CDNI
Internet-Draft
Intended status: Informational
Expires: December 30, 2012

G. Chen
China Telecom
M. Li
H. Xia
ZTE Corporation
J. Liang
China Telecom
June 28, 2012

Request Routing and Content Acquisition
draft-chen-cdni-rr-content-acquisition-00

Abstract

This document illustrates the details of an alternative method that can be used to provide request routing and content acquisition services.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on December 30, 2012.

Copyright Notice

Copyright (c) 2012 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
  1.1. Terminology 3
2. Prerequisite 3
  2.1. CDN Interconnection Framework 3
  2.2. UniContentID 5
  2.3. Use Case Description 7
    2.3.1. Request Routing and Content Delivery 7
    2.3.2. Content Acquisition 9
3. Security Considerations 10
4. IANA Considerations 10
5. Acknowledgments 10
6. References 10
  6.1. Normative References 10
  6.2. Informative References 10
Authors' Addresses 11

1. Introduction

The scope of CDNi work is described in [I-D.ietf-cdni-problem-statement]:

- As illustrated in Figure 1, the acquisition of content between interconnected CDNs is out of scope for CDNI, which deserves some additional explanation. The consequence of such a decision is that the CDNI problem space described in this document is focused on only defining the control plane for CDNI; and the CDNI data plane (i.e., the acquisition and distribution of the actual content objects) is out of scope.

It can be implied that the delivery process of the actual content object requested by the end user from uCDN to dCDN is outside the scope of CDNi work. However, in the cache miss case, i.e., the dCDN receives end user's request redirected by the uCDN and does not find the corresponding content in the cache, the dCDN needs to determine toward which uCDN it should initiate the content acquisition procedure. And this should be covered within CDNi's scope.

This document illustrates the mechanism of request routing and content acquisition between uCDN and dCDN on the basis of Content Identification.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document reuses the terminology defined in:

[I-D.draft-ietf-cdni-problem-statement-06],

[I-D.draft-ietf-cdni-requirements-03],

[I-D.draft-ietf-cdni-framework-00], and

[I-D.draft-ietf-cdni-use-cases-08].

2. Prerequisite

2.1. CDN Interconnection Framework

In the CDNi framework, the process of end user's content request from the uCDN first, and then the content delivery by the dCDN can be

further divided into two sub-procedures: Request Routing and Content Acquisition. The method used in [I-D.ietf-cdni-framework] is that the CDN operators pre-agree for using 'distinguished' CDN-domains and embed them in URLs that end users request. The example used is a CDN-domain peer-a.op-b.net that will be used as the target of redirections from uCDN to dCDN and a CDN-domain op-b-acq.op-a.net that will be used for inter-CDN acquisition of CSP's content from uCDN by dCDN.

When the CDNi framework has even complex topology, i.e., in the case of cascaded CDNs, in order to record the redirection route of the request message, all the redirection routing information is required to be included in the URL, which would result in a very long URL. Limited by the length and format of URL, such approach would cause serious delay and waste of resources, and it would be difficult to implement this. In addition, in the process of request message redirection, there is possibility that the URL is modified by a CDN which may lead to inaccuracy of the redirection and content acquisition information.

This document introduces a content identification parameter called 'UniContentID' to uniquely identify acontent item, the details of which are illustrated in Section 2.2. This document also makes use of static configuration mechanism, i.e., every CDN provides a fixed URL or IP address for other CDN to download content from. This fixed URL or IP address is valid to all CDNs and all request messages sent to this address is considered as content downloading request. By doing this the change of request URL during multiple redirection processes can be avoided, and it would also accelerate the process of locatingcorresponding uCDN and the content item requested by the end user. This method is easy to implement and can be used in cascaded CDNs scenario.

Editor's Note: The mechanism for using dynamic mechanism to acquire the URL or IP address of uCDN for content downloading is FFS.

```
                    +------+
                    |  CP  |
                    +------+        :
                       |            :
                       |            :
                    _,.---.,,       :         _,.---.,,
                  .`          `.     :       .`          `.
                 '             \    :      '             \
                |     uCDN      |---:---- |     dCDN     |
                 ,             / ---:----  ,             /
                  ',         ,-     :       ',         ,-
                    ``''--'``       :         ``''--'``
                                    :             |
                                    :             |
                                    :          +------+
                                    :          | EU B |
                                    :          +------+
                       Provider A   : Provider B
                                    :
                                    :
                                    :
```

Figure 1 Interconnection of UCDN and dCDN

2.2. UniContentID

Given that the CDN needs to support the requirements for ingesting content to multi-screen for the same CMS (as described in ITU-T Y.1910 IPTV Functional Architecture), we need to define a two-tuple, i.e., the parameter of UniContentID to identify uniquely different CMSs.

