Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: February 20, 2012

G. Bertrand, Ed.
France Telecom - Orange
F. Le Faucheur
Cisco Systems
L. Peterson
Verivue, Inc.
August 19, 2011

Content Distribution Network Interconnection (CDNI) Experiments
draft-bertrand-cdni-experiments-01

Abstract

This document reports studies and related experiments on CDN interconnection performed by France Telecom-Orange Labs. The document summarizes implications of CDN interconnection to CDN service providers and lessons learned through CDNI experiments.

The main purpose of the experiments was to test the interconnection of CDN solutions from two vendors (namely, Cisco and Verivue) and to identify the gaps and needs for standardization work for CDN interconnection.

Table of Contents

1. Introduction

This document reports studies and related experiments on CDN interconnection performed by France Telecom-Orange Labs. The document summarizes implications of CDN interconnection to CDN service providers and lessons learned through CDNI experiments.

The main purpose of the experiments was to test the interconnection of CDN solutions from two vendors (namely, Cisco and Verivue) and to identify the gaps and needs for standardization work for CDN interconnection.

This study is not intended to explore the entire scope of CDNI, and in fact, it purposely takes a minimalist approach. That is, we focus on what's minimally required to interconnect two cooperating CDNs in a "best effort" way. This provides a constructive foundation for adding requirements and mechanisms only after they prove essential in practice.

1.1. Terminology

Except for the terms defined below, we adopt the terminology described in [RFC3466], [RFC3568], [RFC3570], the problem statement draft [I-D.jenkins-cdni-problem-statement] and the use cases draft [I-D.bertrand-cdni-use-cases].

Content Distribution Network (CDN) / Content Delivery Network (CDN):

A type of network in which the components are arranged for more effective delivery of content to User Agents.

Content Delivery Service

Set of services offered to content providers (CPs) for delivering their content through a single Content Delivery Network or interconnections of Content Delivery Networks.

CDN Service Provider (CDSP):

An administrative entity who operates a CDN over a Network Service Provider (NSP) or over the Internet.

Authoritative CDN (aCDN):

The CDSP contracted by the CP for delivery of content by this CDN or by its downstream CDNs.

Downstream CDN (dCDN):

A CDSP which is contracted by an upstream CDN to achieve the delivery of content to users.

CDN Interconnection (CDNI):

Relationship between two CDNs that enables a CDN to provide content delivery services on behalf of another CDN. It relies on a set of interfaces over which two CDNs communicate in order to achieve the delivery of content to end-users by one CDN (the downstream CDN) on behalf of another CDN (the upstream CDN).

Recursive Request Routing:

Recursive: Where a process is repeated, but embedded within the original process. In the case of Request Routing, this means that the initial request received by the Authoritative CDN is processed downstream from one CDN to another and that the responses are send back upstream to the Authoritative CDN which then replies to the initial request.

Iterative Request Routing:

Iterative: Where a process is repeated multiple times to make progress towards a goal. In the case of Request Routing, this means that the initial request is received by the Authoritative CDN, which replies it with a redirection directive to a dowstream CDN. When the end-user sends its request to the downstream CDN, the same process is repeated, until the request arrives to the delivering CDN.

User-Agent (UA):

A client application acting as the end point of a communication session.

2. CDN Interconnection Experiments

2.1. Experiment Configuration

The interconnection of two CDN solutions from different vendors has been tested. These tests have been run with CDN solutions from Cisco (hereafter referred to as Vendor A) and from Verivue/CoBlitz (hereafter referred to as Vendor B).

As depicted in Figure 1, we have interconnected two experimental CDNs (CDN A and CDN B) operated by different subsidiaries of a large CDSP. The CDNs lab equipment were located in two different countries, henceforth referred to as Country A and Country B and they relied on

CDN solutions from two different equipment vendors, namely, Vendor A and Vendor B. The CDNI experiment supported the services of two emulated CPs (CP A and CP B).

