des informations sur la base de la ressource. Tout d'abord, le DNS n'a jamais eté destiné à traiter autant d'enregistrements. Ensuite, la taille limitée des enregistrement est inapproprié pour des informations de catalogue. Enfin, les noms de domaines ne sont pas appropriés comme URN.

Donc notre approche est d'utiliser le DDDS pour localiser les "résolveurs" qui peuvent fournir des informations sur les ressources individuelles, incluant potentiellement la ressource elle-même. Pour réaliser cela, on "réécrit" l'URI dans une clé suivant les règles de DDDS. Le présent document décrit la résolution d'URI comme une application du DDDS et spécifie l'utilisation d'au moins une base de données fondée sur le DNS.

2. Terminologie

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

Tous les termes en majuscules sont tirés du vocabulaire qui se trouve dans la spécification de l'algorithme DDDS de la [RFC3403].

3. Distinction entre URN et URI

Du point de vue de ce système, il n'y a pas de différence théorique entre la résolution des URI dans le cas général et celle des URN dans ce cas spécifique. Du point de vue du fonctionnement, il y a cependant une différence qui vient du fait que la résolution d'URI pourrait éventuellement n'être pas d'un usage très répandu. Si la résolution d'URN est fondue dans la résolution générique d'URI, les URN pourraient souffrir du manque d'adoption de la résolution d'URI.

La solution est de permettre de court-circuiter la résolution d'URN. Dans la présente spécification, la résolution générique d'URI commence par l'insertion de règles pour les schémas d'URI connus dans le registre "uri.arpa.". Pour le schéma d'URI "URN:", une des règles trouvées dans 'uri.arpa.' sera pour le schéma d'URI "urn". Cette règle va simplement déléguer à la zone "urn.arpa." les pointeurs d'autorité de dénomination (NAPTR) supplémentaires fondés sur l'espace de nom d'URN. Essentiellement, la règle de réécriture de la résolution d'URI pour "URN:" est la première règle bien connue de l'application de résolution d'URN.

Donc, le présent document spécifie deux applications DDDS. L'une est pour la résolution d'URI et l'autre est pour la résolution d'URN. Toutes deux sont techniquement identiques mais en les séparant; la résolution d'URN peut toujours être traitée de façon indépendante.

4. Spécifications d'application de résolution d'URI et d'URN

Ce schéma définit l'application DDDS de résolution d'URI et d'URN conformément aux règles et exigences qu'on trouve dans la [RFC3403]. La base de données DDDS utilisée par cette application se trouve dans la [RFC3403] qui est le document qui définit le type d'enregistrement de ressource (RR) du DNS Pointeur d'autorité de désignation (NAPTR, *Naming Authority Pointer*).

4.1 Chaîne d'application unique

La chaîne unique d'application est l'URI ou l'URN pour lequel un serveur d'autorité est localisé. Cet URI ou URN DOIT être canonisé et codé en hexadécimal conformément à la production de "uri absolu" qu'on trouve dans l'ANBF collecté dans la [RFC2396].

4.2 Première règle bien connue

Dans le cas de l'URI, la première règle connue est créée en prenant le schéma d'URI. Dans le cés de l'URN, la première règle connue est l'identifiant d'espace de nom. Par exemple, l'URI "http://www.example.com/" aurait une clé "http". L'URN "urn:foo:foospace" aurait comme première clé "foo".

4.3 Fanions

Pour le moment, seuls quatre fanions, "S", "A", "U", et "P", sont définis. Les fanions "S", "A" et "U" sont pour une recherche terminale. Cela signifie que la règle est la dernière et que le fanion détermine quelle devrait être la dernière étape. Le fanion "S" signifie que le résultat de cette règle est un nom de domaine pour lequel un ou plluisuers enregistrments SRV [RFC2782] existent. Voir à la Section 5 des informations complémentaires sur la façon dont la résolution d'URI et d'URN utilise le type d'enregistrement SRV. Un "A" signifie que le résultat de la règle est un nom de domaine et devrait être utilisé pour rechercher des enregistrements A, AAAA, ou A6 pour ce domaine. Le fanion "U" signifie que le résultat de la règle est un URI [RFC2396].

