

Groupe de travail Réseau
Request for Comments : 2169
Catégorie : Expérimentale
Traduction Claude Brière de L'Isle

R. Daniel
Los Alamos National Laboratory
juin 1997

Convention triviale pour l'utilisation de HTTP dans la résolution d'URN

Statut de ce mémoire

Le présent mémoire définit un protocole expérimental pour la communauté de l'Internet. Il ne spécifie pas une norme de l'Internet. Des discussions et suggestions sont demandées pour son amélioration. La diffusion du présent mémoire n'est soumise à aucune restriction.

Résumé

Le groupe de travail Noms de ressources uniformes (URN-WG, *Uniform Resource Names Working Group*) a été constitué pour spécifier des noms persistants, indépendants de la localisation pour les ressources accessibles par le réseau, ainsi que des mécanismes de résolution pour restituer les ressources étant donné ce nom. Le groupe de travail URN examine actuellement un mécanisme de résolution particulier, la proposition NAPTR [RFC2168]. Cette proposition spécifie comment un client peut trouver un "résolveur" pour un URN. Un résolveur est une base de données qui peut fournir des informations sur la ressource identifiée par un URN, comme la localisation de la ressource, une description bibliographique, ou même la ressource elle-même. Le protocole utilisé pour que le client communique avec le résolveur n'est pas spécifié dans la proposition NAPTR. L'enregistrement de ressource NAPTR fournit plutôt un champ qui indique le "protocole de résolution" et les "demandes de service de résolution" offertes par le résolveur.

Le présent document spécifie le protocole de résolution "THTTP" – une convention triviale pour coder les demandes et réponses de service de résolution comme du HTTP 1.0 ou comme du HTTP 1.1. L'objectif principal de THTTP est d'être simple à mettre en œuvre de façon que les serveurs HTTP existants puissent facilement ajouter la prise en charge de la résolution d'URN. On s'attend à ce que les bases de données utilisées par les premiers résolveurs soient utiles lorsque des protocoles de résolution plus sophistiqués seront ultérieurement développés.

1. Introduction

La spécification NAPTR [RFC2168] définit un nouvel enregistrement de ressource du DNS qui peut être utilisé pour découvrir les résolveurs pour les identifiants de ressource uniforme. Cet enregistrement de ressource présente le champ "services" pour spécifier le "protocole de résolution" parlé par le résolveur, ainsi que les "services de résolution" qu'il offre. Les protocoles de résolution mentionnés dans cette spécification sont Z3950, THTTP, RCDS, HDL, et RWHOIS. (On pense que cette liste augmentera à l'avenir). La spécification de NAPTR énumère aussi divers services de résolution, tels que N2L (si on donne un URN, il retourne un URL) ; N2R (si on donne un URN, il retourne la ressource désignée) etc.

Le présent document spécifie le protocole de résolution "THTTP" (Trivial HTTP). THTTP est une simple convention de codage des demandes et réponses de service de résolution comme les demandes et réponses HTTP 1.0 ou 1.1. Le principal objectif de THTTP est d'avoir un protocole de résolution d'URN qui puisse aisément être ajouté aux démons HTTP existants. D'autres protocoles de résolution émergeront avec le temps, de sorte que le présent document sert un objectif secondaire d'illustration des informations qui doivent être spécifiées pour un protocole de résolution d'URN. Un des protocoles de résolution dont on espère le développement est une extension de HTTP avec de nouvelles méthodes pour les services de résolution. Nous utilisons donc "THTTP" comme l'identifiant pour ce protocole afin de laisser "HTTP" libre pour les développements ultérieurs.

Le lecteur est supposé être familiarisé avec les spécifications HTTP/1.0 [RFC1945] et 1.1 [RFC2068]. Ceux qui mettent en œuvre la présente spécification devraient être familiarisés avec les scripts CGI, ou les interfaces spécifiques du serveur, pour les recherches dans les bases de données.

2. Approche générale

L'approche générale utilisée pour coder les demandes de service de résolution en THTTP est assez simple :

```
GET /uri-res/<service>?<uri> HTTP/1.0
```

Par exemple, si on a l'URN "urn:foo:12345-54321" et si on veut un URL, on va envoyer la demande :

```
GET /uri-res/N2L?urn:foo:12345-54321 HTTP/1.0
```

La demande pourrait aussi être codée comme demande HTTP 1.1. Cela ressemblerait à :

```
GET /uri-res/N2L?urn:foo:12345-54321 HTTP/1.1
Host: <c'est l'hôte auquel on envoie la demande>
```

Les réponses du serveur HTTP suivent la pratique HTTP standard. Les codes d'état, tels que 200 (OK) ou 404 (Pas trouvé) devront être retournés. Les règles normales pour déterminer la possibilité de mettre en antémémoire, la négociation des formats, etc., s'appliquent.