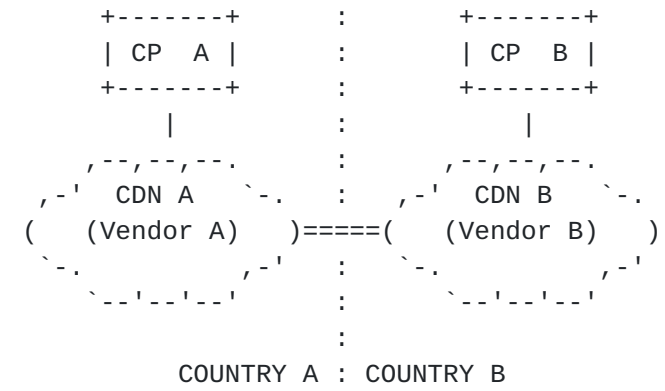


Figure 1

More precisely, we have run the experiments represented in Figure 2 and Figure 4. We base our description on Figure 2. In this experiment, CP A has an agreement with CDSP A for content delivery to end-users located in Country A and Country B. However, CDSP A operates a CDN (CDN A), whose footprint does not include country B. Therefore, CDSP A has an agreement with CDSP B, so that CDN A can delegate to CDN B the delivery of some content. More specifically, CDN A is allowed to delegate to CDN B the handling of requests sent by end-users located in Country B for CP A's content.

When CDN A receives a content request related to CP A and from an end-user in Country B, it redirects the end-user to CDN B. If CDN B does not have a local copy of the requested content yet (cache miss), CDN B ingests the content from CDN A (or from the CP's origin servers, depending on the test scenario); if CDN A also does not have a local copy of the requested content, it requests this asset from the CP's origin servers before sending the asset to CDN B.

There are several differences between the tests in Figure 2 and Figure 4, in addition to the different role played by the two CDN solutions. We list the main ones below.

- We have tested different content acquisition methods (see Section 2.6).

- Specific URL schemes were involved in providing content acquisition source information to the downstream CDN. As we have tested different content acquisition methods, depending on which solution played the role of dCDN, the two solutions have used different URL schemes to address content. Therefore, the tests required the configuration of different content delivery metadata on the uCDN (see Section 2.4).

- The two solutions use different methods to identify the end-user's geographic locations (see Section 2.4).

2.2. Control

The tested CDN solutions support control APIs but those are proprietary, so that the tested CDN solutions do not support a common inter-operable CDNI control interface. Therefore, we have not tested CDNI control operations and we had to perform manually most operations related to the configuration of the CDNI.

2.3. Logging

Proprietary mechanisms to export transaction logs were available in the tested CDN solutions, but have not been covered by our tests.

2.4. Request Routing and Content Delivery

As defined in [I-D.bertrand-cdni-use-cases], two main types of request-routing call flows can be used for CDNI:

1. iterative request-routing,

2. recursive request-routing.

Moreover, two main methods can be used for redirecting an end-user from the authoritative CDN to the downstream CDN:

1. DNS-based redirection,

2. Service-level redirection, for example HTTP-based or RTSP-based.

We have focused our tests on iterative request-routing. Our tests involved HTTP-based request redirection by the authoritative CDN, as they focused on the delivery of large objects, such as movies.

The tested CDN solutions did not feature a CDNI request-routing interface allowing exchange of "CDN routing" information among CDNs. Therefore, we have manually configured appropriate policies on the authoritative CDN to permit iterative request-routing (e.g., CDN A

redirects the end-user to CDN B request-routing system, cf. Figure 3).

2.4.1. HTTP Redirection by CDN A and Delivery by CDN B

This section describes the tested request routing and content delivery features in the scenario depicted in Figure 2, with HTTP redirection by CDN A and delivery by CDN B.

We have tested the selection of the downstream CDN based on the content type in the client request and/or the geographic location of the client.

- File-type based selection relied on static XML files where content file extensions can be associated to a specific delivery CDN.
- The geographic location of the end-user was determined by an external geolocation server.

