Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 8, 2012

G. Bertrand
E. Stephan
France Telecom - Orange
G. Watson
T. Burbridge
P. Eardley
BT
K. Ma
Azuki Systems

July 7, 2011

Use Cases for Content Delivery Network Interconnection
draft-bertrand-cdni-use-cases-02

Abstract

Content Delivery Networks (CDNs) are commonly used for improving the footprint and the end-user experience of a content delivery service, at a reasonable cost. This document outlines real world use-cases (not technical solutions) for interconnecting CDNs. It provides the business motivations for CDNI Working Group, which can be used to validate different interconnection arrangements, and requirements of the various CDNI interfaces.

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 January 8, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document now merges input from [I-D.watson-cdni-use-cases] and [I-D.ma-cdni-publisher-use-cases].

Content Delivery Networks (CDNs) are commonly used for improving the footprint and the end-user experience of a content delivery service, at a reasonable cost. This document outlines real world use-cases (not technical solutions) for interconnecting CDNs. It provides the business motivations for CDNI Working Group, which can be used to validate different interconnection arrangements, and requirements of the various CDNI interfaces.

There are many possible combinations for the relationships between the different parties (Network Service Provider (NSP), CDN Provider, Content Service Provider (CSP) and End User) involved in end-to-end content delivery. However, in the context of interconnecting CDNs the key relationships are listed below.

- How the CSP interacts with the CDN provider, so that the CDN delivers content in a manner compliant with CSP's distribution policies.
- How the End User interacts with the CSP and one or more CDNs to request and receive content.
- How the different CDN providers, operating their CDNs, interact with one another to deliver the CSP's content to the End User while continuing to enforce the CSP's distribution policies.

This document describes a number of use cases that motivate CDN Interconnection.

1.1. Terminology

We adopt the terminology described in [I-D.jenkins-cdni-problem-statement], [RFC3466], and [RFC3568], except for the terms defined below.

CDN Provider:

An administrative entity who operates a CDN over a NSP or over the Internet.

Authoritative CDN (aCDN):

A CDN provider contracted by the CSP for delivery of content by its CDN or by its downstream CDNs.

Downstream CDN (dCDN):

A CDN provider which is contracted by an uCDN to achieve the delivery of content to users.

Access CDN:

A CDN that is connected to the end-user's access and has information about the end-user's profile and access capabilities.

Delivering CDN:

The CDN that delivers the requested content asset to the end-user. In particular, the delivering CDN can be an access CDN.

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).

CDN peering: A business relation between two CDN providers based on one or more CDN interconnections.

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 downstream 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.

Asymmetric Distribution:

A distribution scenario where different NSPs have distribution rights to the same content, but at different levels of quality (e.g., high

definition vs. low definition video), which places restrictions on delivery delegation.

1.2. Abbreviations

[Ed. Note: List of abbreviations to be updated later]

- CSP: Content Service Provider
- dCDN: downstream CDN
- ISP: Internet Service Provider
- NSP: Network Service Provider
- PC: Personal Computer
- QoE: Quality of Experience
- QoS: Quality of Service
- SLA: Service Level Agreement
- STB: Set-Top-Box
- uCDN: upstream CDN
- UA: User Agent
- UE: User Equipment
- VoD: Video on Demand
- WiFi: Wireless Fidelity

1.3. High Level Use Cases for Multi-CDN Systems

Content Delivery Networks (CDNs) are used to deliver content because they can:

- improve the experience for the End User; for instance delivery has lower latency and better robustness,
- reduce the operator’s costs; for instance lower delivery cost (reduced bandwidth usage) for cacheable content,
- reduce the Content Service Provider costs, such as datacenter capacity, space, and electricity consumption.

Indeed, many network service providers and enterprise service providers are deploying or have deployed their own CDNs. Despite the potential benefits of interconnecting CDNs, today each CDN is a standalone network. The objective of CDN interconnection is to overcome this restriction: the interconnected CDNs should be able to collectively behave as a single delivery infrastructure.

Let's take an example, as depicted in Figure 1. Two CDN Providers establish a CDN Interconnect. The Content Service Provider CSP-1 reaches an agreement with CDN Provider A for the delivery of its content. CDN Provider A and CDN Provider B agree to interconnect their CDNs. When a User Agent that is connected to CDN Provider B's network requests Content from CSP-1, the Content is actually delivered from CDN-B, because handling of requests for CSP-1's Content has been delegated as part of the CDN Interconnect agreement. The End User benefits through a better quality of experience, because the Content is delivered from a nearby Surrogate. CDN Provider A benefits because it doesn't need to deploy such an extensive CDN, whilst CDN Provider B receives some compensation for the delivery. CSP-1 benefits because it only needs to make one business agreement and one physical connection, with CDN Provider A, but its End Users get a service quality as though it had also gone to the trouble of making a business agreement with CDN Provider B.

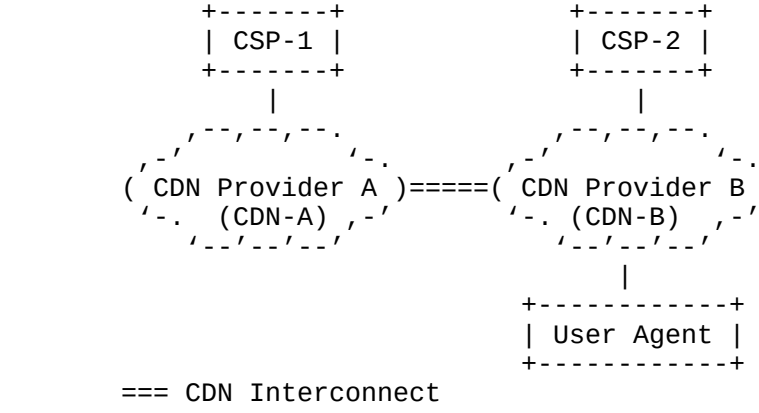


Figure 1

To extend the example, another Content Service Provider, CSP-3, may also reach an agreement with CDN Provider A. But it does not want its Content to be distributed by CDN Provider B, for example, CSP-3 may not have distribution rights in the country where CDN Provider B operates. This example illustrates that policy considerations are an

important part of CDNI.

This document identifies three main motivations for a CDN Provider to interconnect its CDN:

- CDN Footprint Extension Use Cases (Section 2)
- CDN Offload Use Cases (Section 3)
- CDN Capability Use Cases (Section 4)

1.4. The Need for CDNI Standards

Existing CDN interfaces are proprietary and an external CDN typically cannot use them, especially if the two CDNs rely on different solutions. Nevertheless, [I-D.bertrand-cdni-experiments] shows that some level of CDN interconnection can be achieved experimentally without standardized interfaces between the CDNs. The methods used in these experiments are hardly usable in an operational context, because they suffer from several limitations in terms of functionalities, scalability, and security level.

The aim of the CDNI standards work is therefore to overcome such shortcomings; a full list of requirements is being developed in [I-D.lefaucheur-cdni-requirements].

2. Footprint Extension Use Cases

Footprint extension is expected to be a major use case for CDN interconnection.

2.1. Geographic Extension

In this use case, the CDN Provider wants to extend the geographic distribution that it can offer CSPs, without

- compromising the quality of delivery
- attracting transit and other network costs by serving from geographically or topologically remote surrogates.

If there are several CDN Providers that have a geographically limited footprint (e.g., restricted to one country), or do not serve all end-users in a geographic area, then interconnecting their CDNs enables CDN Providers to provide their services beyond their own footprint.

As an example, suppose a French CSP wants to distribute its TV

programs to End Users located in various countries in Europe and North Africa. It asks a French CDN Provider to deliver the content. The French CDN Provider's network only covers France, so it makes an agreement with another CDN Provider that covers North Africa. Overall, from the CSP's perspective the French CDN Provider provides a CDN service for both France and North Africa.

In addition to video, this use case applies to other types of content such as automatic software updates (browser updates, operating system patches, virus database update, etc).

2.2. Region to Region Interconnection

In the previous section, we have described the case of geographic extension between CDNs operated by different entities. A large CDN Provider may also operate CDNs from several subsidiaries (which may rely on different CDN solutions, see Section 4.2). In certain circumstances, the CDN Provider needs to make its CDNs interoperate to provide a consistent service to its customers on its whole footprint.

2.3. Nomadic Users

In this scenario a CSP wishes to allow users who move to other geographic regions to continue to access their content. The motivation in this case is to allow nomadic users to maintain access, rather than to allow all residents within a region access to the content.

This use case covers situations like users moving between different CDN Providers within the same geographic region, or users switching between different devices, as discussed in Section 4.

2.4. Delivery Restrictions

The content distribution policies that a CSP attaches to a content asset depend on many criteria. Distribution rights for audiovisual content are often negotiated using a combination of temporal licensing (e.g., available for 24 hours, available 28 days after DVD release, etc.), resolution-based licensing (e.g., high definition vs. standard definition), and geo- location-based licensing (e.g., per country).

"Geo-blocking" rules may specify:

- the geographic regions where content can be delivered from (i.e. the location of the Surrogates), or

- o geographic locations where content can be delivered to (i.e., the location of the End Users).

Hence, the exchange through the CDN interconnection of information for controlling the footprint of the delivery is an important use case.

The delivery of content may be further influenced by policies which may include time-based rules that specify:

- o an activation time (i.e., the time when the content should become available for delivery),
- o a deactivation time (i.e., time after which the content should no longer be delivered), or
- o an expiration time (i.e., the time at which the content files should be expunged from all CDN storage).

The delivery of content may be further influenced by policies which may include quality of service rules that specify:

- o the maximum resolution deliverable to specific devices,
- o the maximum resolution deliverable though a specific NSP, or
- o the maximum resolution deliverable to users based on their subscription levels.

The enforcement of CSP licensing rules when making CDN delegation decisions is another important use case for CDN interconnection.

3. Offload Use Cases

3.1. Overload Handling and Dimensioning

A CDN is likely to be dimensioned to support the prime-time traffic. However, unexpected spikes in content popularity may drive load beyond the expected peak. The prime recurrent time peaks of content distribution may differ between two CDNs. Taking advantage of the different traffic peak times, a CDN may interconnect with another CDN to increase its effective capacity during the peak of traffic. This brings dimensioning savings to the CDNs as they can use the resources of each other during their peaks of activity.

Offload also applies to planned situations where a CDN Provider needs CDN capacities in a particular region during a short period of time.

For example, a CDN can offload traffic to another CDN during a specific maintenance operation or for covering the distribution of a special event. For instance, consider a TV-channel which has exclusive distribution rights on a major event, such as a celebrities' wedding, or a major sport competitions. The CDNs that the TV-channel uses for delivering the content related to this event are likely to experience a flash crowd during the event and to need offloading traffic, while other CDNs will support a more usual traffic load and be able to handle the offloaded traffic load.

3.2. Resiliency

3.2.1. Failure of Content Delivery Resources

It is important for CDNs to be able to guarantee service continuity during partial failures (e.g., failure of some Surrogates). In partial failure scenarios, a CDN Provider could redirect some requests towards another CDN, which must be able to serve the redirected requests or, depending on traffic management policies, to forward these requests to the CSP's origin server.

3.2.2. Failure of Content Acquisition

Source content acquisition is typically handled in one of two ways:

- CDN origin, where a downstream CDN acquires content from an upstream CDN, and the authoritative CDN acquires content from an origin server of the CSP, or

- CSP origin, where the CDNs acquire content directly from an origin server of the CSP.

Resiliency may be required against failure to ingest content from the CSP. If a CDN is unable to retrieve the content, it may be that the CSP's origin server is inaccessible to only this CDN, in which case redirection of the end-users to an alternative CDN may circumvent the problem. A CSP may also choose to specify one or more backup origin servers.

3.3. Branding Consideration

There are situations where one CDN Provider cannot or does not want to operate all the functions of a CDN. For instance, it always acts as an uCDN and offloads the content delivery to dCDNs, i.e., it uses the surrogates of other CDSPs. In this model, the uCDN acquires content and receives the initial routing requests from the user agent; whereas, the dCDNs operate the content delivery functions. The uCDN also retrieves and presents the logging for the CSP.

Preserving branding elements could interest the CSP or CDSPs. The CSP might desire to offer content services under its name, even if the associated CDN service involves other organizations. Therefore, the CSP could request that the name of the CDSPs does not appear in the URLs. Similarly, in offload situations, the uCDN might want to offer CDN services under its own branding. This highlight a requirement for exchanging branding related constraints over a CDNI.

4. CDN Capability Use Cases

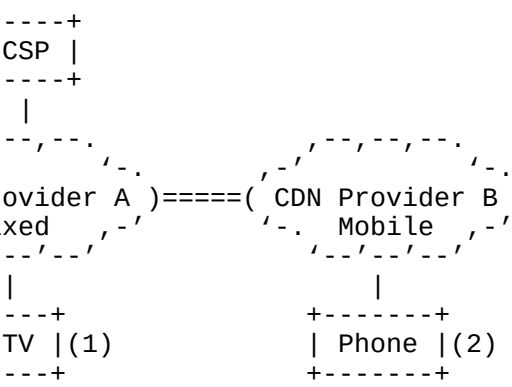
4.1. Device and Network Technology Extension

In this use case, the CDN Provider may have the right geographic footprint, but wishes to support the delivery of content to alternative devices, such as smartphones connected to a mobile network. In this case, the CDN Provider may federate with another CDN Provider that offers service to these devices.