Le fanion "P" dit que le reste de l'algorithme DDDS est ignoré et que le reste du processus est spécifique de l'application et sort du domaine d'application du présent document. Une application peut utiliser la partie Protocole qui se trouve dans le champ Services pour identifier quel ensemble de règles spéciques de l'application devrait être suivi ensuite. L'enregistrement qui contient le fanion 'P' est le dernier enregistrement qui est interprété par les règles dans ce document. On peut penser que cela ferait aussi du fanion "P" un indicateur d'une recherche terminale, mais cela serait incorrect car une règle "terminale" est un concept du DDDS et ce fanion indique que tout ce qu'il y a après cette règle n'adhère pas du tout au concepts du DDDS.

Les fanions alphabétiques restants sont réservés pour de futures versions de cette spécification. Les fanions numériques peuvent être utilisés pour une expérimentation locale. Les fanions S, A, U et P s'excluent mutuellement, et les bibliothèques de résolution PEUVENT signaler une erreur si il en est donné plus d'un. (Le code expérimental et le code pour aider à la création des règles de réécriture signaleront plus vraisemblablement une telle erreur qu'un client comme un navigateur.) On prévoit que plusieurs fanions seront permis à l'avenir, de sorte que les mises en œuvre NE DOIVENT PAS supposer que le champ Fanions ne peut contenir que des caractères 0 ou 1. Finalement, si un client rencontre un enregistrement avec un fanion inconnu, il DOIT l'ignorer et passer à la règle suivante. Cette vérification prend le pas sur tout ordre dans la mesure où les fanons peuvent contrôler l'interprétation des champs. Un nouveau fanion peut changer l'interprétation des champs "regexp" et/ou Remplacement de façon qu'il est impossible de déterminer si un enregistrement correspond à une certaine cible.

Les fanions "S", "A", et "U" sont appelés des fanions "terminaux" car ils mettent un terme aux boucles de l'algorithme DDDS. Si ces fanions ne sont pas présents, les clients peuvent supposer qu'il existe une autre règle à la cle produite par la règle de réécriture actuelle.

4.4 Paramètres de service

Les paramètres de service pour cette application prennent la forme d'une chaîne de caractères qui suivent cet ABNF :

```
champ_service = [ [protocole] *("+" rs)]
protocole     = ALPHA *31ALPHANUM
rs            = ALPHA *31ALPHANUM   ; Les champs Protocole et rs sont limités à 32 caractères et doivent
                                    commencer par un caractère alphabétique.
```

En d'autres termes, une spécification de protocole facultative suivie par 0, un ou plusieurs services de résolution. Chaque service de résolution est indiqué par un carctère '+' initial.

La chaîne vide est aussi valide. On verra normalement cela au début d'une série de règles, quand il est impossible de savoir quels services et protocoles seront offerts à la fin d'un chemin de délégation particulier.

4.4.1 Services

Les identifiants de service qui constituent la production "rs" sont génériques pour la résolution d'URI comme d'URN car la valeur d'entrée se fait sur la base du schéma d'URI. La liste des services valides est définie dans la [RFC2483].

Voici quelques exemples de ces services :

I2L : un URI donné retourne un URI qui identifie une localisation où on peut retrouver l'URI original.
I2Ls : un URI donné retourne un ou plusieurs URI qui identifient plusieurs localisations où l'URI original se retrouve.
I2R : un URI donné retourne une instance de la ressource identifiée par cet URI.
I2Rs : un URI donné retourne une ou plusieurs instances de la ressources identifiée par cet URI.
I2C : un URI donné retourne une instance d'une description de cette ressource.
I2N : un URI donné retourne un URN qui nomme la ressource (Attention : l'égalité par rapport aux URN n 'est pas triviale. Voir la [RFC1737] qui explique pourquoi.)

4.4.2 Protocoles

Les identifiants de protocole qui sont valides pour la production "protocole" DOIVENT être définis par des documents qui sont spécifiques de la résolution d'URI. À présent, le protocole HTTP [RFC2169] est la seule spécification de cette sorte.