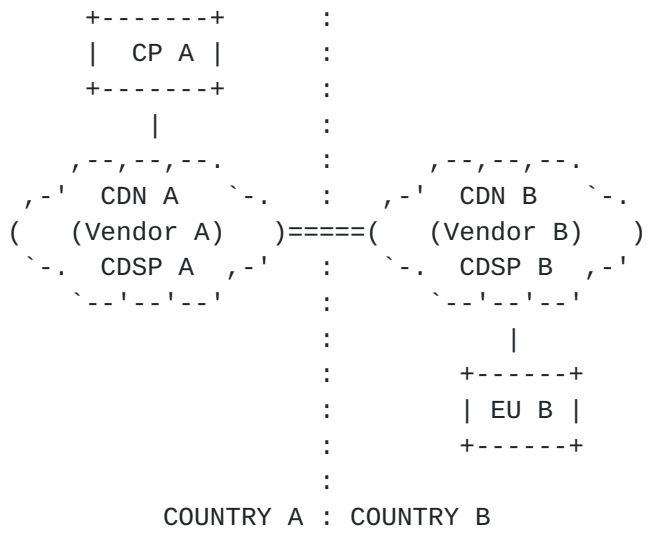
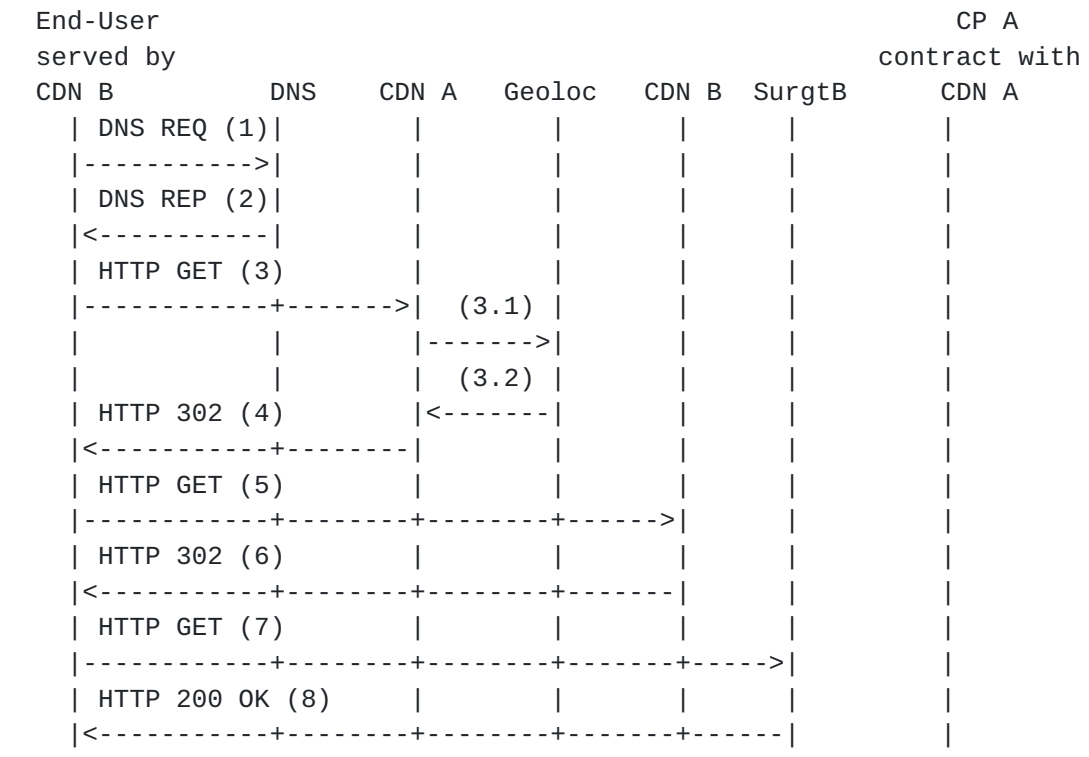


Figure 2

Figure 3 details the messages exchanged by the components involved in the experiment.



HTTP redirection by CDN A and delivery by CDN B

Figure 3

Message details

(1) The user-agent sends a DNS request to resolve the FQDN of the content URI.