Consider the scenario shown in Figure 2. In this example, a nomadic user switches from a TV going through a cable provider to a smartphone going through a mobile operator. The CDN Provider on the cable network may wish to delegate delivery of Content to the CDN Provider on the mobile network. There are several possible differences that may arise in this use case compared with the ones discussed earlier, for example:

- the phone may require the Content at lower resolution than the TV;
- the CSP may want to license only lower resolution Content to CDN Provider 2;
- the CSP may not want CDN Provider 2 to deliver Content if the connection quality is below some threshold;
- the CSP may want to tailor the Content in some special way depending on whether the End User is on cable or mobile, for example, different adverts / DRMs / codecs / container formats / delivery protocols...

These examples suggest the requirement for Asymmetric Distribution of Content across the CDN interconnect. In the nomadic scenario, the switch of CDN should be as seamless as possible from the End User’s perspective.



=== CDN Interconnect

Fixed-Mobile Session Shifting

Figure 2

4.2. Technology and Vendor Interoperability

A CDN Provider may deploy a new CDN to run alongside its existing CDN, as a simple way of migrating its CDN service to a new technology. A CDN Provider may have a multi-vendor strategy for its CDN deployment. A CDN Provider may want to deploy a separate CDN for a particular CSP or a specific network. In all these circumstances, CDNI benefits the CDN Provider, as it simplifies or automates some inter-CDN operations (e.g., migrating the request routing function progressively).

4.3. QoE and QoS Improvement

Some CSPs are willing to pay a premium for enhanced delivery of Content to their End Users. In some cases, even if the CDN Provider could deliver the content to the end users, it cannot meet the CSP's service level agreement. So, it makes a CDN Interconnect agreement with another CDN Provider that can meet the SLA, for instance an Access CDN, which is able to deliver content from Surrogates located closer to the end-user.

5. Acknowledgments

The authors would like to thank Francois Le Faucheur and Ben Niven-Jenkins for lively discussions.

They also thank the contributors of the EU FP7 OCEAN and ETICS projects for valuable inputs.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

CDN interconnect, as described in this document, has a wide variety of security issues that should be considered. The security issues fall into three general categories:

- CSP Trust: where the CSP may have negotiated service level agreements for delivery quality of service with the uCDN, and/or configured distribution policies (e.g., geo-restrictions, availability windows, or other licensing restrictions), which it assumes will be upheld by dCDNs to which the uCDN delegates requests. Furthermore, billing and accounting information must be aggregated from dCDNs with which the CSP may have no direct business relationship. These situtations where trust is delegated must be handled in a secure fashion to ensure CSP confidence in the CDN interconnection.

- Client Transparency: where the client device or application which connects to the CDN must be able to interact with any dCDN using its existing security and DRM protocols (e.g., cookies, certificate-based authentication, custom DRM protocols, URL signing algorithms, etc.) in a transparent fashion.

- CDN Infrastructure Protection: where the dCDNs must be able to identify and validate delegated requests, in order to prevent unauthorized use of the network and to be able to properly bill for delivered content. A dCDN may not wish to advertise that it has access to or is carrying content for the uCDN or CSP, especially if that information may be used to enhance denial of service attacks. In general, CDNI interfaces and protocols should minimize overhead for dCDNs.

This document focuses on the motivational use cases for CDN interconnect, and does not analyze these threats in detail.

8. References

8.1. Normative References

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

8.2. Informative References

[I-D.bertrand-cdni-experiments]
Bertrand, G., Faucheur, F., and L. Peterson, "Content Distribution Network Interconnection (CDNI) Experiments", draft-bertrand-cdni-experiments-00 (work in progress), February 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]
Faucheur, F., Viveganandhan, M., Watson, G., and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-lefaucheur-cdni-requirements-01 (work in progress), March 2011.

[I-D.ma-cdni-publisher-use-cases]
Nair, R. and K. Ma, "Content Distribution Network Interconnection (CDNI) Publisher Use", draft-ma-cdni-publisher-use-cases-00 (work in progress), March 2011.

[I-D.watson-cdni-use-cases]
Watson, G., "CDN Interconnect Use Cases", draft-watson-cdni-use-cases-00 (work in progress), January 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.

Authors' Addresses

Gilles Bertrand
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

Stephan Emile
France Telecom - Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: emile.stephan@orange-ftgroup.com

Grant Watson
BT
pp GDC 1 PP14, Orion Building, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: grant.watson@bt.com

Trevor Burbridge
BT
B54 Room 70, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: trevor.burbridge@bt.com

Philip Eardley
BT
B54 Room 77, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: philip.eardley@bt.com

Kevin Ma
Azuki Systems
43 Nagog Park
Acton, MA 01720
USA

Phone: +1 978 844 5100
Email: kevin.ma@azukisystems.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

B. Davie, Ed.
Cisco Systems, Inc.
L. Peterson, Ed.
Verivue, Inc.
October 31, 2011

Framework for CDN Interconnection
draft-davie-cdni-framework-01

Abstract

This document presents a framework for Content Distribution Network Interconnection (CDNI). The purpose of the framework is to provide an overall picture of the problem space of CDNI and to describe the relationships among the various components necessary to interconnect CDNs. CDN Interconnection requires the specification of several interfaces and mechanisms to address issues such as request routing, metadata exchange, and the acquisition of content by one CDN from another. The intent of this document is to outline what each interface needs to accomplish, and to describe how these interfaces and mechanisms fit together, while leaving their detailed specification to other documents.

Table of Contents

1. Introduction

The interconnection of Content Distribution Networks (CDNs) is motivated by several use cases, such as those described in [I-D.ietf-cdni-use-cases]. The overall problem space for CDN Interconnection is described in [I-D.ietf-cdni-problem-statement]. The purpose of this document is to provide an overview of the various components necessary to interconnect CDNs. CDN Interconnection requires the specification of several interfaces and mechanisms to address issues such as request routing, metadata exchange, and the acquisition of content by one CDN from another. The intent of this document is to describe how these interfaces and mechanisms fit together, leaving their detailed specification to other documents. We make extensive use of message flow examples to illustrate the operation of interconnected CDNs, but these examples should be considered illustrative rather than prescriptive.

1.1. Terminology

This document draws freely on the terminology defined in [RFC3466] and [I-D.ietf-cdni-problem-statement].

We also introduce the following terms:

CDN Domain: a host name (FQDN) at the beginning of a URL, representing a set of content that is served by a given CDN. For example, in the URL http://cdn.csp.com/...rest of url..., the CDN domain is cdn.csp.com.

Distinguished CDN Domain: a CDN domain that is allocated by a CDN for the purposes of communication with a peer CDN, but which is not found in client requests. Such CDN domains may be used for inter-CDN acquisition, or as redirection targets, and enable a CDN to distinguish a request from a peer CDN from an end-user request.

Recursive CDNI request routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can query the Downstream CDN Request Routing system via the CDNI Request Routing interface (or use information cached from earlier similar queries) to find out how the Downstream CDN wants the request to be redirected, which allows the Upstream CDN to factor in the Downstream CDN response when redirecting the user agent. This approach is referred to as "recursive" CDNI request routing. Note that the Downstream CDN may elect to have the request redirected directly to a Surrogate inside the Downstream CDN, to the Request-Routing System of the Downstream CDN, to another CDN, or to any other system that the Downstream CDN sees as fit for handling the redirected request.

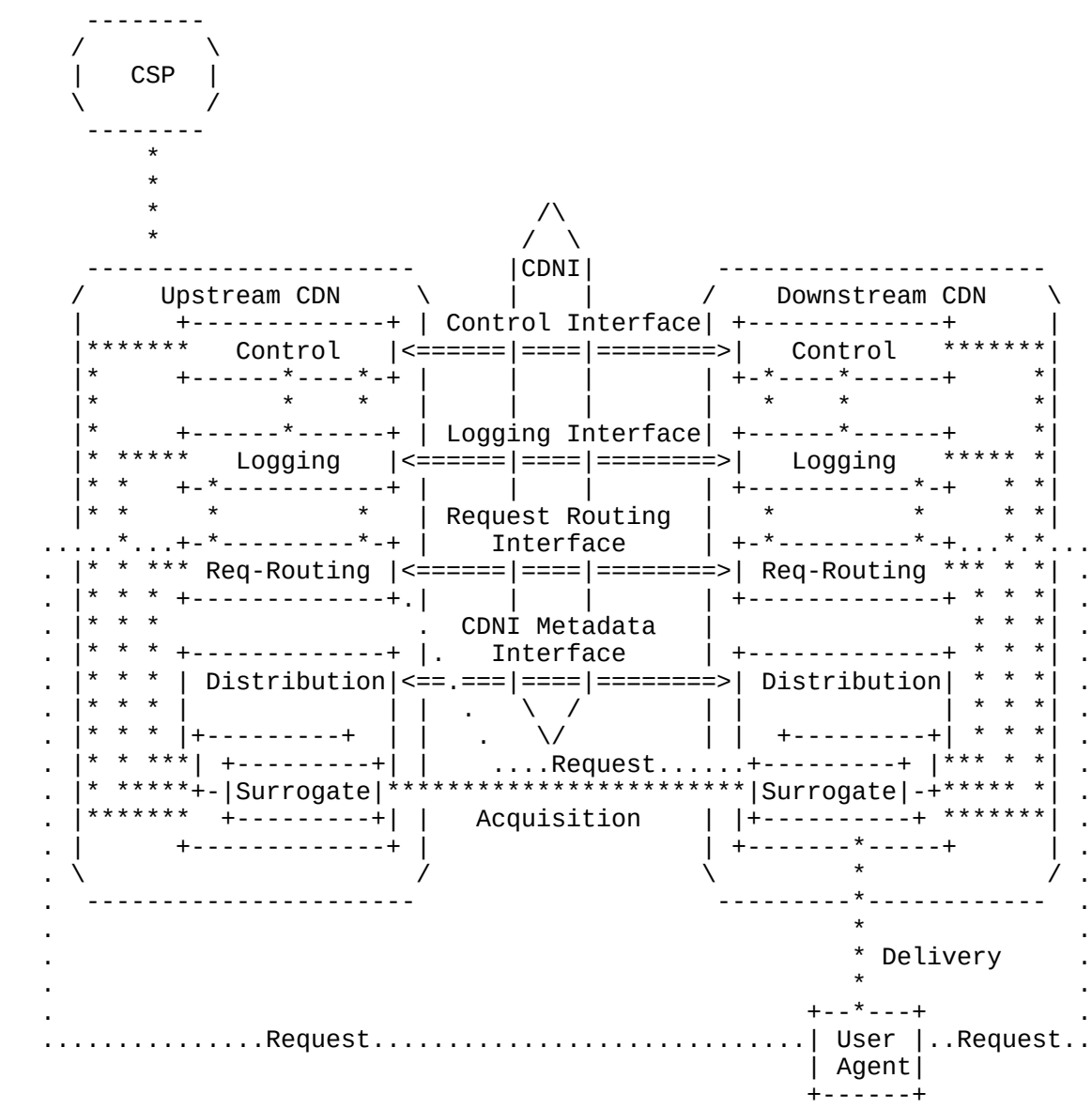
Iterative CDNI Request Routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can base its redirection purely on a local decision (and without attempting to take into account how the Downstream CDN may in turn redirect the user agent). In that case, the Upstream CDN redirects the request to the request routing system in the Downstream CDN, which in turn will decide how to redirect that request: this approach is referred to as "iterative" CDNI request routing.

Synchronous CDNI operations: operations between CDNs that happen during the process of servicing a user request, i.e. between the time that the user agent begins its attempt to obtain content and the time at which that request is served.

Asynchronous CDNI operations: operations between CDNs that happen independently of any given user request, such as advertisement of footprint information or pre-positioning of content for later delivery.

1.2. Reference Model

This document uses the reference model in Figure 1 as originally created in [I-D.ietf-cdni-problem-statement].



<==> interfaces inside the scope of CDNI

**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Model and CDNI Interfaces

We note that while some interfaces in the reference model are "out of scope" for the CDNI WG (in the sense that there is no need to define new protocols for those interfaces) we still need to refer to them in this document to explain the overall operation of CDNI.

We also note that, while we generally show only one uCDN serving a given CSP, it is entirely possible that multiple uCDNs can serve a single CSP. In fact, this situation effectively exists today in the sense that a single CSP can connect to more than one CDN today.

Definitions of the four CDNI interfaces follow. More discussion of these interfaces appears in Section 4.