Il est extrêmement important de réaliser que la simple spécification d'un protocole dans le champ services est insuffisante car il y a une sémantique supplémentaire qui entoure la résolution d'URI et qui n'est pas définie dans les protocoles. Par exemple, si Z39.50 devait être spécifié comme protocole valide, il faudrait définir de plus comment il va coder les demandes de services spécifiques, comment l'URI est codé, et quelles informations sont retournées.

4.4.3 Applicabilité des services

Comme il est possible qu'il y ait un ensemble complexe de protocoles et services possibles, une application cliente peut souvent avoir besoin d'appliquer un processus plus complexe de prise de décision à un ensemble d'enregistrements que de simplement faire correspondre une liste ordonnée de protocoles. Par exemple, si il y a quatre règles qui sont applicables, la dernière peut avoir un champ Service plus désirable que la première. Mais comme le client peut être satisfait par la première il ne saura jamais que la quatrième pourrait être "meilleure".

Pour atténuer cela, le client peut vouloir modifier légèrement l'algorithme DDDS (pour cetre seule application !) afin de déterminer si il existe des protocoles/servicesf plus applicables. Cela peut être fait en toute sécurité pour cette application en utilisant une interaction plus complexe entre les étapes 3 et 4 de l'algorithme DDDS afin de trouver le chemin optimal à suivre. Par exemple, une fois qu'un client a trouvé une règlele dont l'expression de substitution produit un résultat et dont la description de service est acceptable, il peut le noter mais continuer à chercher d'autres règles qui s'appliquent (tout en respectant l'ordre !) afin d'en trouver une meilleure. Si il ne s'en trouve pas, il peut utiliser celle dont il a pris note.

Il faut garder en mémoire que pour que ceci reste sûr, l'entrée de l'étape 3 et le résultat de l'étape 4 DOIVENT être identiques à l'algorithme de base. Le logiciel client NE DOIT PAS essayer de faire cette optimisation en-dehors d'un ensemble spécifique de règles de réécriture (c'est-à-dire, à travers des chemins de délégation).

4.5 Bases de données valides

Pour l'instant, une seule base de données DDDS est spécifiée pour cette application. "Système de découverte dynamique de délégation (DDDS) Partie III : base de données du système de noms de domaines (DNS)" [RFC3403] spécifie une base de données DDDS qui utilise l'enregistrement de ressource NAPTR du DNS pour conserver les règles de réécriture. Les clés pour cette base de données sont codées comme noms de domaines.

Le résultat de la première règle bien connue pour l'application de résolution d'URI est le schéma d'URI. Afin de convertir cela en une clé unique dans cette base de données, la chaîne ".uri.arpa." est ajoutée à la fin. Ce nom de domaine est utilisé pour demander des enregistrements NAPTR qui produisent de nouvelles clés sous forme de noms de domaines.

Le résultat de la première règle bien connue de l'application de résolution d'URN est l'espace de noms d'URN "id". Pour

convertir cela en une clé unique dans cette base de données la chaîne ".urn.arpa." est ajoutée à la fin. Ce nom de domaine est utilisé pour demander des enregistrements NAPTR qui produisent de nouvelles clés sous la forme de noms de domaines.

Les serveurs du DNS PEUVENT interpréter les valeurs de fanions et utiliser ces informations pour inclure les enregistrements SRV et A appropriés dans la portion d'informations supplémentaires du paquet DNS. Les clients sont invités à vérifier les informations supplémentaires mais ne sont pas obligés de le faire. Voir la section sur le traitement des informations supplémentaires dans la [RFC3403] pour des informations complémentaires sur les enregistrements NAPTR et dans la section Informations supplémentaires du paquet de réponse du DNS.

Le jeu de caractères utilisé pour coder l'expression de substitution est l'UTF-8. Les caractères d'entrée admis sont tous les caractères qui sont admis partout dans un URI. Les caractères admis pour être dans une clé sont ceux qui sont actuellement définis pour les noms de domaines du DNS. Le fanion "i" à l'expression de substitution est utilisé pour noter que, lorsque c'est approprié pour le codet en question, toutes les correspondances devraient être faites de façon insensible à la casse.