(2) The DNS answers with the IP address of a request-router in CDN A.

(3) The user-agent sends to CDN A request-router an HTTP GET request for the content URI.

(3.1) CDN A request-router analyzes the request and queries a geolocation database to identify the geographic location of the end-user.

(3.2) The geolocation database answers with geolocation information related to the end-user's IP address. The end-user is in country B; thus, CDN A determines that the end-user's request must be served by CDN B.

(4) CDN A request-router replies to the user-agent with an HTTP 302 redirection message, which provides the URI of the content on CDN B.

(5) If necessary, the user-agent resolves the FQDN on the redirection URI (steps not represented in the figure), and thus, determines the IP address of a request-router in CDN B. Then, it sends an HTTP GET request to this request-router.

(6) CDN B request-router analyzes the request and replies to the user-agent with an HTTP 302 redirection message that provides the URI of the content on a surrogate in CDN B.

(7) If necessary, the user-agent resolves the FQDN of the redirection URI (steps not represented in the figure), and thus, determines the IP address of a surrogate in CDN B. Then, it sends an HTTP GET request to the surrogate.

(8) The surrogate analyzes the request and delivers the requested content to the end-user, through an HTTP 200 OK message.

2.4.2. HTTP Redirection by CDN B and Delivery by CDN A

This section describes the tested request routing and content delivery features in the scenario depicted in Figure 4, with HTTP redirection by CDN B and delivery by CDN A.

We have tested the selection of the downstream CDN based on end-user's geolocation. For these tests, the geolocation database had been populated manually with the mapping of IP prefixes to countries. Alternative solutions, such as geolocation based on BGP communities or on the extraction of per country IP prefixes thanks to commercial geoIP databases, exist but they have not been tested in this experiment.

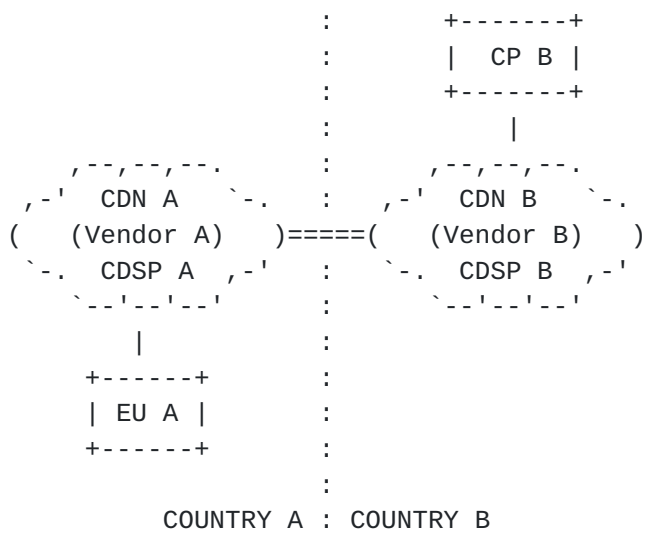
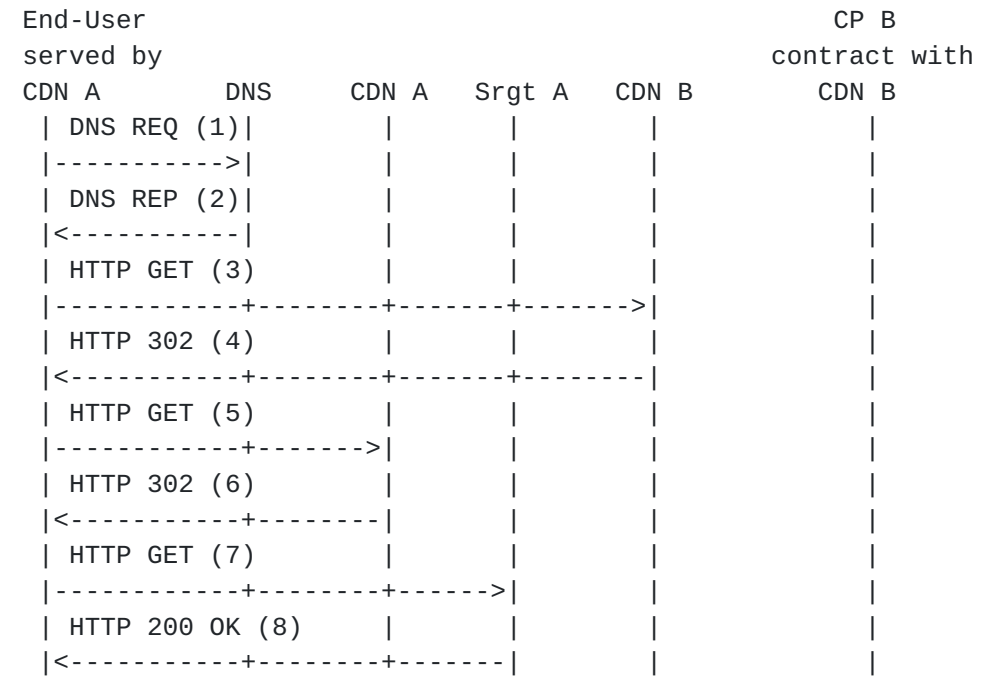