- Control Interface: Operations to discover, initialize, and parameterize the other CDNI interfaces. Once established, all runtime control over CDNI behavior is under the purview of one of these other interfaces.

- Request Routing Interface: Operations to determine what CDN (and optionally what surrogate within a CDN) is to serve end-user's requests. May include a combination of:

  * Asynchronous operations to exchange routing information (e.g., the network footprint served by a given CDN) that enables CDN selection for subsequent user requests; and

  * Synchronous operations to select a delivery CDN (surrogate) for a given user request.

- Metadata Interface: Operations to communicate metadata that governs the how content is delivered by interconnected CDNs. Examples of CDNI metadata include geo-blocking directives, availability windows, access control mechanisms, and purge directives. May include a combination of:

  * Asynchronous operations to exchange metadata that govern subsequent user requests for content; and

  * Synchronous operations that govern behavior for a given user request for content.

- Logging Interface: Operations that allow interconnected CDNs to exchange relevant activity logs. May include a combination of:

  * Real-time exchanges, suitable for runtime traffic monitoring; and

* Off-line exchanges, suitable for analytics and billing.

1.3. Structure Of This Document

The remainder of this document is organized as follows:

- Section 2 describes some essential building blocks for CDNI, notably the various options for redirecting user requests to a given CDN.
- Section 3 provides a number of illustrative examples of various CDNI operations.
- Section 4 describes the functionality of the four main CDNI interfaces.
- Section 5 shows how various deployment models of CDNI may be achieved using the defined interfaces.
- Section 6 describes the trust model of CDNI and the issues of transitive trust in particular that CDNI raises.

2. Building Blocks

2.1. Request Redirection

At its core, CDN Interconnection requires the redirection of requests from one CDN to another. For any given request that is received by an upstream CDN, it will either respond to the request directly, or somehow redirect the request to a downstream CDN. Two main mechanisms are available for redirecting a request to a downstream CDN. The first leverages the DNS name resolution process and the second uses in-protocol redirection mechanisms such as the HTTP 302 redirection response. We discuss these below as background before discussing some examples of their use in Section 3.

2.1.1. DNS Redirection

DNS redirection is based on returning different IP addresses for the same DNS name, for example, to balance server load or to account for the client’s location in the network. A DNS server, sometimes called the Local DNS (LDNS), resolves DNS names on behalf of an end-user. The LDNS server in turn queries other DNS servers until it reaches the authoritative DNS server for the CDN-domain. The network operator typically provides the LDNS server, although the user is free to choose other DNS servers (e.g., OpenDNS, Google Public DNS).

The advantage of DNS redirection is that it is completely transparent to the end user--the user sends a DNS name to the LDNS server and gets back an IP address. On the other hand, DNS redirection is problematic because the DNS request comes from the LDNS server, not the end-user. This may affect the accuracy of server selection that is based on the user's location. The transparency of DNS redirection is also a problem in that there is no opportunity to modify the path component of the URL being accessed by the client. We consider two main forms of DNS redirection: simple and CNAME-based.

In simple DNS redirection, the authoritative DNS server for the name simply returns an IP address from a set of possible IP addresses. The answer is chosen from the set based on characteristics of the set (e.g., the relative loads on the servers) or characteristics of the client (e.g., the location of the client relative to the servers). Simple redirection is straightforward. The only caveats are (1) there is a limit to the number of delivery nodes a single DNS server can manage; and (2) DNS responses are cached by downstream servers so the TTL on the response must be set to an appropriate value so as to preserve the timeliness of the redirection.

In CNAME-based DNS redirection, the authoritative server returns a CNAME response to the DNS request, telling the LDNS server to restart the name lookup using a new name. A CNAME is essentially a symbolic link in the DNS namespace, and like a symbolic link, redirection is transparent to the client--the LDNS server gets the CNAME response and re-executes the lookup. Only when the name has been resolved to an IP address does it return the result to the user. Note that DNAME would be preferable to CNAME if it becomes widely supported.

2.1.2. HTTP Redirection

HTTP redirection makes use of the "302" redirection response of the HTTP protocol. This response contains a new URL that the application should fetch instead of the original URL. By changing the URL appropriately, the server can cause the user to redirect to a different server. The advantages of 302 redirection are that (1) the server can change the URL fetched by the client to include, for example, both the DNS name of the particular server to use, as well as the original HTTP server that was being accessed; and (2) the client sends the HTTP request to the server, so that its IP address is known and can be used in selecting the server.

The disadvantages of HTTP redirection are (1) it is visible to the application, so it requires application support and may affect the application behavior (e.g., web browsers will not send cookies if the URL changes to a different domain); (2) HTTP is a heavy-weight protocol layered on TCP so it has relatively high overhead; and (3)

the results of HTTP redirection are not cached so that all redirections must go through to the server.

3. Overview of CDNI Operation

To provide a big-picture overview of the various components of CDN Interconnection, we walk through a "day in the life" of a content item that is made available via a pair of interconnected CDNs. This will serve to illustrate many of the functions that need to be supported in a complete CDNI solution. We give examples using both DNS-based and HTTP-based redirection. We begin with very simple examples and then how additional capabilities, such as recursive request redirection and content removal, might be added.

Before walking through some specific examples, we present a high-level view of the operations that may take place. This high-level overview is illustrated in Figure 2. Note that most operations will involve only a subset of all the messages shown below, and that the order and number of operations may vary considerably, as more detailed examples illustrate below.

The following shows Operator A as the upstream CDN (uCDN) and Operator B as the downstream CDN (dCDN), where the former has a relationship with a content provider and the latter being the best CDN to deliver content to the end-user. The interconnection relationship may be symmetric between these two CDN operators, but for simplicity we show the interaction in one direction only.

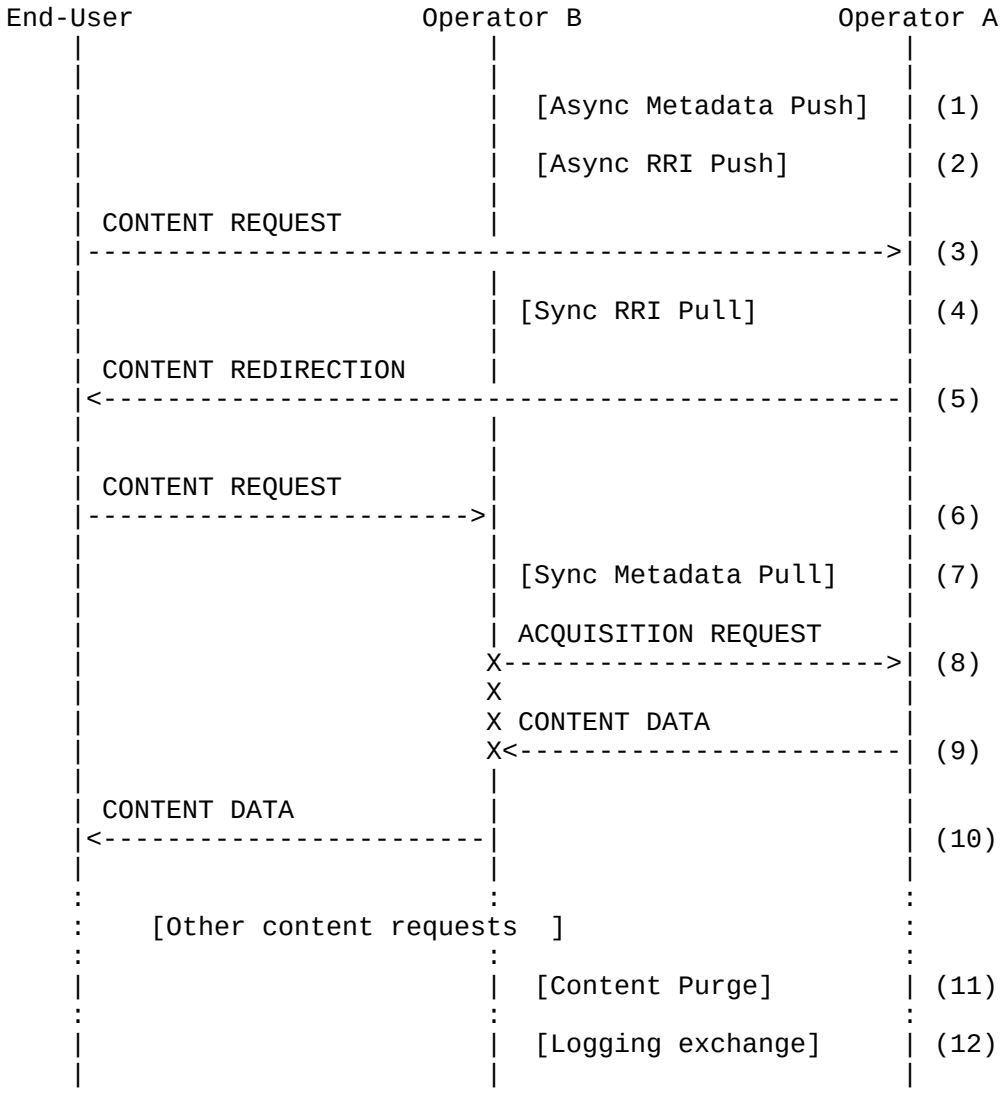


Figure 2: Overview of Operation

The operations shown in the Figure are as follows:

1. Prior to any content request, metadata may be asynchronously pushed from uCDN to dCDN so that it is available in readiness for later content requests.

2. dCDN may advertise information relevant to its delivery capabilities (e.g. geographic footprint, reachable address prefixes) prior to any content requests being redirected.

3. A content request from a user agent arrives at uCDN.

4. uCDN may synchronously request information from dCDN regarding its delivery capabilities to decide if dCDN is a suitable target for redirection of this request.

5. uCDN redirects the request to dCDN by sending some response (DNS, HTTP) to the user agent.

6. The user agent requests the content from dCDN.

7. dCDN may synchronously request metadata related to this content from uCDN, e.g. to decide whether to serve it.

8. If the content is not already in a suitable cache in dCDN, dCDN may acquire it from uCDN.

9. The content is delivered to dCDN from uCDN.

10. The content is delivered to the user agent by dCDN.

11. Some time later, perhaps at the request of the CSP (not shown) uCDN may instruct dCDN to purge the content to ensure it is not delivered again.

12. After one or more content delivery actions by dCDN, a log of delivery actions may be provided to uCDN.

The following sections show some more specific examples of how these operations may be combined to perform various delivery, control and logging operations across a pair of CDNs.

3.1. Preliminaries

Initially, we assume that there is at least one CSP that has contracted with an upstream CDN (uCDN) to deliver content on its behalf. We are not particularly concerned with the interface between the CSP and uCDN, other than to note that it is expected to be the same as in the "traditional" (non-interconnected) CDN case. Existing mechanisms such as DNS CNAMEs or HTTP redirects (Section 2) can be used to direct a user request for a piece of content from the CSP towards the CSP’s chosen upstream CDN.

We use the term "CDN-domain" to refer to the host name (a FQDN) at

the beginning of each URL. We assume Operator A provides an upstream CDN that serves content on behalf of a CSP with CDN-domain cdn.csp.com. We assume that Operator B provides a downstream CDN. An end user at some point makes a request for URL

http://cdn.csp.com/...rest of url...

It may well be the case that cdn.csp.com is just a CNAME for some other CDN-domain (such as csp.op-a.net). Nevertheless, the HTTP request in the examples that follow is assumed to be for the example URL above.

Our goal is to enable content identified by the above URL to be served by the CDN of operator B. In the following sections we will walk through some scenarios in which content is served, as well as other CDNI operations such as the removal of content from a downstream CDN.

3.2. HTTP Redirect Example

In this section we walk through a simple, illustrative example using HTTP redirection from uCDN to dCDN. The example also assumes the use of HTTP redirection inside uCDN and dCDN; however, this is independent of the choice of redirection approach across CDNs, so an alternative example could be constructed still showing HTTP redirection from uCDN to dCDN but using DNS for handling of request inside each CDN.

We assume for this example that Operators A and B have established an agreement to interconnect their CDNs, with A being upstream and B being downstream. (It is likely that the agreement would be made in both directions, but we focus on just one here for clarity.)

The operators agree that a CDN-domain peer-a.op-b.net will be used as the target of redirections from uCDN to dCDN. The name of this domain must be communicated by some means to each CDN. (This could be established out-of-band or via a CDNI interface.) We refer to this domain as a "distinguished" CDN domain to convey the fact that its use is limited to the interconnection mechanism; such a domain is never embedded in URLs that end-users request.

The operators must also agree on some distinguished CDN-domain that will be used for inter-CDN acquisition of CSP’s content from uCDN by dCDN. In this example, we’ll use op-b-acq.op-a.net.

The operators must also exchange information regarding which requests dCDN is prepared to serve. For example, dCDN may be prepared to serve requests from clients in a given geographical region or a set

of IP address prefixes. This information may again be provided out of band or via a defined interface.