5. Exemples

5.1 Exemple avec un URN

Considérons un URN qui utilis l'espace de noms hypothétique FOO. Les numéros FOO sont des identifiants pour environ 30 millions d'entreprises enregistrées tout autour du monde, alloués et gérés par Fred, Otto et Orvil, SA. L'URN pourrait ressembler à :

urn:foo:002372413:annual-report-1997

La première étape du processus de résolution est de trouver l'espace de nom de FOO. L'identifiant de l'espace de noms [RFC2141], "foo", est extrait de l'URN et ajouté devant ".urn.arpa.", ce qui produit "foo.urn.arpa.". Le DNS est interrogé sur les enregistrements NAPTR pour ce domaine, ce qui produit les résultats suivants :

foo.urn.arpa.

;;	ordre	pref	fanions	service	regexp	remplacement
IN NAPTR	100	10	"s"	"foolink+I2L+I2C"	""	foolink.udp.example.com.
IN NAPTR	100	20	"s"	"rcds+I2C"	""	rcds.udp.example.com.
IN NAPTR	100	30	"s"	"thttp+I2L+I2C+I2R"	""	thttp.tcp.example.com.

Le champ ordre contient des valeurs égales, ce qui indique qu'aucun ordre n'est à suivre. Le champ préférence indique que le fournisseur aimerait que les clients utilisent le protocole "foolink" particulier, suivi par le protocole RCDS, et que THTTP soit offert en dernier recours. Tous les enregistrements spécifient le fanion "s" qui signifie que l'enregistrement est terminal et que la prochaine étape est de restituer un enregistrement SRV du DNS pour le nom de domaine en question.

Le champ Service dit que si on parle de foolink, on doit être capable de produire des demandes I2L, I2C ou I2R pour obtenir un URI ou demander quelque chose de compliqué au sujet de la ressource. Le service de catalogue et de distribution de ressources (RCDS, *Resource Cataloging and Distribution Service*) [Catalog] pourrait être utilisé pour obtenir des métadonnées sur la ressource, tandis que THTTP pourrait être utilisé pour obtenir un URI pour la localisation actuelle de la ressource.

En supposant que notre client ne connait pas le protocole foolink mais connait le protocole RCDS, notre prochaine action est de rechercher des RR SRV pour "rcds.udp.example.com", ce qui nous dira les hôtes qui peuvent fournir le service de résolution nécessaire. Cette rechercher pourrait retourner :

;;		Préf	Pondération	Accès	Cible
rcds.udp.example.com	IN SRV	0	0	1000	deffoo.example.com.
	IN SRV	0	0	1000	dbexample.com.au.
	IN SRV	0	0	1000	ukexample.com.uk.

nous donnant trois hôtes qui pourrait bien faire la résolution, et nous indiquant l'accès qu'on devrait utiliser pour communiquer avec leur serveur RCDS. (Le lecteur se reportera à la spécification de SRV [RFC2782] pour l'interprétation des champs ci-dessus.)

Il y a ici l'opportunité d'une optimisation significative. La RFC3403 définit que la section des informations supplémentaires peut être disponible. Dans ce cas, les enregistrements SRV peuvent être retournés comme informations supplémentaires

pour des recherches de NAPTR terminaux (ainsi que d'enregistrements A pour ces SRV). C'est une optimisation significative. En conjonction avec un long TTL pour les enregistrements "*.urn.arpa.", le nombre moyen d'interrogations au DNS pour résoudre la plupart des URI serait proche de un.

Noter que l'exemple des enregistrements NAPTR ci-dessus est destiné à représenter le résultat d'une recherche de NAPTR en utilisant un logiciel client du genre de "nslookup" ; les administrateurs de zone devraient consulter la documentation qui accompagne leurs serveurs de noms de domaines pour vérifier la syntaxe précise qu'ils devraient utiliser pour les fichiers de zone.