UniConentID is defined by a two-tuple as (ProviderID, ContentID), which can uniquely identify only one content item in the CDN. Here the parameter ContentID denotes only one content item in a specific domain of a CMS. The parameter ProviderID is defined as the provider for specific domain and is only for a specific CMS in a specific domain. We define the configuration table in the CDN which needs to be used in the request routing as follows:

* CMSID: denotes the DNS name or IP address of the CMS.

* Domain: denotes the different mark for IPTV services, PC service and mobile service etc. For example, we define 0 for IPTV service, 1 for PC service, 2 for mobile service and 3 for reserved other service.

* ProviderID: denotes the mark of specific domain content provider, which is singular in this configuration table. For the same

parameter of CMSID, different parameter of Domain indicates different parameters of ProviderID.

* ProviderType: denotes the provider type as B2B service or B2C service. For example, we define 1 for B2B and 2 for B2C.

* PlaybackURLprefix: denotes the prefix of the specific content service URL of the portal. There is a one-to-one relationship between PlaybackURLpredix and ProviderID. For example, the URL is http://video.netitv.com/01234567890123456789012345678900?..., then the PlaybackURLpredix is video.netitv.com.

* ContentURLParseRule: denotes resolution rule for the parameter of URL and ContentID, which is a regular expression.

```
+-------------------------------------------------------------+
|CMS| Domain| ProviderID| Provider Type| Playback | ContentURL|
|                                        URLPrefix  ParseRule |
+-------------------------------------------------------------+
```

Figure 2 The Configuration Table

Example 1: the parameter of ProviderID matched through the parameter of CMSID and Domain

In the configuration table for IPTV service of specific CMS, we set the parameter of ProviderID is iptv.netitv.com and the parameter of CMSID is '10.17.45.233'. Then one content is ingested to the CDN and the parameter of ContentID is '01234567890123456789012345678900'. Domain is '0'. The two tuple of UniContentID is ('iptv.netitv.com','01234567890123456789012345678900').

In the website portal, the URL for the content serving is 'RTSP RR IP:PORT/ ContentID?CMSID=XXX &Domain=XXX...'. When the streaming server received the URL, it analyses the parameter of ProviderID is 'iptv.netitv.com' through looking up the configuration table. Then the two tuple UniContentID is('iptv.netitv.com', '01234567890123456789012345678900'). We can request the content in the local cache when cache hit or relay the content from the uCDN when the cache is not hit.

Example 2: the parameter of ProviderID matched for mobile service

In the configuration table for mobile service of specific CMS, we set the parameter of ProviderID is mobile.netitv.com and the parameter of PlaybackURLPrefix is 'mobile.netitv.com'. Then one content is

ingested to the CDN and the parameter of ContentID is '01234567890123456789012345678900'. Domain is '2'. The two tuple of UniContentID is ('mobile.netitv.com', '01234567890123456789012345678900').

In the website portal, the URL for the content serving is 'RTSP://mobile.netitv.com/01234567890123456789012345678900?...'. When the streaming server received the URL, it analyses the parameter of PlaybackURLprefix is 'mobile.netitv.com' and finds the parameter of ProviderID is 'mobile.netitv.com' through looking up the configuration table. Then the two tuple UniContentID is ('mobile.netitv.com', ' 01234567890123456789012345678900'). We can request the content in the local cache when cache hit or relay the content from the uCDN when the cache is not hit.

Example 3: ProviderID matched for PC service

In the configuration table for mobile service of specific CMS, we set the ProviderID is pc.netitv.com and PlaybackURLPrefix is 'video.netitv.com'. Then one content is ingested to the CDN and the ContentID is '01234567890123456789012345678900'. Domain is '1'. The two tuple of UniContentID is ('pc.netitv.com', '01234567890123456789012345678900').

In the website portal, the URL for the content serving is 'http://video.netitv.com/01234567890123456789012345678900?...'. When the streaming server received the URL, it analyses the parameter of PlaybackURLprefix is 'video.netitv.com' and finds the parameter of ProviderID is 'pc.netitv.com' through looking up the configuration table. Then the two tuple UniContentID is ('pc.netitv.com', ' 01234567890123456789012345678900'). We can request the content in the local cache when cache hit or relay the content from the uCDN when the cache is not hit.