Figure 4

Figure 5 details the messages exchanged by the components involved in the experiment.



HTTP redirection by CDN B and delivery by CDN A

Figure 5

Message details

(1) The user-agent sends a DNS request to resolve the FQDN of the content URI.

(2) The DNS answers with the IP address of a request-router in CDN B.

(3) The user-agent sends to CDN B request-router an HTTP GET request for the content URI.

(4) CDN B request-router analyzes the request. The end-user is in country A; thus, CDN B determines that the end-user's request must be served by CDN A. Consequently, CDN B replies to the user-agent with an HTTP 302 redirection message that provides the URI of the content on CDN A.

(5) If necessary, the user-agent resolves the FQDN on the redirection URI (steps not represented in the figure), and thus, determines the IP address of a request-router in CDN A. Then, it sends an HTTP GET request to this request-router.

(6) CDN A request-router analyzes the request and replies to the user-agent with an HTTP 302 redirection message, which provides the URI of the content on a surrogate in CDN A.

(7) If necessary, the user-agent resolves the FQDN of the redirection URI (steps not represented in the figure), and thus, determines the IP address of a surrogate in CDN A. Then, it sends an HTTP GET request to the surrogate.

(8) The surrogate analyzes the request and delivers the requested content to the end-user, through an HTTP 200 OK message.

2.4.3. Test Result

HTTP redirection by the authoritative CDN was successful in the tests: end-users were redirected to the CDN that served their country. This guaranteed that:

- content from CP A be delivered by CDN B to end-users in country B, even if CP A had no direct relationship with CDSP B;

- content from CP B be delivered by CDN A to end-users in country A, even if CP B had no direct relationship with CDSP A.

2.5. Content Delivery Metadata

The tested CDN solutions feature proprietary metadata APIs, but these APIs have not been covered by the tests. We had to configure distribution metadata consistently in the dCDN and the uCDN (e.g., rules to determine upstream source for content acquisition).

Content pre-positioning in the dCDN has not been tested: only dynamic content acquisition has been covered by the experiments.

Proprietary APIs were available for content purge, but those have not been covered by tests.