Noter aussi qu'il aurait pu y avoir une première étape supplémentaire où l'URN serait résolu comme URI générique par une recherche sur "urn.uri.arpa". La règle résultante aurait spécifié que le NID étant extrait de l'URN et ".urn.arpa." y étant ajouté résulterait en une nouvelle clé "foo.urn.arpa." qui est la première étape à partir de ci-dessus.

5.2 Exemple de schéma d'URI CID

Considérons un schéma d'URI fondé sur les identifiants de contenu MIME. L'URI pourrait ressembler à ceci :

cid:199606121851.1@bar.example.com

(Noter que cet exemple est choisi à des fins pédagogiques, et ne se conforme pas au schéma d'URI CID.)

Le première étape du processus de résolution est de trouver ce qu'il en est du schéma CID. Le schéma est extrait de l'URI, ajouté à ".uri.arpa.", et le NAPTR pour "cid.uri.arpa." cherché dans le DNS. Il pourrait retouner un enregistrement de la forme :

cid.uri.arpa.

;;	ordre	préf	fanions	service	regexp	remplacement
IN NAPTR	100	10	""	""	"!^cid:.+@([^\.]+\.)(.*)$!\2!i"	.

Comme il n'y a qu'un seul enregistrement, l'ordre des réponses n'est pas un problème. Le champ Remplacement est vide, de sorte que le schéma fourni dans le champ Regexp est utilisé. On applique ce regexp à l'URI entier pour voir si il correspond, ce qu'il fait. La partie \2 de l'expression de substitution retourne la chaîne "example.com". Comme le champ Fanion est vide, la recherche n'est pas terminale et notre prochain essai sur le DNS est pour trouver plus d'enregistrements NAPTR où le nouveau domaine est "example.com".

Noter que la règle n'extrait pas le nom de domaine complet du CID, elle suppose plutôt que le CID vient d'un hôte et extrait son domaine. Bien que tous les hôtes, comme "bar", puissent avoir leur propre NAPTR, la conservation de tous ces enregistrements pour toutes les machines sur un site pourrait constituer une charge intolérable. Les caractères génériques ne sont pas appropriés ici car ils ne retournent des résultats que lorsque il n'y a pas de nom correspondant exactement déjà dans le système.

L'enregistrement retourné de l'interrogation sur "example.com" pourrait ressembler à ceci :

example.com.

;;	ordre	préf	fanions	service	regexp	remplacement
IN NAPTR	100	50	"s"	"z3950+I2L+I2C"	""	z3950.tcp.example.com.
IN NAPTR	100	50	"s"	"rescap+I2C"	""	rescap.udp.example.com.
IN NAPTR	100	50	"s"	"thttp+I2L+I2C+I2R"	""	thttp.tcp.example.com.

En continuant avec cet exemple, on note que les valeurs des champs Ordre sont égaux pour tous les enregistrements, de sorte que le client est libre de prendre n'importe lequel. L'application définit le fanion "s" comme signifiant une recherche terminale et que le résultat de la réécriture sera un nom de domaine pour lequel un enregistrement SRV devrait être demandé. Une fois que le client a fait cela, il a les informations suivantes : l'hôte, l'accès, le protocole, et les services disponibles via ce protocole. Ces bits d'informations étant donnés, le client en a assez pour être capable de contacter ce serveur et poser ses questions sur l'URI cid.

On se rappelle que l'expression régulière utilisait \2 pour extraire un nom de domaine du CID, et \. pour faire correspondre les caractères littéraux '.' qui séparent les composants du nom de domaine. Comme '\' est le caractère d'échappement, les occurrences littérales de barre oblique inverse doivent être transformées par une autre barre oblique inverse. Pour le cas de l'enregistrement "cid.uri.arpa" ci-dessus, l'expression régulière entrée dans le fichier maître devrait être "!^cid:.+@([^\\.]+\\.)(.*)$!\\2!i". Lorsque le code client reçoit en fait l'enregistrement, le schéma aura été converti en "!^cid:.+@([^\.]+\.)(.*)$!\2!i".

5.3 Résolution d'un schéma d'URI HTTP

Même si les systèmes d'URN étaient maintenant en place, il y aurait encore un nombre considérable d'hôtes fondés sur des URI. Il devrait être possible de développer un système de résolution d'URI qui puisse aussi fournir l'indépendance de la localisation pour ces URI.