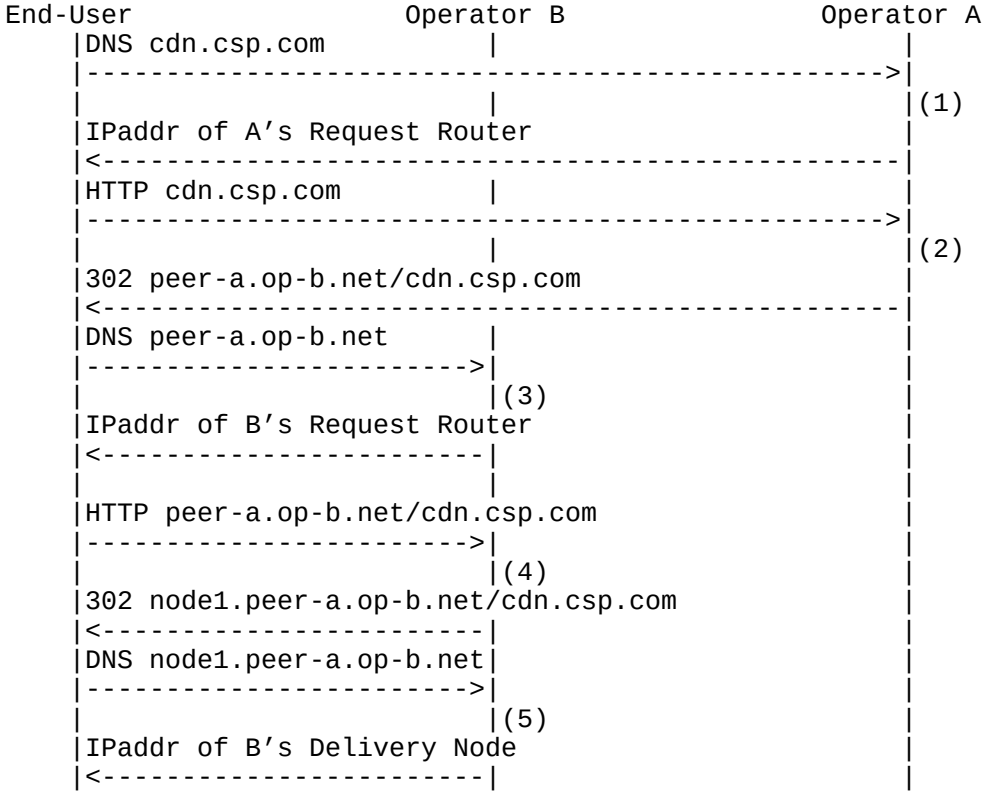
DNS must be configured in the following way:

- The content provider must be configured to make operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).
- Operator A must be configured so that a DNS request for op-b-acq.op-a.net returns a request router in Operator A.
- Operator B must be configured so that a DNS request for peer-a.op-b.net/cdn.csp.com returns a request router in Operator B.

Figure 3 illustrates how a client request for

http://cdn.csp.com/...rest of url...

is handled.



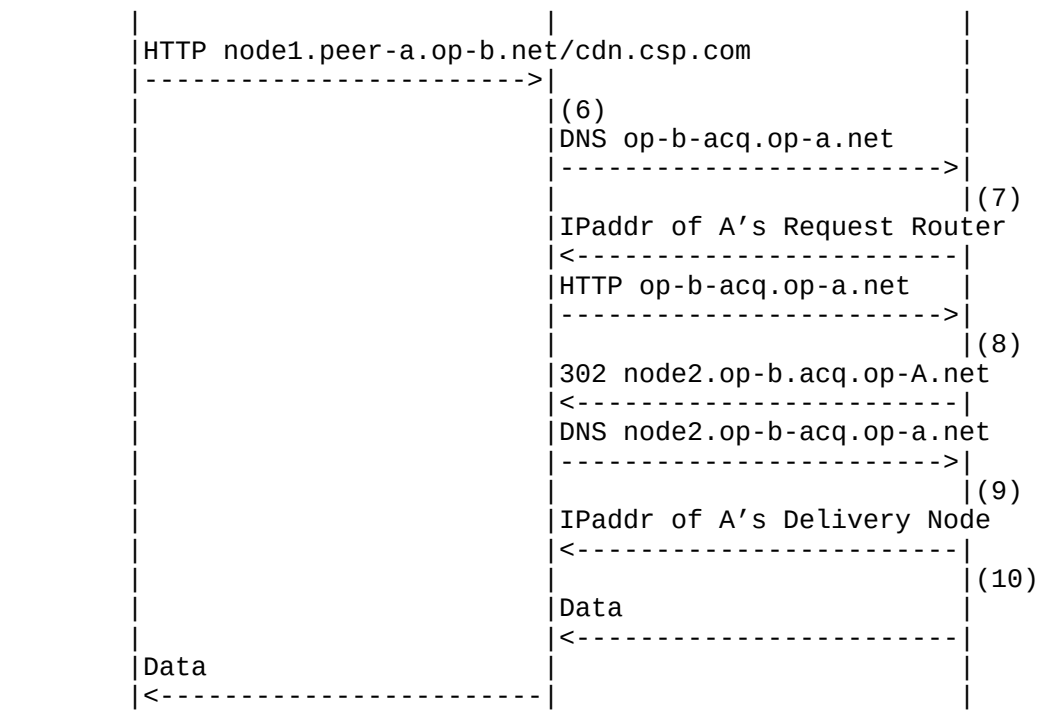


Figure 3: Request Trace for HTTP redirection method

The steps illustrated in the figure are as follows:

1. A DNS resolver for Operator A processes the DNS request for its customer based on CDN-domain cdn.csp.com. It returns the IP address of a request router in Operator A.

2. A Request Router for Operator A processes the HTTP request and recognizes that the end-user is best served by another CDN--specifically one provided by Operator B--and so it returns a 302 redirect message for a new URL constructed by "stacking" Operator B's distinguished CDN-domain (peer-a.op-b.net) on the front of the original URL. (Note that more complex URL manipulations are possible, such as replacing the initial CDN-domain by some opaque handle.)

3. The end-user does a DNS lookup using Operator B's distinguished CDN-domain (peer-a.op-b.net). B's DNS resolver returns the IP address of a request router for Operator B. Note that if request routing within dCDN was performed using DNS instead of HTTP redirection, B's DNS resolver would also behave as the request router and directly return the IP address of a delivery node.

4. The request router for Operator B processes the HTTP request and selects a suitable delivery node to serve the end-user request, and returns a 302 redirect message for a new URL constructed by replacing the hostname by a subdomain of the Operator B's distinguished CDN-domain that points to the selected delivery node.

5. The end-user does a DNS lookup using Operator B's delivery node subdomain (node1.peer-a.op-b.net). B's DNS resolver returns the IP address of the delivery node.

6. The end-user requests the content from B's delivery node. In the case of a cache hit, steps 6, 7, 8, 9 and 10 below do not happen, and the content data is directly returned by the delivery node to the end-user. In the case of a cache miss, the content needs to be acquired by dCDN from uCDN (not the CSP). The distinguished CDN-domain peer-a.op-b.net indicates to dCDN that this content is to be acquired from uCDN; stripping the CDN-domain reveals the original CDN-domain cdn.csp.com and dCDN may verify that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for an inter-CDN acquisition CDN-domain as agreed above (in this case, op-b-acq.op-a.net).

7. Operator A's DNS resolver processes the DNS request and returns the IP address of a request router in operator A.

8. The request router for Operator A processes the HTTP request from Operator B delivery node. Operator A request router recognizes that the request is from a peer CDN rather than an end-user because of the dedicated inter-CDN acquisition domain (op-b-acq.op-a.net). (Note that without this specially defined inter-CDN acquisition domain, operator A would be at risk of redirecting the request back to operator B, resulting in an infinite loop). The request router for Operator A selects a suitable delivery node in uCDN to serve the inter-CDN acquisition request and returns a 302 redirect message for a new URL constructed by replacing the hostname by a subdomain of the Operator A's distinguished inter-CDN acquisition domain that points to the selected delivery node.

9. Operator A DNS resolver processes the DNS request and returns the IP address of the delivery node in operator A.

10. Operator A serves content for the requested CDN-domain to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has

been registered, and determines the content provider that owns the origin server. It may also perform its own content acquisition steps if needed before returning the content to dCDN.

3.2.1. Comments on the example

The main advantage of this design is that it is simple: each CDN need only know the distinguished CDN-domain for each peer, with the upstream CDN "pushing" the downstream CDN-domain onto the URL as part of its redirect (step 2) and the downstream CDN "popping" its CDN-domain off the URL to expose a CDN-domain that the upstream CDN can correctly process. Neither CDN needs to be aware of the internal structure of the other's URLs. Moreover, the inter-CDN redirection is entirely supported by a single HTTP redirect; neither CDN needs to be aware of the other's internal redirection mechanism (i.e., whether it is DNS or HTTP based).

One disadvantage is that the end-user's browser is redirected to a new URL that is not in the same domain of the original URL. This has implications on a number of security or validation mechanisms sometimes used on endpoints. For example, it is important that any redirected URL be in the same domain (e.g., csp.com) if the browser is expected to send any cookies associated with that domain. As another example, some video players enforce validation of a cross domain policy that needs to allow for the domains involved in the CDN redirection. These problems are generally soluble, but the solutions complicate the example, so we do not discuss them further in this version of the draft.

We note that this example begins to illustrate some of the interfaces that may be required for CDNI, but does not require all of them. For example, obtaining information from dCDN regarding the set of client IP addresses or geographic regions it might be able to serve is an aspect of the request routing interface. Important configuration information such as the distinguished names used for redirection and inter-CDN acquisition could also be conveyed via a CDNI interface (e.g., perhaps the control interface). The example also shows how existing HTTP-based methods suffice for the acquisition interface. Arguably, the absolute minimum metadata required for CDNI is the information required to acquire the content, and this information was provided "in-band" in this example by means of the URI handed to the client in the HTTP 302 response. Hence, there is no explicit metadata interface invoked in this example. There is also no explicit logging interface discussed in this example.

We also note that the step of deciding when a request should be redirected to dCDN rather than served by uCDN has been somewhat

glossed over. It may be as simple as checking the client IP address against a list of prefixes, or it may be considerably more complex, involving a wide range of factors, such as the geographic location of the client (perhaps determined from a third party service), CDN load, or specific business rules.

This example uses the "iterative" CDNI request routing approach. That is, uCDN performs part of the request routing function to determine that dCDN should serve the request, and then redirects the client to a request router in dCDN to perform the rest of the request routing function. If request routing is performed in the dCDN using HTTP redirection, this translates in the end-user experiencing two successive HTTP redirections. By contrast, the alternative approach of "recursive" CDNI request routing effectively coalesces these two successive HTTP redirections into a single one, sending the end-user directly to the right delivery node in the dCDN. This "recursive" CDNI request routing approach is discussed in the next section.

3.3. Recursive Redirection Example

The following example builds on the previous one to illustrate the use of the Request Routing interface to enable "recursive" CDNI request routing. We build on the HTTP-based redirection approach because it illustrates the principles and benefits clearly, but it is equally possible to perform recursive redirection when DNS-based redirection is employed.

In contrast to the prior example, the operators need not agree in advance on a CDN-domain to serve as the target of redirections from uCDN to dCDN. The operators still must agree on some distinguished CDN-domain that will be used for inter-CDN acquisition of CSP's content by dCDN. In this example, we'll use op-b-acq.op-a.net.

The operators must also exchange information regarding which requests dCDN is prepared to serve. For example, dCDN may be prepared to serve requests from clients in a given geographical region or a set of IP address prefixes. This information may again be provided out of band or via a defined protocol.

DNS must be configured in the following way:

- The content provider must be configured to make operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).

- Operator A must be configured so that a DNS request for op-b-acq.op-a.net returns a request router in Operator A.

- Operator B must be configured so that a request for node1.op-b.net/cdn.csp.com returns the IP address of a delivery node. Note that there might be a number of such delivery nodes.

Figure 3 illustrates how a client request for

http://cdn.csp.com/...rest of url...

is handled.