2.6. Content Acquisition

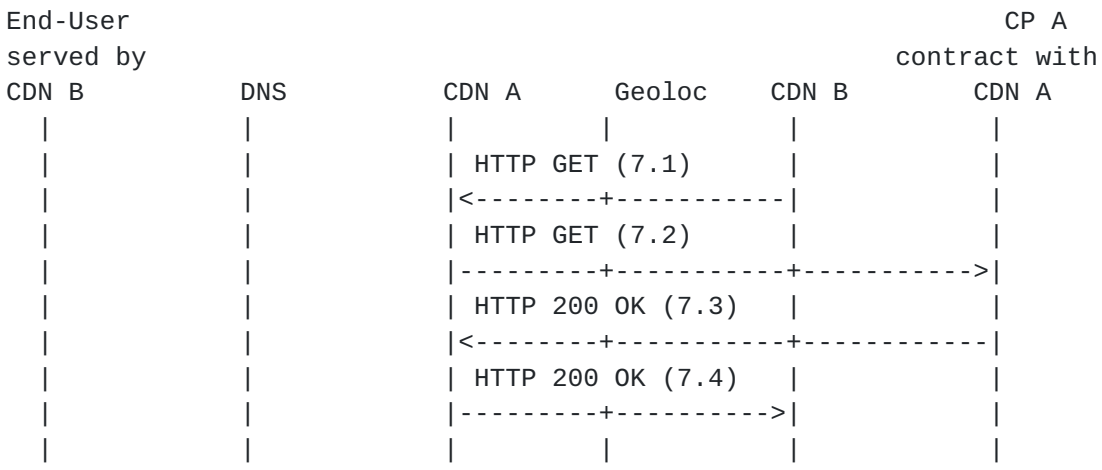
We have used regular HTTP for content acquisition. We have relied on HTTP custom headers to transfer trivial metadata such as content integrity check (MD5 hash).

The correct acquisition and delivery of the requested file has been tested for Adobe Flash and MS HTTP smooth-streaming files.

2.6.1. Content Acquisition by CDN B through CDN A

We describe here the content acquisition operations triggered in case of cache miss, for the test scenario depicted in Figure 2 and Figure 3, with HTTP redirection by CDN A and delivery by CDN B. In this scenario, the dCDN (CDN B) does not have the requested content in cache and must request it to the uCDN (CDN A).

The uCDN treats the dCDN surrogate as an end-user: Figure 6 provides a summary (the involved internal entities of the uCDN are not detailed) of the related content acquisition operations. Section 3.1.3 provides more details on specific issues related to this content acquisition mode.



Pull content acquisition by CDN B through CDN A in case of cache miss
(continuation of Figure 3)

Figure 6

Message details

(7.1) CDN B surrogate or parent cache sends a content acquisition request to the uCDN (CDN A). In the tests, (7.1) was triggered by a cache miss on delivery request. Stated differently, the tests implemented dynamic acquisition, as opposed to content pre-positioning.

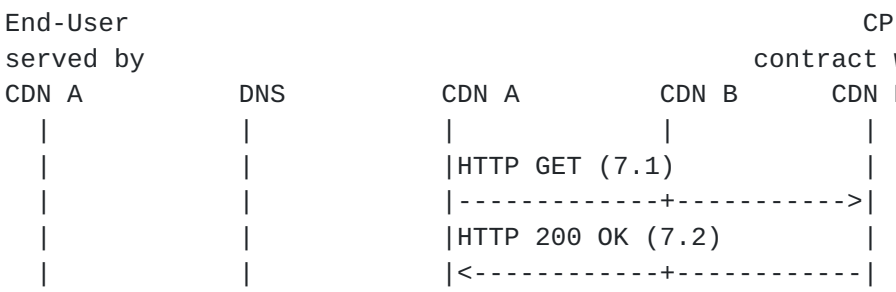
(7.2) CDN A analyzes the request. In case of cache miss, it sends an acquisition request to the CP's origin servers.

(7.3) The CP's origin server authorizes the request and delivers the requested content to CDN A, through an HTTP 200 OK.

(7.4) CDN A delivers the requested content to CDN B surrogate or parent cache, through an HTTP 200 OK.

2.6.2. Content Acquisition by CDN A Directly from CP B

We describe here (Figure 7) the content acquisition operations triggered in case of cache miss, for the test scenario with HTTP redirection by CDN B and delivery by CDN A (Figure 4 and Figure 5). In this scenario, the dCDN (CDN A) does not have the requested content in cache and must request it to CP B.