Le traitement de ces demandes du côté du serveur est facile à mettre en œuvre en utilisant CGI ou un autre mécanisme d'extension, spécifique du serveur. Les scripts CGI verront l'URI entrant dans la variable d'environnement `QUERY_STRING`. Tout caractère codé en % dans l'URN va rester dans l'état codé en % dans cette chaîne. Le script peut prendre l'URN, le chercher dans une base de données, et retourner les informations demandées.

Un avertissement devrait cependant être gardé en mémoire. Le document qui décrit la syntaxe des URN [RFC2141] discute de la notion d'équivalence lexicale et exige que les résolveurs retournent des résultats identiques pour les URN qui sont lexicalement équivalents. Les mises en œuvre de la présente spécification doivent être attentives à respecter cette règle. Par exemple, les deux demandes ci-dessous DOIVENT retourner des résultats identiques, car les URN sont lexicalement équivalents.

```
GET /uri-res/N2L?urn:cid:foo@huh.com HTTP/1.0
GET /uri-res/N2L?URN:CID:foo@huh.com HTTP/1.0
```

3. Détails spécifiques du service

On examine dans la présente section les divers services de résolution établis dans le document sur les services d'URN [RFC2483] et on établit comment coder chacun d'eux, comment les résultats devraient être retournés, et tous les codes d'état particuliers qui ont une probabilité d'apparaître.

Sauf mention contraire, les demandes THHTTP sont formées conformément à la simple convention donnée ci-dessus pour HTTP/1.0 ou HTTP/1.1. La réponse est supposée être une entité avec des en-têtes et un corps normaux; sauf mention contraire. (N2L est la seule demande qui n'a pas besoin de retourner un corps).

3.1 N2L (d'URN à URL)

La demande est codée comme ci-dessus. L'URL DOIT être retourné dans un en-tête "Location:" pour le confort de l'utilisateur qui veut dans la plupart des cas restituer la ressource. Si la recherche est réussie, une ligne d'état 30X DEVRAIT être retournée. Il devrait être envoyé aux clients HTTP/1.1 le code d'état 303. Il devrait être envoyé aux clients HTTP/1.0 le code d'état 302 (Déplacé temporairement) sauf si le résolveur a des raisons particulières d'utiliser les codes 301 (Déplacement permanent) ou 304 (Non modifié).

Noter que les contrôles d'accès peuvent être appliqués à cette demande de service de résolution, ou à toute autre. Donc les codes d'état 401 (Non autorisé) et 403 (Interdit) sont des réponses légalles. Le serveur peut souhaiter fournir un corps dans la réponse pour expliquer la raison du refus d'accès, et/ou fournir des informations de remplacement sur la ressource, comme le prix qu'il en coûtera pour obtenir l'URL de la ressource.

3.2 N2Ls (d'un URN à des URL)

La demande est codée comme ci-dessus. Le résultat est une liste de zéro, un ou plusieurs URL. Le type de support Internet (autrement dit, le type de contenu) du résultat peut être négocié en utilisant les mécanismes HTTP standard si désiré. Au minimum, le résolveur devrait prendre en charge le type de support `text/uri-list`. (Voir à l'Appendice A la définition de ce type de support). Ce type de support convient pour le traitement machine de la liste des URL. Les résolveurs peuvent aussi retourner les résultats comme type de support `text/html`, `text/plain`, ou tout autre qu'ils estiment convenable.

Sans considération du type de support particulier, le résultat DOIT être une liste des URL qui peuvent être utilisés pour obtenir une instance de la ressource identifiée par l'URN. Tous les URI doivent être codés conformément à la spécification d'URI [RFC1630].

Si le client a demandé que le résultat soit retourné comme `text/html` ou `application/html`, le résultat devrait être un

document HTML valide contenant le fragment :

```
<UL>
<LI><A HREF="...url 1...">...url 1...</A>
<LI><A HREF="...url 2...">...url 2...</A>
etc.
</UL>
```

où les chaînes ...url n... sont remplacées par le n^{ème} URL de la liste.