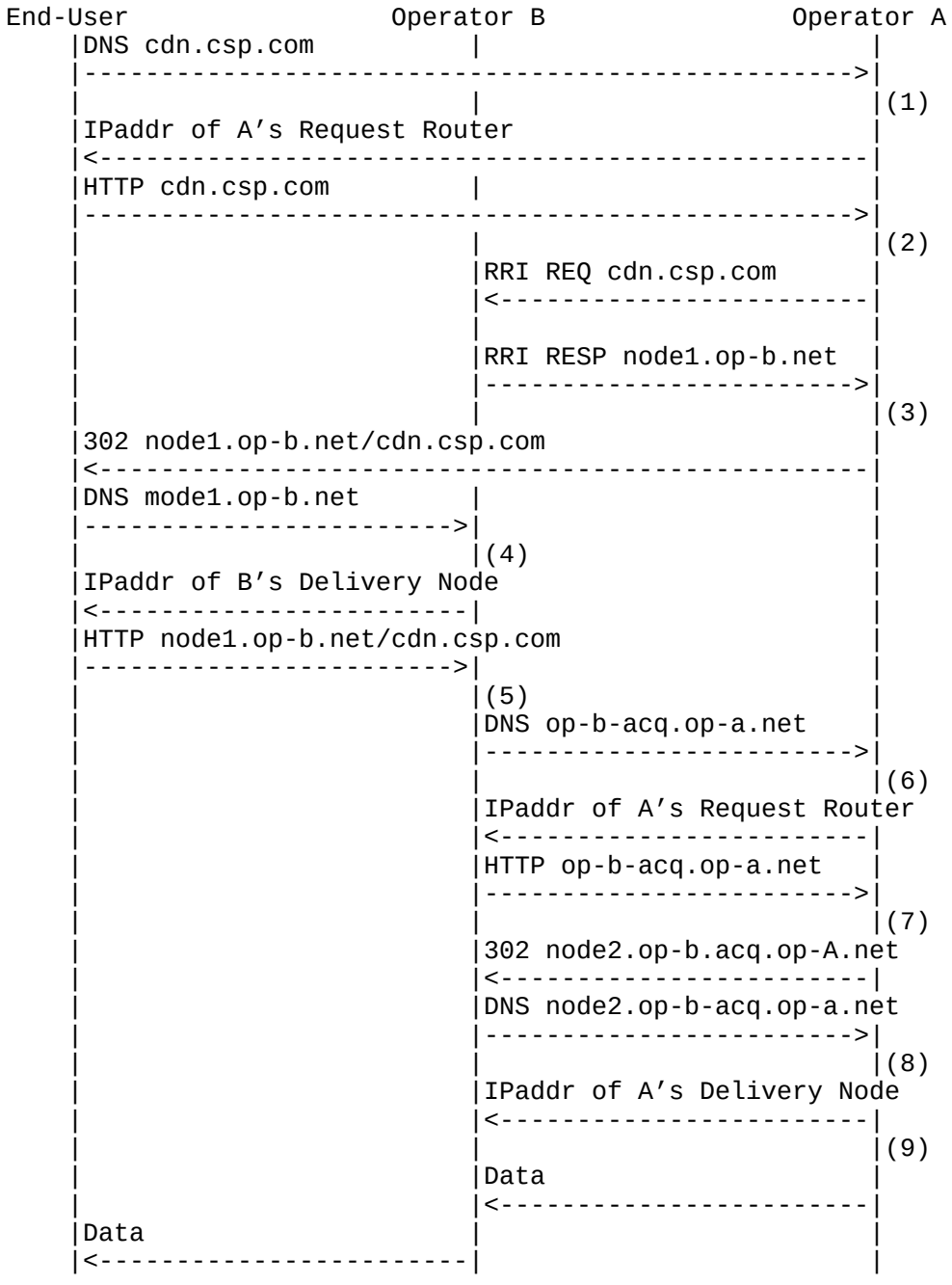


Figure 4: Request Trace for Recursive HTTP redirection method

The steps illustrated in the figure are as follows:

1. A DNS resolver for Operator A processes the DNS request for its customer based on CDN-domain cdn.csp.com. It returns the IP address of a Request Router in Operator A.

2. A Request Router for Operator A processes the HTTP request and recognizes that the end-user is best served by another CDN--specifically one provided by Operator B--and so it queries the CDNI Request Routing interface of Operator B, providing a set of information about the request including the URL requested. Operator B replies with the DNS name of a delivery node.

3. Operator A returns a 302 redirect message for a new URL obtained from the Request Routing Interface.

4. The end-user does a DNS lookup using the host name of the URL just provided (node1.op-b.net). B's DNS resolver returns the IP address of the corresponding delivery node. Note that, since the name of the delivery node was already obtained from B using the CDNI Request Routing Interface, there should not be any further redirection here (in contrast to the iterative method described above.)

5. The end-user requests the content from B's delivery node, potentially resulting in a cache miss. In the case of a cache miss, the content needs to be acquired from uCDN (not the CSP.) The distinguished CDN-domain op-b.net indicates to dCDN that this content is to be acquired from another CDN; stripping the CDN-domain reveals the original CDN-domain cdn.csp.com, dCDN may verify that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for the inter-CDN Acquisition "distinguished" CDN-domain as agreed above (in this case, op-b-acq.op-a.net).

6. Operator A DNS resolver processes the DNS request and returns the IP address of a request router in operator A.

7. The request router for Operator A processes the HTTP request from Operator B delivery node. Operator A request router recognizes that the request is from a peer CDN rather than an end-user because of the dedicated inter-CDN acquisition domain (op-b-acq.op-a.net). (Note that without this specially defined inter-CDN acquisition domain, operator A would be at risk of redirecting the request back to operator B, resulting in an infinite loop). The request router for Operator A selects a suitable delivery node in uCDN to serve the inter-CDN acquisition request and returns a 302 redirect message for a new URL

constructed by replacing the hostname by a subdomain of the Operator A's distinguished inter-CDN acquisition domain that points to the selected delivery node.

8. Operator A recognizes that the DNS request is from a peer CDN rather than an end-user (due to the internal CDN-domain) and so returns the address of a delivery node. (Note that without this specially defined internal domain, Operator A would be at risk of redirecting the request back to Operator B, resulting in an infinite loop.)

9. Operator A serves content for the requested CDN-domain to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has been registered, and determines the content provider that owns the origin server. It may also perform its own content acquisition steps if needed before returning the content to dCDN.

3.3.1. Comments on the example

Recursive redirection has the advantage over iterative of being more transparent from the end-user's perspective, but the disadvantage of each CDN exposing more of its internal structure (in particular, the addresses of edge caches) to peer CDNs. By contrast, iterative redirection does not require dCDN to expose the addresses of its edge caches to uCDN.

This example happens to use HTTP-based redirection in both CDN A and CDN B, but a similar example could be constructed using DNS-based redirection in either CDN. Hence, the key point to take away here is simply that the end user only sees a single redirection of some type, as opposed to the pair of redirections in the prior (iterative) example.

The use of the Request Routing Interface requires that interface to be appropriately configured and bootstrapped, which is not shown here. More discussion on the bootstrapping of interfaces is provided in Section 4

3.4. DNS-based redirection example

In this section we walk through a simple example using DNS-based redirection for request redirection from uCDN to dCDN (as well as for request routing inside dCDN and uCDN). As noted in Section 2.1, DNS-based redirection has certain advantages over HTTP-based redirection (notably, it is transparent to the end-user) as well as some drawbacks (notably the client IP address is not visible to the

request router).

As before, Operator A must learn the set of requests that dCDN is willing or able to serve (e.g. which client IP address prefixes or geographic regions are part of the dCDN footprint). Operator B must have and make known to operator A some unique identifier that can be used for the construction of a distinguished CDN domain, as shown in more detail below. (This identifier strictly needs only to be unique within the scope of Operator A, but a globally unique identifier, such as an AS number assigned to B, is one easy way to achieve that.) Also, Operator A must obtain the NS records for Operator B's externally visible redirection servers. Also, as before, a distinguished CDN-domain, such as op-b-acq.op-a.net, must be assigned for inter-CDN acquisition.

DNS must be configured in the following way:

- The CSP must be configured to make Operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).

- When uCDN sees a request best served by dCDN, it returns CNAME and NS records for "b.cdn.csp.com", where "b" is the unique identifier assigned to Operator B. (It may, for example, be an AS number assigned to Operator B.)

- dCDN must be configured so that a request for "b.cdn.csp.com" returns a delivery node in dCDN.

- uCDN must be configured so that a request for "op-b-acq.op-a.net" returns a delivery node in uCDN.

Figure 5 depicts the exchange of DNS and HTTP requests. The main differences from Figure 3 are the lack of HTTP redirection and transparency to the end-user.

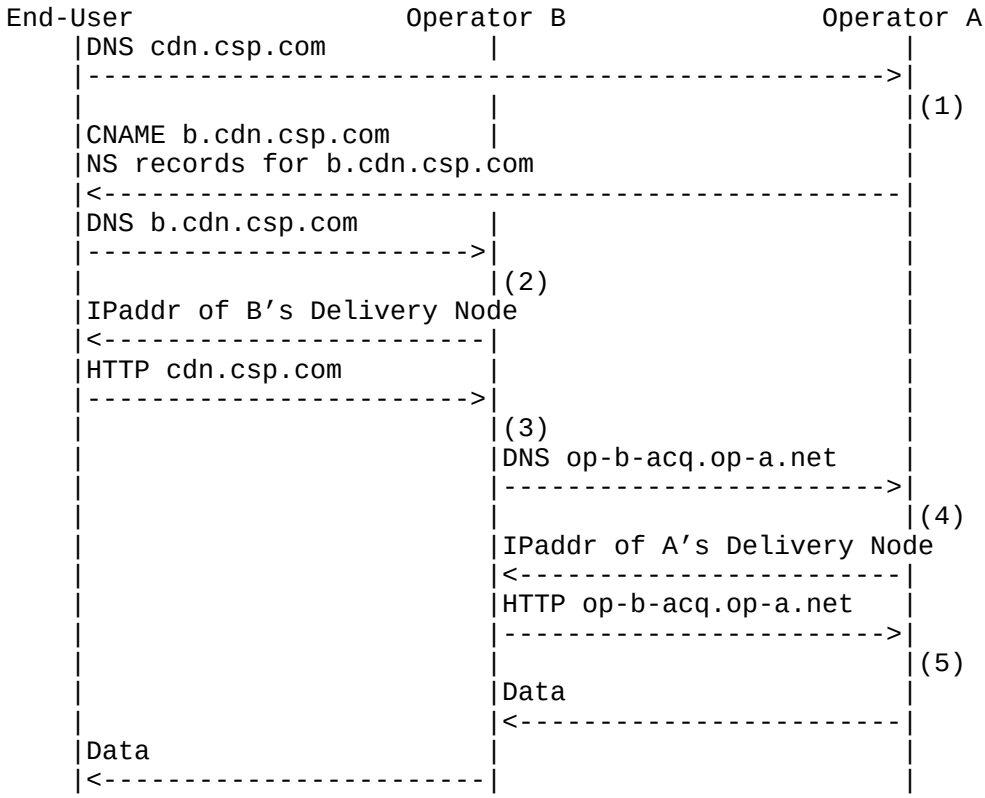


Figure 5: Request Trace for DNS-based Redirection Example

The steps illustrated in the figure are as follows:

1. Request Router for Operator A processes the DNS request for CDN-domain cdn.csp.com and recognizes that the end-user is best served by another CDN. (This may depend on the IP address of the user's local DNS resolver, or other information discussed below.) The Request Router returns a DNS CNAME response by "stacking" the distinguished identifier for Operator B onto the original CDN-domain (e.g., b.cdn.csp.com), plus an NS record that maps b.cdn.csp.com to B's Request Router.

2. The end-user does a DNS lookup using the modified CDN-domain (i.e., b.cdn.csp.com). This causes B's Request Router to respond with a suitable delivery node.

3. The end-user requests the content from B's delivery node. The requested URL contains the name cdn.csp.com. (Note that the returned CNAME does not affect the URL.) At this point the

delivery node has the correct IP address of the end-user and can do an HTTP 302 redirect if the redirections in steps 2 and 3 were incorrect. Otherwise B verifies that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for an "internal" CDN-domain as agreed above (op-b-acq.op-a.net).

4. Operator A recognizes that the DNS request is from a peer CDN rather than an end-user (due to the internal CDN-domain) and so returns the address of a delivery node in uCDN.

5. Operator A serves content to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has been registered, and determines the content provider that owns the origin server.

3.4.1. Comments on the example

The advantages of this approach are that it is more transparent to the end-user and requires fewer round trips than HTTP-based redirection. A potential problem is that the upstream CDN depends on being able to learn the correct downstream CDN that serves the end-user from the client address in the DNS request. In standard DNS operation, uCDN will only obtain the address of the client's local DNS resolver (LDNS), which is not guaranteed to be in the same network (or geographic region) as the client. If not--e.g., the end-user uses a global DNS service--then the upstream CDN cannot determine the appropriate downstream CDN to serve the end-user. In this case, one option is for the upstream CDN to treat the end-user as it would any user not connected to a peer CDN. Another option is for the upstream CDN to "fall back" to a pure HTTP-based redirection strategy in this case (i.e., use the first method). Note that this problem affects existing CDNs that rely on DNS to determine where to redirect client requests, but the consequences are arguably less serious since the LDNS is likely in the same network as the dCDN serves. One approach to ensuring that the client's IP address prefix is correctly determined in such situations is described in [I-D.vandergaast-edns-client-subnet].