Pull content acquisition by CDN A directly from content provider's origin servers in case of cache miss (continuation of Figure 5)

Figure 7

Message details

(7.1) CDN A surrogate sends a content acquisition request to an origin server of the CP. In the tests, (7.1) was triggered by a cache miss on a delivery request. Stated differently, the tests implemented dynamic acquisition, as opposed to content pre-positioning.

(7.2) The CP's origin server authorizes the request and delivers the requested content to CDN A, through an HTTP 200 OK.

3. Lessons Learned

For basic interconnection tests, we have relied on extremely limited information exchanges between the two interconnected CDNs and we have configured most CDNI related features manually. This is because, while present CDN technologies support APIs allowing configuration of some of this information, those are difficult to use in multi-vendor environments since:

- they are proprietary APIs;
- they are designed as "internal" APIs and therefore lack the necessary inter-domain security and policy control.

Therefore, those APIs have not been used in these tests.

In the present section, we highlight some of the limitations induced by the lack of standard CDNI interfaces that we have faced in our tests.

One of the insights from this work is that by encoding information

inside the redirection URI, it is possible to communicate some essential CDNI-related information across CDNs "in-band" (i.e., as part of HTTP), rather than communicating it through an out-of-band interface. In these tests, the information communicated in-band was restricted to the most fundamental information, that is, a handle allowing the Downstream CDN to determine where to acquire the content. This was key to achieving multi-CDN operations without any common CDNI "out-of-band" interface supported by existing CDN technologies. This raises an interesting general question: what subset of inter-CDN information is to be communicated between interconnected CDNs in-band (possibly using existing methods) as opposed to communicated via out-of-band interfaces?

3.1. Request Routing

3.1.1. Request-Routing Information and Policies

Because of the lack of CDNI interfaces allowing CDNs to exchange information such as their coverage, capabilities, and performance, we had to configure request-routing policies manually in the CDNs that acted as uCDNs. While this may be tolerable for initial limited deployments of CDNI scenarios with a small number of participants, this is expected to create operational constraints in larger scale deployments.

3.1.2. Iterative and Recursive Redirection

Because of the lack of CDNI interfaces allowing an upstream CDN to query dCDN for how to redirect a request, the tests only covered iterative redirection (i.e. uCDN redirects the user-agent to the dCDN request-routing system, which redirects the user-agent to ...), not recursive redirection.

While iterative redirection allows supporting redirection across CDNs, it has some limitations:

- multiple redirections are exposed to the end-user;
- redirection latency cannot easily be reduced for future requests, through the caching of request-routing decisions;
- some client implementations support a limited number of successive redirections;
- the dCDN cannot reject a redirection, while allowing the uCDN to handle the rejected request.

A standard request-routing API would allow supporting recursive

redirection, which removes these shortcomings.

3.1.3. Request Looping Avoidance

In case of cache-miss, the downstream CDN must fetch the requested content, either through the authoritative CDN, or directly from the CP's origin server.

Consider the situation where the downstream CDN fetches the content from the authoritative CDN, as illustrated in Section 2.6.1. In this case, the authoritative CDN must not redirect the acquisition request to the downstream CDN, because this would create the following request-routing loop: dCDN -> uCDN -> dCDN. Consequently, the upstream CDN must be able to determine that the source of the request is a partner CDN and not a regular end-user. In addition, the upstream CDN must be able to acquire content from the CP's origin server on behalf of the downstream CDN, if necessary: dCDN -> uCDN -> CP.

In the tests, we have successfully solved the request-looping issue, through the use of separated URL spaces for regular users and CDN users, as well as the manual configuration of appropriate request-routing policies for every URL space. In other words, the URL used by regular users to fetch content was different from the one used by the downstream CDN to fetch the content. This way, we eliminated the loops. More automated operation would be required in larger-scale deployments.

3.2. Content Delivery Metadata

CDN technology typically supports APIs allowing creation, update, and deletion of content delivery metadata in the CDN. However, while often similar, those are proprietary and would require custom support. In the tests, passing of the most essential information, i.e., the upstream source for content acquisition, was achieved indirectly via conveying a handle inside the URI and configuring manually in the downstream CDN rules for extracting the upstream source.

