Distributed Universal Resource Name Resolution based on Distributed DNS

draft-licanhuang-dnsop-urnresolution-04.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 13, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This file is a proposal for Universal Resource Name resolution based on
semantic P2P network-VIRGO.
1. Introduction

This file is a proposal for P2P based Universal Resource Name resolution based on Internet draft --Distributed DNS Implementation in IpV6[lican].

In Internet, every resource needs an uniform name. There are RFCs URI[RFC2396], URL[RFC1738] and URN[RFC2611]. However, when there are more and more computers connected in Internet and more and more users use Internet, resources in Internet will be more and more. In this large, decentralized, distributed resource sharing environments today or in the future, the efficient discovery of static and/or dynamic resources (the dynamic resources are meant both in dynamic change of attributes or dynamic exist in the network) is a big challenge.

The resource resolution based on Client/Server is not suitable for huge amount of resources. Here we argue that the resources can be classified as hierarchical Domains. It is nature way to use DNS to resolute resources. Although DNS implementation today may be scalable for translating Domain Names into IP addresses, it may encounter problem if we use it to resolute huge amount resources today and even more amount resources in the future. The framework proposed here is suitable for the huge amount of resources’ resolution, which is based on Distributed DNS Implementation in IpV6[lican] and on the paper-"Resource Discovery Based on VIRGO P2P Distributed DNS Framework"[LicanICCS].

2. Convention for Universal Resource Name

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Convention for Universal Resource Name</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Universal Resource Name Resolution framework</td>
<td>3</td>
</tr>
<tr>
<td>3.1</td>
<td>Universal Resource Name Resolution</td>
<td>4</td>
</tr>
<tr>
<td>3.1.1</td>
<td>SQL-like Query language</td>
<td>4</td>
</tr>
<tr>
<td>3.1.2</td>
<td>SQL-like Maintanence Language</td>
<td>6</td>
</tr>
<tr>
<td>3.2</td>
<td>Hierarchical Management of global-hier-part</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>References</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Normative References</td>
<td>7</td>
</tr>
<tr>
<td>4.2</td>
<td>Informative References</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Author’s Address</td>
<td>8</td>
</tr>
</tbody>
</table>
The convention for resource name is formatted as global unique hierarchical domains like domain names in DNS, and a local name to specify the local attributes of resource name. The format of convention of resource names is as follows syntax:

    functionscheme "":"global-hier-part"/"local-name"

The functionscheme is type or function or schema for the resource; the global-hier-part is the global hierarchical name similar as domain name and the local-name is the local name for the resource.

for example,

    blog:Britney.popular.music/storyofBritney

This is a resource name for a blog article about the story of Britney Spears

    song:Britney.popular.music/Lonely

This is a resource name for the song Lonely by Britney Spears.

Britney Spears is classified as Britney.popular.music.

3. Universal Resource Name Resolution framework

We treat global-hier-part in the invention of resource name as domain name. We also use the similar concept of DNS zone for DNS server to manage the domain names. For example, if a DNS server’s zone is Science.Biology.Bioinformatics, then this DNS server manages all resource information under the catalogue of Science.Biology.Bioinformatics. In other case, if a DNS server’s zone is Science.Biology, then this DNS server manages all resource information under the catalogue of Science.Biology, which may includes sub domains of BioInformatics, botany, zoology, etc.

Resource names are published in their authoritative DNS server which keeps local domain RRs[RFC1035]. The extension RRs in DNS server contains the parts of the functionscheme and local-part of conventions of resource name. The local-part of resource can be a URL for the other repository which contains the resource metadata and/or other information. For example, the location information for blog:Britney.popular.music/storyofBritney is stored on the server of Britney.popular.music.

Every DNS server is the same but some coexist in more than one layer. Every DNS Server maintains a route table and RRs related to its
authoritative domain zone. Route Tables in servers are the same as [lican].

The DNS servers join virtual groups in VIRGO network as their authoritative domain names. When a DNS server joins, its route table will add some other nodes' entities, and some other nodes will update their route table by adding the joining node’s entity. The detail of the algorithm can be found in [lican].

3.1 Universal Resource Name Resolution

For the Universal Resource Name Resolution, we first locate the domain name server as [lican]. For example, to resolve resource "blog:Britney.popular.music/storyofBritney", we first locate the server Britney.popular.music, then in the server Britney.popular.music, we check its extension RRs for the resource blog:Britney.popular.music/storyofBritney for its location, etc.

The query process is as the following:

User program sends QUERY MESSAGE, which contains SQL-like query statement described in section 3.1.1, to Local Name Server. If Local Name Server is the authoritative Domain Name Server, then the Local Name Server will check its extension RRs to resolve the request resource. Otherwise, The Local Name Server will route to the Foreign Name Server which is closer to the authoritative Domain Name Server by calculating theoretical hops. Then the Foreign Name Server routes to the even closer Foreign Domain Name Server. Repeat this process, until the authoritative Domain Name Server has been found. Finally, the authoritative Domain Name Server resolves request resource by check its extension RR record, and responses to the Local Name Server. The latter will forward the response to the User Program. The details of the algorithm can be found [lican].