3.3 N2R (de l'URN à la ressource)

La demande est codée comme ci-dessus. La ressource est retournée en utilisant les mécanismes HTTP standard. La demande peut être modifiée en utilisant l'en-tête Accept: comme en HTTP normal pour spécifier que le résultat est donné dans un type de support Internet préféré.

3.4 N2Rs (de l'URN à des ressources)

Ce service de résolution retourne plusieurs instances d'une ressource, par exemple, les versions GIF et JPEG d'une image. Le jugement sur le fait que les ressources sont "les mêmes" appartient à l'autorité de désignation qui a produit l'URN.

La demande est codée comme ci-dessus. Le résultat devra être un message MIME multipart/alternatif avec les versions de remplacement de la ressource dans des parties de corps séparées. Si il n'y a qu'une seule version de la ressource identifiée par l'URN, elle PEUT être retournée sans l'enveloppe multipart/alternative. Le logiciel de résolveur DEVRAIT examiner l'en-tête Accept:, s'il en est un, et ne retourner que les versions de la ressource qui sont acceptables selon cet en-tête.

3.5 N2C (d'URN à URC)

Les caractéristiques de ressource uniforme (URC, *Uniform Resource Characteristics*) sont des descriptions d'autres ressources. Cette demande nous permet d'obtenir une description de la ressource identifiée par un URN, par opposition à la ressource elle-même. La description pourrait être une citation bibliographique, une signature numérique, un historique des révisions, etc. Le présent document ne spécifie pas le contenu des réponses à une demande d'URC. Ce contenu devrait varier d'un résolveur à l'autre.

Le format de toute réponse à une demande N2C DOIT être communiqué en utilisant l'en-tête Type de contenu, comme c'est la pratique HTTP standard. L'en-tête Accept: DEVRAIT être honoré.

3.6 N2Ns (d'un URN à des URN)

Bien que les URN soient supposés identifier une seule ressource, cela ne signifie pas qu'une ressource ne peut avoir qu'un seul URN. Par exemple, considérons une ressource qui a quelque chose comme "carte-de-la-météorologie-actuelle" pour un URN et "carte-de-la-météorologie-à-l'heure-x" pour un autre URN. La demande du service N2Ns nous fait obtenir les listes des URN qui sont estimées équivalents à l'heure de la demande. Comme le montre l'exemple de la carte météorologique, certaines des équivalences seront transitoires, de sorte que les mécanismes HTTP standard pour la mise en antémémoire communicante DOIVENT être honorés.

La demande est codée comme ci-dessus. Le résultat est une liste de tous les URN, connus du résolveur, qui identifient la même ressource que l'URN d'entrée. Le résultat devra être codé comme pour les demandes N2Ls ci-dessus (text/uri-list sauf spécification contraire par un en-tête Accept:).

3.7 L2Ns (d'un URL à des URN)

La demande est codée comme ci-dessus. La réponse est une liste de tous les URN connus pour être alloués à la ressource à l'URL en question. Le résultat devra être codé comme pour les demandes N2Ls et N2Ns.

3.8 L2Ls (d'un URL à des URL)

La demande est codée comme décrit ci-dessus. Le résultat est une liste de tous les URL que le résolveur connaît pour être associés à la ressource localisée par l'URL en question. Ceci est codé comme pour les demandes N2Ls, N2Ns, et L2Ns.

3.9 L2C (d'un URL à un URC)

La demande est codée comme ci-dessus, la réponse est la même que pour la demande N2C.

Appendice A Type de support Internet text/uri-list

[Cet appendice sera augmenté ou remplacé par l'enregistrement de l'IMT text/uri-list une fois que cet enregistrement aura été effectué].

Plusieurs des demandes de service de résolution, telles que les N2Ls, N2Ns, L2Ns, L2Ls, résultent en un retour d'une liste des URI au client. Le type de support Internet text/uri-list est défini pour fournir un format simple du traitement automatique de telles listes des URI.

Le format de la ressource text/uri-list est :

- 1) Toute ligne commençant par le caractère '#' est une ligne de commentaires et est ignorée durant le traitement. (Noter que '#' est un caractère qui peut apparaître dans les URI, de sorte qu'il ne note un commentaire que quand il est le premier caractère de la ligne).
- 2) Les lignes restantes qui ne sont pas des commentaires DOIVENT être des URI (URN ou URL) codés conformément à la spécification des URI [RFC1630]. Chaque URI devra apparaître sur une ligne et une seule.
- 3) Comme pour tous les formats text/*, les lignes sont terminées par une paire CR LF, bien que les clients devraient être libéraux en acceptant des lignes avec seulement un de ces caractères.