As with the prior example, this example partially illustrates the various interfaces involved in CDNI. Operator A could learn dynamically from Operator B the set of prefixes or regions that B is willing and able to serve via the request routing interface. The distinguished name used for acquisition and the identifier for Operator B that is prepended to the CDN domain on redirection are examples of information elements that might also be conveyed by CDNI interfaces (or, alternatively, statically configured). As before,

minimal metadata sufficient to obtain the content is carried "in-band" as part of the redirection process, and standard HTTP is used for inter-CDN acquisition. There is no explicit logging interface discussed in this example.

3.5. Dynamic Footprint Discovery

There could be situations where being able to dynamically discover the set of requests that a given dCDN is willing and able to serve is beneficial. For example, a CDN might at one time be able to serve a certain set of client IP prefixes, but that set might change over time due to changes in the topology and routing policies of the IP network. The following example illustrates this capability. We have chosen the example of DNS-based redirection, but HTTP-based redirection could equally well use this approach.

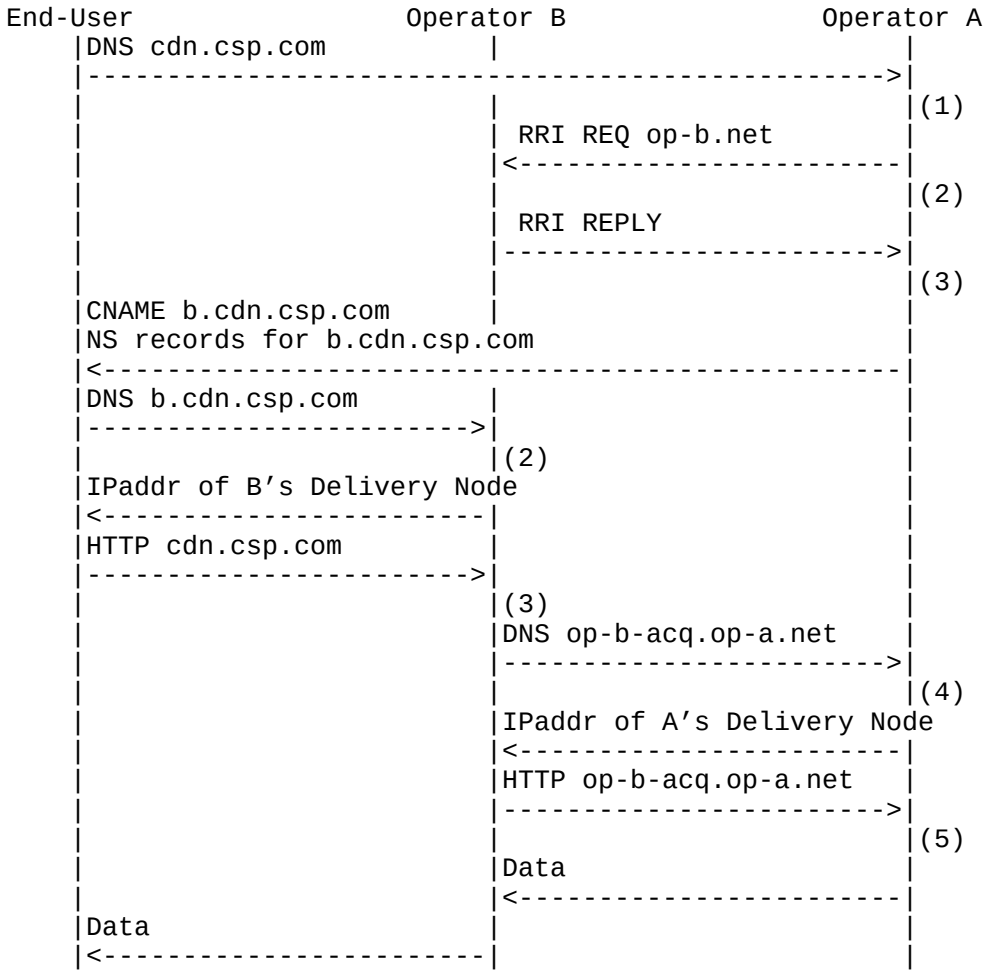


Figure 6: Request Trace for Dynamic Footprint Discovery Example

This example differs from the one in Figure 5 only in the addition of a CDNI Request Routing Interface request (step 2) and corresponding response (step 3). The RRI Req could be a message such as "Can you serve clients from this IP Prefix?" or it could be "Provide the list of client IP prefixes you can currently serve". In either case the response might be cached by operator A to avoid repeatedly asking the same question. Alternatively, or in addition, Operator B may spontaneously advertise to Operator A information (or changes) on the set of requests it is willing and able to serve on behalf of operator A; in that case, Operator B may spontaneously issue RRI REPLY messages that are not in direct response to a corresponding RRI REQ message. (Note that the issues of determining the client’s subnet

from DNS requests, as described above, are exactly the same here as in Section 3.4.)

Once Operator A obtains the RRI response, it is now able to determine that Operator B's CDN is an appropriate dCDN for this request and therefore a valid candidate dCDN to consider in its Redirection decision. If that dCDN is selected, the redirection and serving of the request proceeds as before (i.e. in the absence of dynamic footprint discovery).

3.6. Content Removal

The following example illustrates how the Metadata interface may be used to remove an item of content. In this example, user requests for a particular content, and corresponding redirection of such requests from Operator A to Operator B CDN, may (or may not) have taken place earlier. Then, at some point in time, the uCDN (for example, in response to a corresponding trigger from the Content Provider) uses the Metadata Interface to request that content identified by a particular URL be removed from dCDN. The following diagram illustrates the operation.

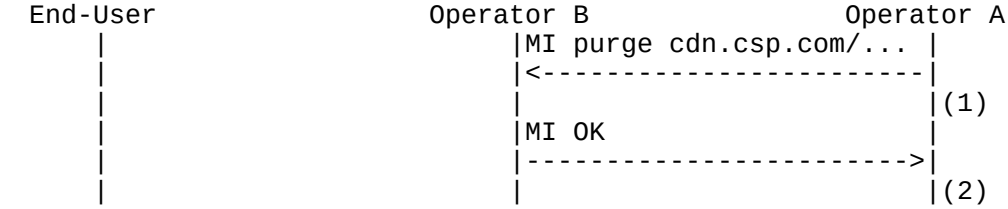


Figure 7: Request Trace for Content Removal

The metadata interface is used to convey the request from uCDN to dCDN that some previously acquired content should be deleted. The URL in the request specifies which content to remove. This example corresponds to a DNS-based redirection scenario such as Section 3.4. If HTTP-based redirection had been used, the URL for removal would be of the form peer-a.op-b.net/cdn.csp.com/...

The dCDN is expected to confirm to the uCDN, as illustrated by the MI OK message, the completion of the removal of the targeted content from all the caches in dCDN.

3.7. Pre-Positioned Content Acquisition Example

The following example illustrates how the metadata interface may be used to pre-position an item of content in the dCDN. In this example, Operator A uses the Metadata Interface to request that

content identified by a particular URL be pre-positioned into Operator B CDN.

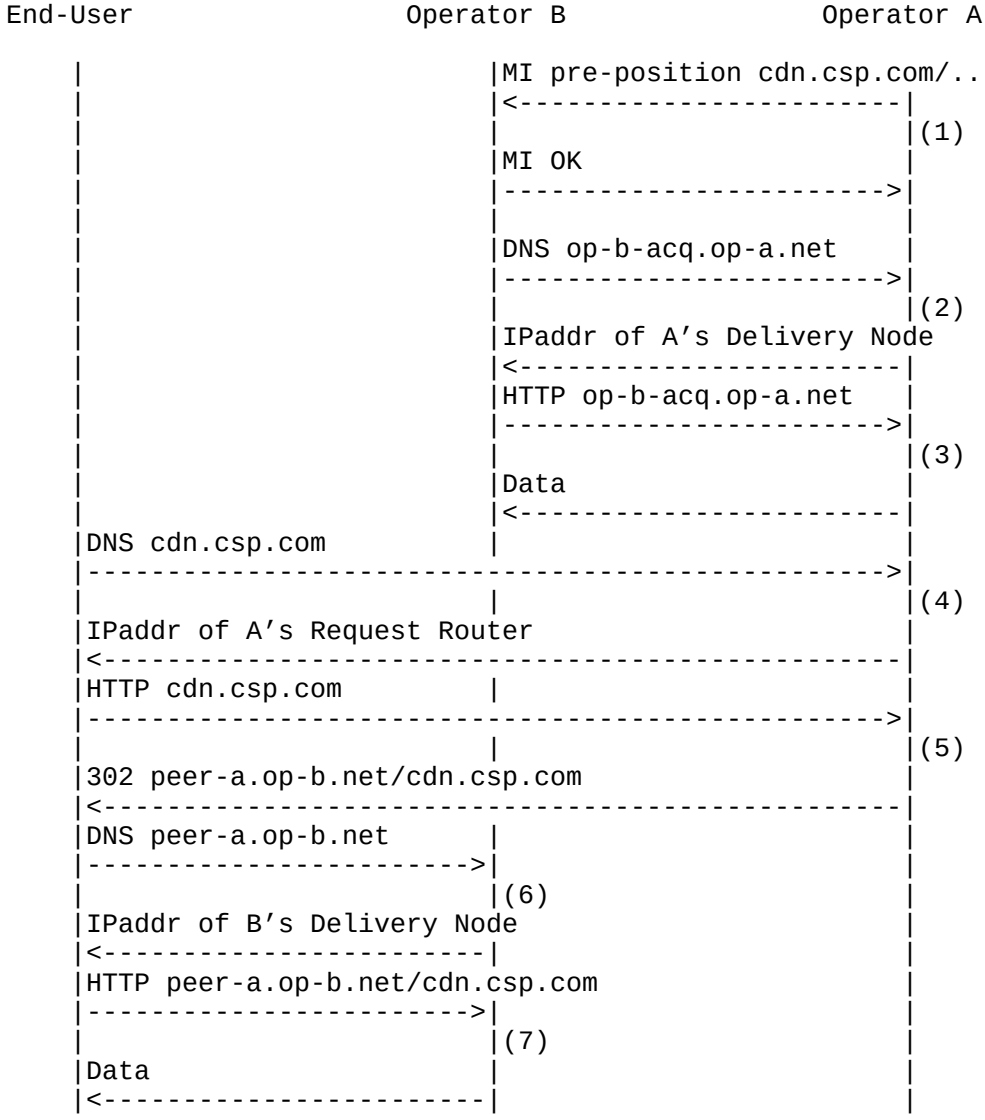


Figure 8: Request Trace for Content Pre-Positioning

The steps illustrated in the figure are as follows:

1. Operator A uses the Metadata Interface to request that Operator B pre-positions a particular content item identified by its URL.

Operator B responds by confirming that it is willing to perform this operation.

Steps 2 and 3 are exactly the same as steps 5 and 6 of Figure 3, only this time those steps happen as the result of the Pre-positioning request instead of as the result of a cache miss.

Steps 4, 5, 6, 7 are exactly the same as steps 1, 2, 3, 4 of Figure 3, only this time Operator B CDN can serve the end-user request without triggering dynamic content acquisition, since the content has been pre-positioned in dCDN. Note that, depending on dCDN operations and policies, the content pre-positioned in the dCDN may be pre-positioned to all, or a subset of, dCDN caches. In the latter case, intra-CDN dynamic content acquisition may take place inside the dCDN serving requests from caches on which the content has not been pre-positioning; however, such intra-CDN dynamic acquisition would not involve the uCDN.

3.8. Asynchronous CDNI Metadata Example

In this section we walk through a simple example illustrating a scenario of asynchronously exchanging CDNI metadata, where the downstream CDN obtains CDNI metadata for content ahead of a corresponding content request. The example that follows assumes that HTTP-based inter-CDN redirection and recursive CDNI request-routing are used, as in Section 3.3. However, asynchronous exchange of CDNI Metadata is similarly applicable to DNS-based inter-CDN redirection and iterative request routing (in which cases the CDNI metadata may be used at slightly different processing stages of the message flows).