3.1.1 SQL-like Query Language

query::=SELECT{SCHEMA|SUBDOMAIN|ENTITY|ATTRIBUTE|*|expr[[AS]c_alias]
   ,expr[[AS]c_alias]...} FROM domainref [WHERE search_condition];

Here, SCHEMA is for querying functionschema for a Domain; SUBDOMAIN is for querying subdomain for a given super Domain; ENTITY is for querying all entities for a given search condition; ATTRIBUTE is for querying all attributes for a given search condition; expr is for the property queried, which may be a set of properties; [AS]
c_alias is the alias name for expr; domainref is for Domain Name, in which we can use * to indicate all sub domains; search_condition is similar to SQL statement in Database.

In the following, there are several examples of query operations.

Operation 1

The form "?:global-hier-part" is to query all functionschema of the global-hier-part domain. For example, "?:Britney.popular.music" queries all functionschema of Britney.popular.music. We can use the following Statement:

```
SELECT SCHEMA FROM Britney.popular.music;
```

Operation 2

The form "functionschema:global-hier-part/?" is to query all resources with the functionschema in the global-hier-parts domain. For example, "song:Britney.popular.music/?" queries all songs by Britney Spears. We can use the following Statement:

```
SELECT * FROM Britney.popular.music WHERE SCHEMA = 'song';
```

Operation 3

Suppose that global-hier-part can be expressed as leafDomain.superDomain. The form "functionschema:?.superDomain/" is to query all leafDomains. This is possible because of the hierarchical structure and the protocols. For example, "song:?.popular.music" queries all popular singers. We can use the following Statement:

```
SELECT SUBDOMAIN FROM *.popular.music WHERE SCHEMA = 'song';
```

Operation 4

The form "functionschema :global-hier-part/local-name" is to query specific resource. For example, "song:Britney.popular.music/Lonely" queries the information of the song-Lonely. We can use the following Statement:

```
SELECT Lonely FROM Britney.popular.music WHERE SCHEMA = 'song';
```

Operation 5

The form "functionschema :global-hier-part_EXPRESSION" is to query
the resources which satisfy the conditions of the expression indicated. Expression can be scope, maximum, minimum, where condition like SQL statements. For example, "song:Britney.popular.music/where year between 2006 and 2007" queries all songs by Britney Spears which are produced between 2006 and 2007. We can use the following Statement:

```
SELECT * FROM Britney.popular.music WHERE SCHEMA = 'song' AND Year >= 2006 AND Year <= 2007;
```

Another example, "hardware:cluster.computer.IT/max(cpu)" queries a cluster with maximum cpu numbers. We can use the following Statement:

```
SELECT max(cpu) FROM cluster.computer.IT WHERE SCHEMA = 'hardware';
```

3.1.2 SQL-like Maintenance Language

The resources are maintained by the SQL-like maintenance language. Insert statement registers new resources into the domain. Delete statement drops the resource registration from the domain. Update statement changes the information registered in the domain. The formats of these statements are as the following:

```
DELETE FROM domainref [WHERE search_condition];

INSERT INTO domainref[(entity.attribute{, entity.attribute...})] (VALUES (expr | NULL {,expr | NULL...}) ;

UPDATE domainref SET entity.attribute = expr | NULL{, entity.attribute = expr | NULL ...};
```

For example, the following statement deletes all songs by Britney Spears which are produced between 2006 and 2007.

```
DELETE FROM Britney.popular.music WHERE SCHEMA = 'song' AND year > 2006 AND year < 2007;
```

3.2 Hierarchical Management of global-hier-part
To be a unique resource name for global scope, The Internet Corporation for Assigned Names and Numbers (ICANN) may govern the root of global-hier-part in Universal Resource Name organized by the users automatically. For example, in www.Madonna.popular.music, music is managed by music virtual organization, which reports to ICANN to approve, popular.music is controlled by popular music virtual organization, which reports to music virtual organization to approve. The request of www.Madonna.popular.music is sent to virtual organization of popular music to be approved. In www.Beethoven.classic.music, classic.music is controlled by classic music virtual organization, which reports to music virtual organization to approve. The request of www.Beethoven.classic.music is managed by virtual organization of classic music.

4. References

4.1. Normative References


4.2. Informative References


[Lican] Lican Huang, "draft-licanhuang-dnsop-distributeddns-06.txt" (work in progress), July 2009.

Authors’ Addresses

Lican Huang
Current Address:
Institute of Network & Distributed Computing,
Zhejiang Sci_Tech University,
Hangzhou, P.R.China
EMail: licanhuang@zist.edu.cn; huang_lican@yahoo.co.uk