En supposant que nouc avons l'URI pour un logiciel très populaire que l'éditeur souhaite refléter sur de nombreux sites du monde entier :

http://www.example.com/software/latest-beta.exe

On extrait le prefixe "http", et on fait une recherche des enregistrements NAPTR pour "http.uri.arpa.". Cela pourrait retourner un enregistrement de la forme :

```
http.uri.arpa. IN NAPTR
;;     ordre    préf    fanions    service          regexp                    remplacement
         100      90       ""         ""      "!^http://([^/:]+)!1!i"              .
```

Cette expression retourne tout ce qu'il y a après la première double barre oblique et avant la prochaine barre oblique ou deux points. On utilise le caractère '!' pour délimiter les parties de l'expression de substitution. Autrement, on devrait utiliser les barres obliques inverses pour l'échappement des barres obliques, et on aurait une regexp dans le fichier zone qui ressemblerait à ceci : "/^http:\\/\\/([^\\/:]+)/\\1/i").

Appliquer ce schéma à l'URI extrait "www.example.com". La recherche d'enregistrements NAPTR pour cela retournerait :

```
www.example.com.
     ;;          ordre    préf    fanions      service          regexp          remplacement
IN NAPTR         100      100       "s"      "thttp+L2R"         ""        thttp.example.com.
IN NAPTR         100      100       "s"       "ftp+L2R"          ""             ftp.example.com.
```

La recherche d'enregistrements SRV pour thttp.example.com retournerait des informations sur les hôtes que example.com a désigné pour être ses sites miroir. Le client peut alors en prendre un pour l'usager.

6. Notes

- Les procédures d'enregistrement pour les zones "urn.arpa." et "uri.arpa." du DNS sont spécifiées dans "Système de découverte dynamique de délégation (DDDS) Partie V : Procédures d'allocation de URI.ARPA" [RFC3405].
- Si un enregistrement à un ordre particulier correspond à l'URI, mais si le client ne connaît pas le protocole et service spécifiés, le client DEVRAIT continuer l'examen des enregistrements qui ont le même ordre. Le client NE DOIT PAS prendre en considération les enregsitrements qui ont une valeur d'ordre supérieure. Ceci est nécessaire pour faire fonctionner la délégation de portions de l'espace de noms. Le champ Ordre est ce qui permet aux administrateurs de site de dire "toutes les demandes pour les URI qui correspondent au schéma "x" vont au serveur 1, toutes les autres vont au serveur 2".
- Noter que les RR SRV imposent des exigences supplémentaires aux clients.

7. Considérations relatives à l'IANA

L'utilisation des zones "urn.arpa." et "uri.arpa." exige que les politiques et procédures d'enregistrement soient suivies pour que le fonctionnement de ces zones du DNS soit préservé. Ces politiques et procédures sont précisées dans le document "Système de découverte dynamique de délégation (DDDS) Partie V : Procédures d'allocation de URI.ARPA" [RFC3405]. Le fonctionnement de ces zones impose des responsabilités opérationnelles et administrative à l'IANA.

La méthode d'enregistrement utilisée pour les valeurs dans les champs Services et Fanions est qu'une spécification soit approuvée par l'IESG et publiée comme une RFC pour information ou en voie de normalisation.

Les politiques d'enregistremnt pour les URI se trouvent dans la [RFC2717]. Les politiques d'enregistremnt de URN NID se trouvent dans la [RFC2611].

8. Considérations pour la sécurité

L'utilisation de "urn.arpa." et "uri.arpa." comme registre des espaces de nom est sujette à des attaques de déni de service, ainsi que des autres attaques de parodie du DNS. Les interactions avec DNSSEC sont actuellement en cours d'étude. On prévoit que les enregistrements NAPTR seront signés avec les enregistrements SIG une fois que les résultats de DNSSEC seront mis en place.

Les règles de réécritures rendent les identifiants provenant d'autres espaces de noms sujets aux mêmes attaques que les noms de domaine normaux. Comme elles n'ont pas été facilement résolvables auparavant, cela peut être considéré ou non comme un problème.