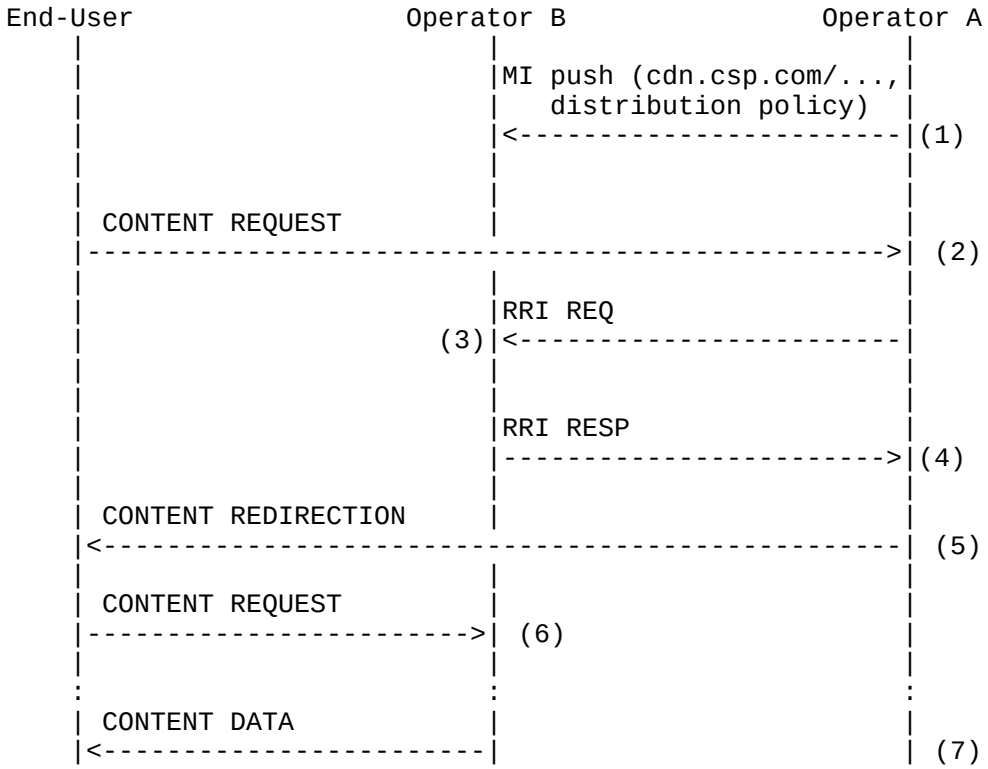


Figure 9: Request Trace for Asynchronous CDNI Metadata

The steps illustrated in the figure are as follows:

1. Operator A uses the Metadata Interface to asynchronously push CDNI metadata to Operator B. The present document does not constrain how the CDNI metadata information is actually represented. For the purposes of this example, we assume that Operator A provides CDNI metadata to Operator B indicating that:

   * this CDNI Metadata is applicable to any content referenced by "cdn.csp.com/op-b.net/..." (assuming HTTP redirection is used - it would be applicable to "cdn.csp.com/..." if DNS redirection were used as in Section 3.4).

   * this CDNI metadata consists of a distribution policy requiring enforcement by the delivery node of a specific per-request authorization mechanism (e.g. URI signature or token validation).

2. A Content Request occurs as usual.

3. A CDNI Request Routing Request (RRI REQ) is issued by operator A CDN, as discussed in Section 3.3. Operator B's request router can access the CDNI Metadata that are relevant to the requested content and that have been pre-positioned as per Step 1, which may or may not affect the response.

4. Operator B's request router issues a CDNI Request Routing Response (RRI RESP) as in Section 3.3.

5. Operator B performs content redirection as discussed in Section 3.3.

6. On receipt of the Content Request by the end user, the delivery node detects that previously acquired CDNI metadata is applicable to the requested content. In accordance with the specific CDNI metadata of this example, the delivery node will invoke the appropriate per-request authorization mechanism, before serving the content. (Details of this authorization are not shown.)

7. Assuming successful per-request authorization, serving of Content Data (possibly preceded by inter-CDN acquisition) proceeds as in Section 3.3.

3.9. Synchronous CDNI Metadata Acquisition Example

In this section we walk through a simple example illustrating a scenario of synchronous CDNI metadata acquisition, in which the downstream CDN obtains CDNI metadata for content at the time of handling a first request for the corresponding content. As in the preceding section, this example assumes that HTTP-based inter-CDN redirection and recursive CDNI request-routing are used (as in Section 3.3), but dynamic CDNI metadata acquisition is applicable to other variations of request routing.

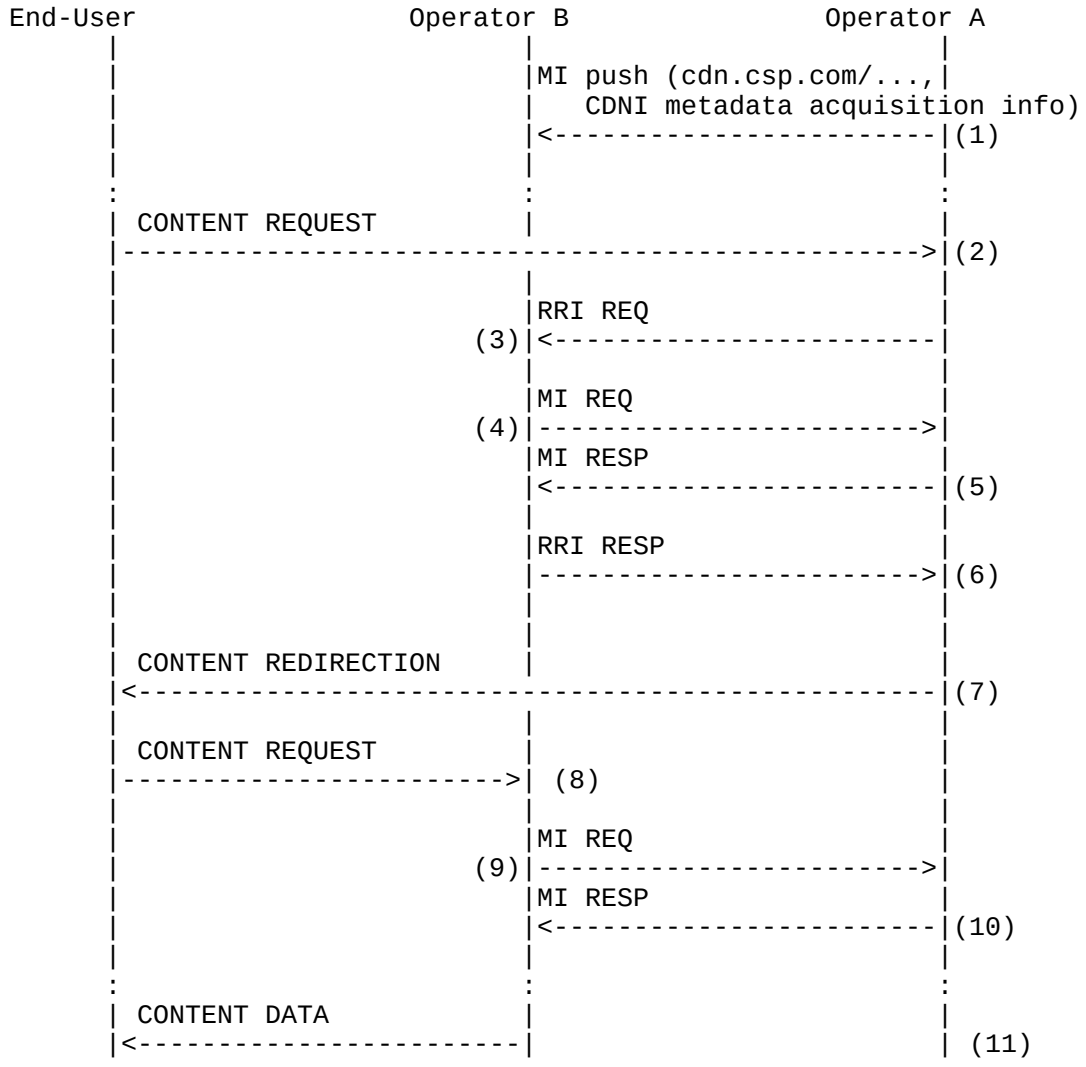


Figure 10: Request Trace for Synchronous CDNI Metadata Acquisition

The steps illustrated in the figure are as follows:

1. Operator A initially uses the Metadata Interface to asynchronously push seed metadata to Operator B. For example, this seed information may include a URI indicating where CDNI Metadata can later be pulled from for some content set. (There are alternative ways that this seeding information may be provided, such as piggybacking on the CDNI RRI REQ message of

Step 3.)

2. A Content Request arrives as normal.

3. A Request Routing Interface request occurs as in the prior example.

4. On receipt of the CDNI Request Routing Request, Operator B's CDN initiates synchronous acquisition of CDNI Metadata that are needed for routing of the end-user request. The seeding information provided in Step 1 is used to determine how to obtain the metadata. Note that there may exist cases in which this step does not occur (e.g., because the CDNI metadata seeding information indicates CDNI metadata are not needed at that stage).

5. On receipt of a CDNI Metadata MI Request, Operator A's CDN responds, making the corresponding CDNI metadata information available to Operator B's CDN. This metadata is considered by operator B's CDN before responding to the Request Routing request. (In a simple case, the metadata could simply be an allow or deny response for this particular request.)

6. Response to the RRI request as normal.

7. Redirection message is sent to the end user.

8. A delivery node of Operator B receives the end user request.

9. The delivery node triggers dynamic acquisition of additional CDNI metadata that are needed to process the end-user content request. Again the seeding information provided in Step 1 is used to determine how to acquire the needed CDNI metadata. Note that there may exist cases where this step need not happen, either because the metadata were already acquired previously, or because the seeding information indicates no metadata are required.

10. Operator A's CDN responds to the CDNI Metadata Request and makes the corresponding CDNI metadata available to Operator B. This metadata influence how Operator B's CDN processes the end-user request.

11. Content is served (possibly preceded by inter-CDN acquisition) as in Section 3.3.

4. Main Interfaces

Figure 1 illustrates the four main interfaces that are in scope for the CDNI WG, along with several others. The detailed specifications of these interfaces are left to other documents (mostly still to be written, but see [I-D.ietf-cdni-problem-statement] and [I-D.ietf-cdni-requirements] for some discussion of the interfaces).

One interface that is not shown in Figure 1 is the interface between the user and the CSP. While for the purposes of CDNI that interface is out of scope, it is worth noting that it does exist and can provide useful functions, such as end-to-end performance monitoring and some forms of authentication and authorization.

There is also an important interface between the user and the Request Routing function of both uCDN and dCDN. As we saw in some of the preceding examples, that interface can be used as a way of passing information such as the metadata that is required to obtain the content in dCDN from uCDN.

In this section we will provide an overview of the functions performed by each of the CDNI interfaces and discuss how they fit into the overall solution. We also examine some of the design tradeoffs. We begin with an examination of one such tradeoff that affects all the interfaces - the use of in-band or out-of-band communication.

4.1. In-Band versus Out-of-Band Interfaces

Before getting to the individual interfaces, we observe that there is a high-level design choice for each, involving the use of existing in-band communication channels versus defining new out-of-band interfaces.

It is possible that the information needed to carry out various interconnection functions can be communicated between peer CDNs using existing in-band protocols. The use of HTTP 302 redirect is an example of how certain aspects of request routing can be implemented in-band (embedded in URIs). Note that using existing in-band protocols does not imply that the CDNI interfaces are null; it is still necessary to establish the rules (conventions) by which such protocols are used to implement the various interface functions.

There are other opportunities for in-band communication beyond HTTP redirects. For example, many of the HTTP directives used by proxy servers can also be used by peer CDNs to inform each other of caching activity. Of these, one that is particularly relevant is the If-Modified-Since directive, which is used with the GET method to make

it conditional: if the requested object has not been modified since the time specified in this field, a copy of the object will not be returned, and instead, a 304 (not modified) response will be returned.

4.2. Request Routing Interface

We may think of the request routing interface as comprising two parts: the asynchronous advertisement of footprint and capabilities by a dCDN that allows a uCDN to decide whether to redirect particular user requests to that dCDN; and the synchronous operation of actually redirecting a user request. (These are somewhat analogous to the operations of routing and forwarding in IP.)

As illustrated in Section 3, the synchronous part of the request routing interface may be implemented in part by DNS and HTTP. Naming conventions may be established by which CDN peers communicate whether a request should be routed or content served.

In support of these exchanges, it is necessary for CDN peers to exchange additional information with each other. Depending on the method(s) supported, this includes

- The operator’s unique id (operator-id) or distinguished CDN-domain (operator-domain);
- NS records for the operator’s set of externally visible request routers;
- The set of requests the dCDN operator is prepared to serve (e.g. a set of client IP prefixes or geographic regions that may be served by dCDN).

Of these, the two operator identifiers are fixed, and can be exchanged off-line as part of a peering agreement. The NS records potentially change with some frequency, but an existing protocol--DNS--can be used to dynamically track this information. That is, a peer can do a DNS lookup on operator-domain to retrieve the set of NS records corresponding to the peer’s redirection service.

The set of requests that dCDN is willing to serve could in some cases be relatively static (e.g., a set of IP prefixes) which could be exchanged off-line, or might even be negotiated as part of a peering agreement. However, it may also be more dynamic, in which case an explicit protocol for its exchange would be be helpful.

A variety of options exist for the dCDN operator to advertise its footprint to uCDN. As discussed in

[I-D.previdi-cdni-footprint-advertisement], footprint is comprised of two components:

- a class of end user requests (represented, for example, by a set of IP prefixes, or a geographic region) that the dCDN is willing and able to serve directly, without use of another dCDN;

- the connectivity of the dCDN to other CDNs that may be able to serve content to users on behalf of dCDN.