Dans les applications où un URI a été transposé en une liste des URI, comme dans une réponse à la demande N2Ls, la première ligne de la réponse text/uri-list DEVRAIT être un commentaire donnant l'URI original.

Un exemple d'un tel résultat pour la demande N2L est donné ci-dessous :

```
# urn:cid:foo@huh.org
http://www.huh.org/cid/foo.html
http://www.huh.org/cid/foo.pdf
ftp://ftp.foo.org/cid/foo.txt
```

Appendice B Script n2l.pl

Ceci est un simple script CGI pour le service de résolution N2L. Il suppose la présence d'une base de données DBM pour mémoriser les transpositions d'URN en URL. Ce script ne spécifie pas le comportement standard, il n'est donné qu'à titre d'indication pour les mises en œuvre. En fait, ce script ne traite pas les en-têtes Accept: entrants, ni ne génère les codes d'état. Un tel comportement devrait faire partie d'un script réel pour tous les services de résolution.

```
#!/bin/perl
# N2L - effectue la résolution d'urn en url

$N2l_File = "...nom de fichier pour la base de données DBM...";

$urn = $ENV{'QUERY_STRING'} ;

# Les vérifications de bonne santé sur l'URN. La longueur minimum d'un URN valide est de 7 caractères - "urn:",
# un identifiant d'espace de nom de un caractère, ":", et une chaîne de un caractère spécifique de l'espace de noms.
# Des vérifications de bonne santé plus élaborées devraient faire partie d'un script réel de résolveur script.

if(length($urn)<7)
{
    $error=1;
}

if(!$error)
{
    # Convertit les versions lexicalement équivalentes d'un URI en une version canonique pour les recherches de BDD.
    $urn =~ s/^urn:([\^:]*):(.*)$/sprintf("urn:%s:%s", lc $1, $2)/ie;

    dbmopen(%lu,$N2l_File,0444);
    if($lu{$urn})
```

```
{
    $url=$lu{$urn};
    print STDOUT "Location: $url\n\n";
} else {
    $error=2;
}
dbmclose(%lu);
}

if($error)
{
    print "Content-Type: text/html \n\n";
    print "<html>\n";
    print "<head><title>URN Resolution: N2L</title></head>\n";
    print "<BODY>\n";
    print "<h1>La résolution d'URN en URL a échoué pour l'URN:</h1>\n";
    print "<hr><h3>$urn</h3>\n";
    print "</body>\n";
    print "</html>\n";
}

exit;
```

Références

- [RFC1630] T. Berners-Lee, "[Identifiants de ressource universels](#) dans la Toile mondiale ; syntaxe unificatrice pour l'expression des noms et adresses des objets du réseau utilisés sur la Toile mondiale", juin 1994. (*Information*)
- [RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk, "[Protocole de transfert Hypertext](#) -- HTTP/1.0", mai 1996. (*Info.*)
- [RFC2068] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Protocole de transfert Hypertext -- [HTTP/1.1](#)", janvier 1997. (*Obsolète, voir RFC2616*) (*P.S.*)
- [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*) (*Ex.*)
- [RFC2483] M. Mealling et R. Daniel, "[Services de résolution d'URI](#) nécessaires pour la résolution d'URN", janvier 1999.

Considérations pour la sécurité

Les communications avec un résolveur peuvent être de nature sensible. Certains résolveurs vont détenir des informations qui ne devraient être livrées qu'à des utilisateurs autorisés. Les résultats provenant des résolveurs peuvent être la cible de mystifications, en particulier lorsque les transactions de commerce électronique sont courantes et qu'on peut gagner de l'argent en redirigeant les usagers sur des sites pirates plutôt que sur ceux qui payent des redevances aux détenteurs de droits. Les demandes de résolution peuvent intéresser les analystes de trafic. Les demandes peuvent aussi être l'objet de mystifications.

Les demandes et réponses de ce projet sont susceptibles d'être codées, signées et authentifiées de la même manière que tout autre trafic HTTP.

Informations pour contacter l'auteur :

Advanced Computing Lab, MS B287
Los Alamos National Laboratory
Los Alamos, NM, USA, 87545
téléphone : +1 505 665 0597
fax : +1 505 665 4939
mél : rdaniel@lanl.gov