Les expressions régulières devraient subir une vérification de bonne santé, et non passées aveuglément à quelque chose comme PERL.

Le présent document a exposé un moyen pour localise un résolveur, mais n'a pas discuté les détails de la façon dont a lieu la communication avec le résolveur. Des considération de sécurité significatives se rattachent à la communication avec un résolveur. Ces considérations sortent du domaine d'application de ce document, et doivent être traitées par les spécifications des protocoles de communication des résolveurs particuliers.

9. Remerciements

Les éditeurs tiennent à remercier Keith Moore pour toutes ses consultations durant le développement de ce document. Nous remercions également Paul Vixie de son assistance au débogage de notre mise en œuvre, et de ses réponses à nos questions. Finalement nous aimerions reconnaître notre immense dette intellectuelle à l'égard de tous les participants à la série de réunions de Knoxville, ainsi qu'aux participants aux groupes de travail URI et URN.

Une reconnaissance particulière est due à Ron Daniel qui était coauteur des versions originales de ces documents. Ses premières mises en œuvre et la précision de sa pensée ont été précieuses pour éclairer de nombreux cas limites potentiels.

Références

[Arms] B. Arms, "The URN Implementors, Uniform Resource Names: A Progress Report", D-Lib Magazine, févier 1996.

[Catalog] Moore, K., Browne, S., Cox, J. and J. Gettler, "Resource Cataloging and Distribution System", Rapport technique CS-97-346, décembre 1996.

[RFC1737] K. Sollins et L. Masinter, "Exigences fonctionnelles pour les noms de ressource uniformes", décembre 1994.

[RFC2141] R. Moats, "Syntaxe des URN", mai 1997.

[RFC2168] R. Daniel, M. Mealling, "Résolution des identifiants de ressource uniformes avec le système des noms de domaines", juin 1997. (*Obsolète, voir* RFC3401, RFC3402, RFC3403, RFC3404) (*MàJ par* RFC2915) *(Exp.)*

[RFC2169] R. Daniel, "Convention triviale pour l'utilisation de HTTP dans la résolution d'URN", juin 1997. *(Exp.)*

[RFC2276] K. Sollins, "Principes d'architecture de la résolution de nom de ressource uniforme", janvier 1998. (*MàJ par* RFC3401) *(Information)*

[RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir* RFC3986)

[RFC2483] M. Mealling et R. Daniel, "Services de résolution d'URI nécessaires pour la résolution d'URN", janvier 1999.

[RFC2611] L. Daigle et autres, "Mécanismes de définition d'espace de nom d'URN", juin 1999. (*Obsolète, voir* RFC3406) (BCP0033)

[RFC2717] R. Petke, I. King, "Procédures d'enregistrement des noms de schéma d'URL", novembre 1999. (*Obsolète, voir* RFC4395) (BCP0035)*)*

[RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "RR DNS pour la spécification de la localisation des services (DNS SRV)", février 2000.

[RFC2915] M. Mealling, R. Daniel, "Enregistrement de ressource DNS Pointeur d'autorité de dénomination (NAPTR)", septembre 2000. (*Obsolète, voir* RFC3401, RFC3402, RFC3403, RFC3404) *(P.S.)*

[RFC3401] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie I : DDDS complet", octobre 2002. *(Info.)*

[RFC3402] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie II : l'algorithme",

octobre 2002. *(P.S.)*

[RFC3403] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie III : base de données du système de noms de domaines (DNS)", octobre 2002. *(P.S.)*

[RFC3405] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie V : Procédures d'allocation de URI.ARPA", octobre 2002. (BCP0065)