The upstream source for content acquisition can be specified through the use of a specific URI scheme. For example, CDN A could use the following scheme: http://cdni.cdna.com/origin-URI to point to a cached copy of the content reachable at "origin-URI". The domain name inside the URI scheme designates the request-routing system of a CDN, and the remainder of the URI defines upstream source for content acquisition: here, the content URI on the CP's origin servers. If the authoritative CDN and the downstream CDN use this URI scheme, the authoritative CDN can easily map the URI that it receives in the end-

user's content request with a valid URI on the downstream CDN. Similarly, the downstream CDN can easily extract a content acquisition URI from the redirection URI.

In our tests, we have configured manually the rules that enabled the authoritative CDN to redirect end-users to a valid URI on the downstream CDN, and the downstream CDN to ingest content from the appropriate upstream source. While this allowed validation of the content distribution model, this solution may not be viable in a production environment, as it imposes constraints on URI structure. In addition, this approach does not support exchange of other distribution metadata (e.g., geo-blocking, content validation,...) which would require to be manually configured in the downstream CDN. In a real-world deployment, the configuration of these policies could rely on information that interconnected CDNs would exchange through a CDNI interface.

3.3. Content Acquisition and Deletion

3.3.1. Content Pre-Positioning in Downstream CDN

CDN technology typically supports APIs that allow triggering of content and metadata pre-positioning in a CDN. However, while often similar, these APIs are proprietary and would require custom support. For this reason content pre-positioning in dCDN was not covered in the tests. While the highest requirements is for support of dynamic acquisition, CDNI use-cases call for support of pre-positioning, which requires a triggering mechanism in a CDNI API.

3.3.2. Content Purge

CDN technology typically supports APIs allowing content purge in a CDN. However, while often similar, these APIs are proprietary and would require custom support. For this reason, content purge was not covered in the tests. There is a strong requirement for content purge in CDNI scenarios, which introduces the need for a purge triggering mechanism in a CDNI API.

4. Acknowledgments

The authors would like to acknowledge the work of Elodie Hemon and Marcin Pilarski on the tests, with the technical support of Sharon Schwartzman and Marc Fiuczynski. They would like to thank Slim Gara, Vincent Lauwers, Emile Stephan, Benoit Gaussen, and Mateusz Dzida for valuable input and discussions. Finally, the authors acknowledge interesting discussions with contributors of the EU FP7 OCEAN project.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

CDN interconnect, as described in this document, has a wide variety of security issues that should be considered. This document focuses on specific experiments for CDN interconnect, and therefore, does not analyze the threats in detail.

7. References

7.1. Normative References

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

7.2. Informative References

[I-D.bertrand-cdni-use-cases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-bertrand-cdni-use-cases-02 (work in progress), July 2011.

[I-D.jenkins-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

[I-D.lefaucheur-cdni-requirements]
Leung, K., Lee, Y., Faucheur, F., Viveganandhan, M., and G. Watson, "Content Distribution Network Interconnection (CDNI) Requirements", draft-lefaucheur-cdni-requirements-02 (work in progress), July 2011.

[RFC3466] Day, M., Cain, B., Tomlinson, G., and P. Rzewski, "A Model for Content Internetworking (CDI)", RFC 3466, February 2003.

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

[RFC3570] Rzewski, P., Day, M., and D. Gilletti, "Content Internetworking (CDI) Scenarios", RFC 3570, July 2003.

Authors' Addresses

Gilles Bertrand (editor)
France Telecom - Orange
38-40 rue du General Leclerc
Issy les Moulineaux 92130
FR

Phone: +33 1 45 29 89 46
Email: gilles.bertrand@orange-ftgroup.com

Francois Le Faucheur
Cisco Systems
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
FR

Phone: +33 4 97 23 26 19
Email: flefauch@cisco.com

Larry Peterson
Verivue, Inc.
2 Research Way
Princeton, NJ 08540
US

Phone: +1 978 303 8032
Email: lpeterson@verivue.com