2.3. Use Case Description

2.3.1. Request Routing and Content Delivery

```
                                        +------------------+
                                        |       dCDN       |
             +-----+        +------+    |+------+ +------+ |
             | EU  |        | uCDN |    || dCDN | | dCDN | |
             +-----+        +------+    ||  RR  | |  DN  | |
                |               |       |+------+ +------+||
                |               |       +------------------+
           RTSP or HTTP://uCDN RR IP:Port/ContentID?     |
                |-------------->|               |        |
                |CMSID=XXX&domain=XXX...        |        |
                |               |               |        |
                |               |               |        |
                |302 dCDN RR IP:Port/           |        |
                |<--------------|               |        |
                |ContentID?CMSID=XXX&domain=XXX...       |
                |               |               |        |
                |               |               |        |
                |RTSP or HTTP://dCDN RR IP:Port/ContentID?
                |------------------------------>|        |
                | CMSID=XXX&domain=XXX...       |        |
                |               |               |        |
                |  IPaddr of dCDN's Delivery Node        |
                |<------------------------------|        |
                |               |               |        |
                |               |               |        |
                | RTSP or HTTP://dCDN DN IP:Port/        |
                |---------------------------------------->|
                | ContentID?CMSID=XXX&domain=XXX...      |
                |               |               |        |
                |               |               |        |
```

Figure 3 Message Exchange for Request Routing

1. A Request Router of uCDN processes the RTSP or HTTP request and recognizes that the end-user is best served by another CDN. It returns the IP address of dCDN.

2. uCDN returns a 302 redirection message for a new URL including the IP address of dCDN.

3. The end-user then sends the request to the Request Router of dCDN (i.e. dCDN RR). dCDN RR returns the IP address of corresponding delivery node.

4. dCDN RR returns a new URL including the IP address of dCDN DN. Note that, since the name of the delivery node was already obtained from dCDN (i.e. dCDN DN), there should not be any further redirection here.

5. The end-user requests the content from dCDN's delivery node,potentially resulting in a cache miss. In the case of cache miss, the content needs to be acquired from uCDN (not the CSP), which is described in the following section.

2.3.2. Content Acquisition

```
+-----+       +------+      +-----+        +------+
| EU  |       | uCDN |      | mCDN|        | dCDN |
+-----+       +------+      +-----+        +------+
   |             |             |              |
   |             |          RTSP or HTTP://mCDN IP:Port/
   |             |             |<-------------|
   |             |          ContentID?ProviderID=XXX...
   |             |             |              |
   |         RTSP HTTP://uCDN IP:Port/        |
   |             |<------------|              |
   |          ContentID?ProviderID=XXX...     |
   |             |             |              |
   |           Content Acquisition Relay      |
   |             |------------>|              |
   |             |             |Content Acquisition Relay
   |             |             |------------> |
   |             |             |              |
   |             |             |              |
   |             |             |              |
   |             |    DATA     |              |
   |<-----------------------------------------|
   |             |             |              |
```

Figure 4 Message Exchange for Content Acquisition

1. The request router of dCDN processes the request from the end-user and forwards the request to the request router of mCDN. Note that the dCDN needs to obtain the ProviderID information by looking up the configuration table. The ProviderID and ContentID information can identify a unique content.

2. The request router of mCDN finds a cache miss case in the delivery nodes of mCDN. It forwards the request to the request router of uCDN for the content.

3. The request router of uCDN finds that the content is cached in the delivery node of uCDN, then the delivery node of uCDN delivers the content to the delivery node of mCDN.

4. The delivery node of mCDN delivers the content to the delivery node of dCDN.

5. The delivery node of dCDN delivers the content to the end-user.

Note: The content acquisition interface in step 3~5 needs to be standardized.

3. Security Considerations

To be added later

4. IANA Considerations

This memo has no IANA Considerations.

5. Acknowledgments

To be added later

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[I-D.ietf-cdni-framework]
Peterson, L. and B. Davie, "Framework for CDN Interconnection", April 2012.

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", May 2012.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", December 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Stephan, E., Burbridge, T., Eardley, P., Ma,

K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", June 2012.

[I-D.narten-iana-considerations-rfc2434bis]
Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

Authors' Addresses

Ge Chen
China Telecom
109 West Zhongshan Ave
Guangzhou, Tianhe District
China

Phone:
Email: cheng@gsta.com

Mian Li
ZTE Corporation
Nanjing, 210012
China

Phone:
Email: li.mian@zte.com.cn

Hongfei Xia
ZTE Corporation
Nanjing, 210012
China

Phone:
Email: xia.hongfei@zte.com.cn

Jie Liang
China Telecom
109 West Zhongshan Ave
Guangzhou, Tianhe District
China

Phone:
Email: liangj@gsta.com