Appendice A Pseudocode

On donne ci-dessous pour l'édification des développeurs le pseudocode pour un sous programme client qui utilise les NAPTR. Ce code est donné à titre de simple information et n'est en aucune façon une norme du traitement des enregistrements NAPTR. On doit avertir le lecteur de ce que comme c'est aussi le cas avec le pseudocode, il n'a jamais été exécuté et peut contenir des erreurs logiques.

```
//
// findResolver(URN)
// Étant donné un URN, trouver un hôte qui puisse le résoudre.
//
findResolver(chaîne d'URN) {
/ ajouter le préfixe à ".urn.arpa."
  sprintf(clé, "%s.urn.arpa.", extraireNS(URN));
  faire {
    fanion_de_réécriture = faux ;
    terminal = faux ;
    si (une clé est vue) {
      quitter avec une signalisation de détection de boucle
    }
ajouter la clé à la liste des enregistrements "vus" = recherche(type=NAPTR, clé); // prendre tous les RR NAPTR pour "clé'
éliminer tout enregistrement avec une valeur inconnue dans le champ "fanions".
trier les enregistrements NAPTR par champ "ordre" et "préférence" ("ordre" étant plus significatif que "préférence").
   n_naptrs = nombre d'enregistrements NAPTR dans la réponse.
   curr_order = records[0].order;
   max_order = records[n_naptrs-1].order;
// Traiter le lot actuel de NAPTR selon le champ "ordre".
     pour (j=0; j < n_naptrs && records[j].order <= max_order; j++) {
     si (fanion_inconnu) // sauter cet enregistrement et passer au suivant ;
        nouvelle_clé = réécrituree(URN, naptr[j].remplacement, naptr[j].regexp);
       si (!nouvelle_clé) // Sauter à l'enregistrement suivant si la réécriture ne correspondait pas au suivant ;
// Faire une réécriture, réduire max_order à la valeur actuelle de façon à ce que la délégation fonctionne correctement.
      max_order = naptr[j].order;
// Sait-on que faire avec les protocoles et services spécifiés dans le NAPTR ? Sinon, essayer le prochain enregistrement.
     si (!isKnownProto(naptr[j].services)) {
          continuer ;
        }
     si (!isKnownService(naptr[j].services)) {
          continuer ;
        }

// À ce moment, on a une réécriture réussie et on sait comment parler au protocole ;  on demande un service de résolution
connu. Avant de faire la prochaine recherche, vérifier les fanions pour voir si on est rendu.
// Note : Il est possible de réécrire cela de façon que cet engegistrement valide soit noté comme tel mais qu'on continue afin
de trouver un "meilleur" enregistrement. Mais ce code serait trop volumineux et trop spécifique des applications pour être
illustré ici.

     si (strcasecmp(fanios, "S")
         || strcasecmp(fanions, "P"))
         || strcasecmp(fanions, "A")) {
          terminal = vrai;
          services = naptr[j].services;
          addnl = tout enregisrement SRV et/ou A retourné comme information supplémentaire pour naptr[j].
        }
        clé = nouvelleclé;
```

```
        rewriteflag = vrai;
        break;
       }
     } alors que (rewriteflag && !terminal);

// A t-on trouvé un chemin vers un résolveur ?
    si (!rewrite_flag) {
       rapporter une erreur
       retourner NUL;
     }

// Passer à un autre protocole ?
    si (strcasecmp(fanions, "P")) {
       retourner la clé  comme hôte à qui parler
     }
// Sinon, rester branché
    si (!addnl) { // Aucun SRV n'est venu au titre des informations supplémentaires, les rechercher.
       srvs = recherche(type=SRV, clé);
     }

     trier les enregistrements SRV par préférence, poids, ... pour chaque (enregistrement SRV)
{ // dans l'ordre de préférence essayer de contacter srv[j].target en utilisant le protocole et une des demandes de service de
résolution à partir du champ "services" du dernier enregistrement NAPTR.
      si (réussite)
        retourner (cible, protocole, service);
// En fait, on va probablement retourner un résultat, mais ce code est juste supposé nous donner un bon hôte à qui parler.
     }
     terminer avec une erreur "incapable de trouver un hôte" ;
}
```

Adresse de l'auteur

Michael Mealling
VeriSign
21345 Ridgetop Circle
Sterling, VA 20166
USA
mél : michael@neonym.net
URI : http://www.verisignlabs.com

Déclaration complète de droits de reproduction

Remerciement

Le financement de la fonction d’édition des RFC est actuellement fourni par la Internet Society.