[I-D.previdi-cdni-footprint-advertisement] describes an approach to advertising such footprint information asynchronously using BGP. In addition to this sort of information, a dCDN might also advertise "capabilities" such as the ability to handle certain types of content (e.g. specific streaming formats) or quality of service (QoS) capabilities. [I-D.xiaoyan-cdni-request-routing-protocol] describes an approach that exchanges CDN "capabilities" over HTTP, while [I-D.seedorf-alto-for-cdni] describes how ALTO [RFC5693] may be used to obtain request routing information.

We also note that the Request Routing interface plays a key role in enabling recursive redirection, as illustrated in Section 3.3. It enables the user to be redirected to the correct delivery node in dCDN with only a single redirection step (as seen by the user). This may be particularly valuable as the chain of interconnected CDNs increases beyond two CDNs.

4.3. Logging Interface

It is necessary for the upstream CDN to have visibility into the delivery of content it originates to end-users connected to the downstream CDN. This allows the upstream CDN to properly bill its customers for multiple deliveries of content cached by the downstream CDN, as well as to report accurate traffic statistics to those content providers. This is one role of the Logging interface.

Other operational data that may be relevant to CDNI can also be exchanged by the Logging interface. For example, dCDN may report the amount of content it has acquired from uCDN, and how much cache storage has been consumed by content cached on behalf of uCDN.

Traffic logs are easily exchanged off-line. For example, the following traffic log is a small deviation from the Apache log file format, where entries include the following fields:

- Domain - the full domain name of the origin server

- IP address - the IP address of the client making the request
- End time - the ending time of the transfer
- Time zone - any time zone modifier for the end time
- Method - the transfer command itself (e.g., GET, POST, HEAD)
- URL - the requested URL
- Version - the protocol version, such as HTTP/1.0
- Response - a numeric response code indicating transfer result
- Bytes Sent - the number of bytes in the body sent to the client
- Request ID - a unique identifier for this transfer
- User agent - the user agent, if supplied
- Duration - the duration of the transfer in milliseconds
- Cached Bytes - the number of body bytes served from the cache
- Referrer - the referrer string from the client, if supplied

Of these, only the Domain field is indirect in the downstream CDN--it is set to the CDN-domain used by the upstream CDN rather than the actual origin server. This field could then used to filter traffic log entries so only those entries matching the upstream CDN are reported to the corresponding operator.

One open question is who does the filtering. One option is that the downstream CDN filters its own logs, and passes the relevant records directly to each upstream peer. This requires that the downstream CDN knows the set of CDN-domains that belong to each upstream peer. If this information is already exchanged between peers as part of the request routing interface, then direct peer-to-peer reporting is straightforward. If it is not available, and operators do not wish to advertise the set of CDN-domains they serve to their peers, then the second option is for each CDN to send both its non-local traffic records and the set of CDN-domains it serves to an independent third-party (i.e., a CDN Exchange), which subsequently filters, merges, and distributes traffic records on behalf of each participating CDN operator.

A second open question is how timely traffic information should be. For example, in addition to off-line traffic logs, accurate real-time

traffic monitoring might also be useful, but such information requires that the downstream CDN inform the upstream CDN each time it serves upstream content from its cache. The downstream CDN can do this, for example, by sending a conditional HTTP GET request (If-Modified-Since) to the upstream CDN each time it receives an HTTP GET request from one of its end-users. This allows the upstream CDN to record that a request has been issued for the purpose of real-time traffic monitoring. The upstream CDN can also use this information to validate the traffic logs received later from the downstream CDN.

There is obviously a tradeoff between accuracy of such monitoring and the overhead of the downstream CDN having to go back to the upstream CDN for every request.

Another design tradeoff in the Logging interface is the degree of aggregation or summarization of data. One situation that lends itself to summarization is the delivery of HTTP-based adaptive bit-rate video. Most schemes to deliver such video use a large number of relatively small HTTP requests (e.g. one request per 2-second chunk of video.) It may be desirable to aggregate logging information so that a single log entry is provided for the entire video rather than for each chunk. Note however that such aggregation requires a degree of application awareness in dCDN to recognize that the many HTTP requests correspond to a single video.

Other forms of aggregation may also be useful. For example, there may be situations where bulk metrics such as bytes delivered per hour may suffice rather than the detailed per-request logs outlined above. It seems likely that a range of granularities of logging will be needed along with ways to specify the type and degree of aggregation required.

4.4. Control Interface

The control interface is primarily used for the bootstrapping of other interfaces. As a simple example, it could be used to provide the address of the logging server in dCDN to uCDN in order to bootstrap the logging interface. It may also be used, for example, to establish security associations for the other interfaces. We discuss the relationship between the Control and Metadata interfaces in the next section.

4.5. Metadata Interface

The role of the metadata interface is to enable CDNI distribution metadata to be conveyed to the downstream CDN by the upstream CDN. Such metadata includes geo-blocking restrictions, availability windows, access control policies, and so on. It may also include

policy information such as the desire to pre-position content rather than fetch it on demand.

Some metadata may be able to be conveyed using in-band mechanisms. For example, to inform the downstream CDN of any geo-blocking restrictions or availability windows, the upstream can elect to redirect a request to the downstream CDN only if that CDN's advertised delivery footprint is acceptable for the requested URL. Similarly, the request could be forwarded only if the current time is within the availability window.

Similarly, some forms of access control may also be performed on a per-request basis using HTTP directives. For example, being able to respond to a conditional GET request gives the upstream CDN an opportunity to influence how the downstream CDN delivers its content. Minimally, the upstream CDN can invalidate (purge) content previously cached by the downstream CDN.

Fine-grain control over how the downstream CDN delivers content on behalf of the upstream CDN is also possible. For example, by including the X-Forwarded-For HTTP header with the conditional GET request, the downstream CDN can report the end-user's IP address to the upstream CDN, giving it an opportunity to control whether the downstream CDN should serve the content to this particular end-user. The upstream CDN would communicate its directive through its response to the conditional GET. The downstream CDN can cache information for a period of time specified by the upstream CDN, thereby reducing control overhead.

Thinking beyond what metadata operations can be done in-line, we note that all CDNs already export a "content purge" operation to their customers. The CDNI metadata interface could support a similar "content purge" API call. When a CSP invokes purge on the upstream CDN, that CDN in turn invokes purge on all downstream CDNs that might be caching the content. Of course, agreement as to the syntax and semantics of this call is required.

One open question is how to distinguish between what functionality is supported by the Metadata interface and what functionality is supported by the Control interface. The approach taken in this document is to assume a minimal Control interface that is used to bootstrap the other interfaces. We assume all information that governs peer CDN behavior at the granularity of individual content items is exchanged via the Metadata interface. We note that some other documents have suggested that the purge operation should be part of the Control Interface. The authors' view is that purging a piece of content is just another form of metadata, similar to an availability window. In effect, a purge is equivalent to a statement

that the availability window for that content has now expired. The timeliness requirements for purge operations may affect the detailed design of the metadata interface.

5. Deployment Models

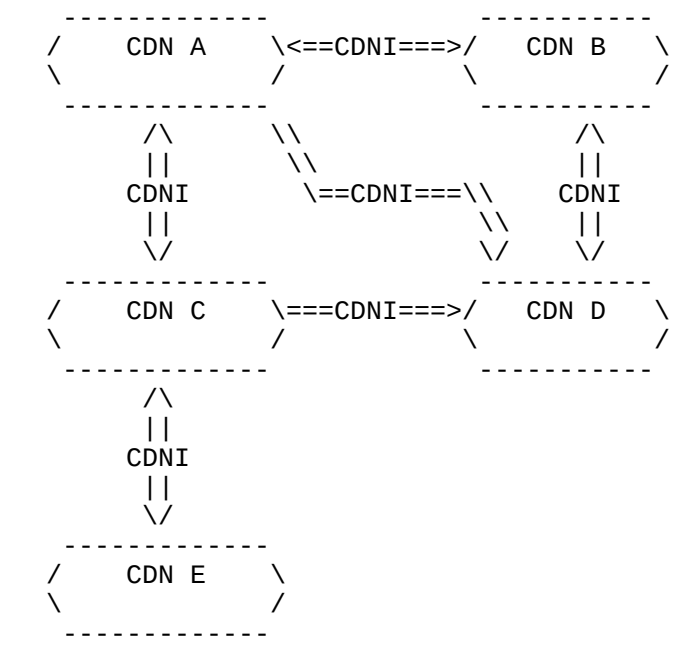
In this section we describe a number of possible deployment models that may be achieved using the CDNI interfaces described above. We note that these models are by no means exhaustive, and that may other models may be possible.

Although the reference model of Figure 1 shows all CDN functions on each side of the CDNI interface, deployments can rely on entities that are involved in any subset of these functions, and therefore only support the relevant subset of CDNI interfaces. As already noted in Section 3, effective CDNI deployments can be built without necessarily implementing all four interfaces. Some examples of such deployments are shown below.

Note that, while we refer to upstream and downstream CDNs, this distinction applies to specific content items and transactions. That is, a given CDN may be upstream for some transactions and downstream for others, depending on many factors such as location of the requesting client and the particular piece of content requested.

5.1. Meshed CDNs

Although the reference model illustrated in Figure 1 shows a unidirectional CDN interconnection with a single uCDN and a single dCDN, any arbitrary CDNI meshing can be built from this, such as the example meshing illustrated in Figure 11. (Support for arbitrary meshing may or may not be in the initial scope for the working group, but the model allows for it.)



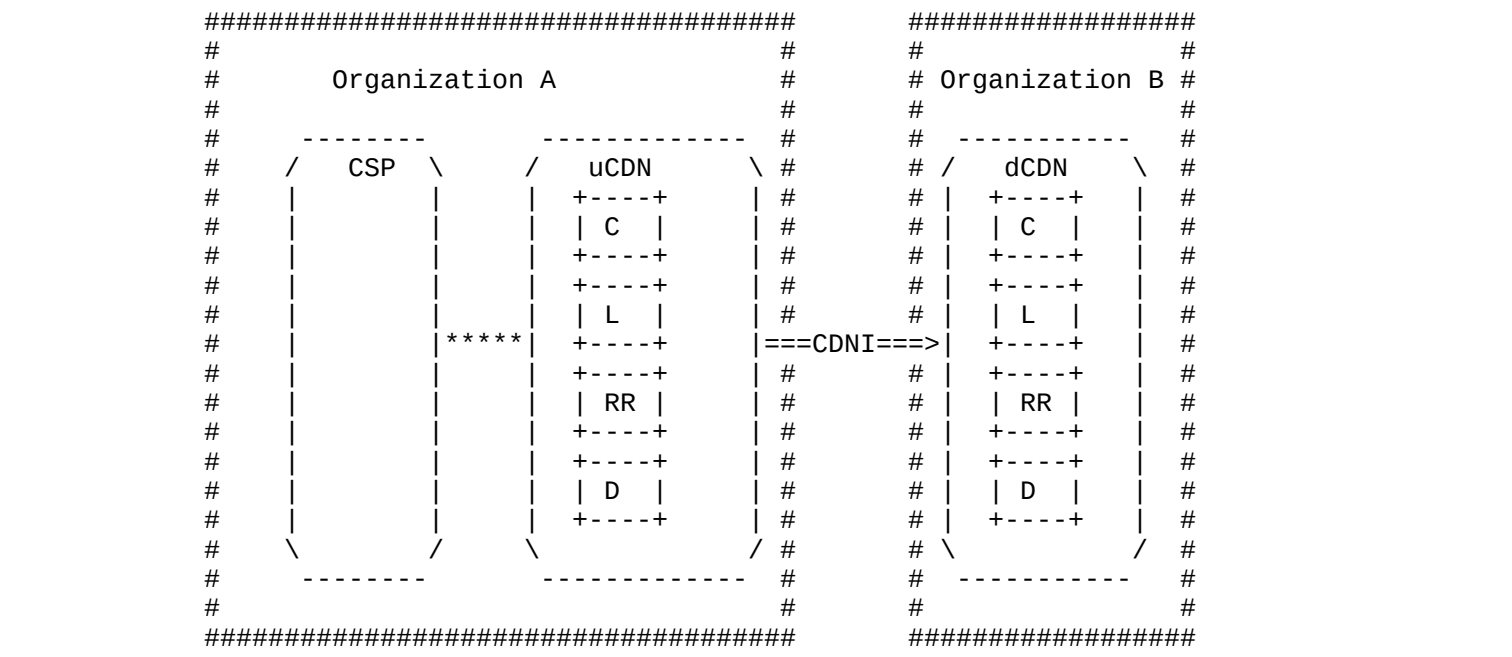
===> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN

<==> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN and with left-hand side CDN acting as dCDN to right-hand side CDN

Figure 11: CDNI Deployment Model: CDN Meshing Example

5.2. CSP combined with CDN

Note that our terminology refers to functional roles and not economic or business roles. That is, a given organization may be operating as both a CSP and a fully-fledged uCDN when we consider the functions performed, as illustrated in Figure 12.

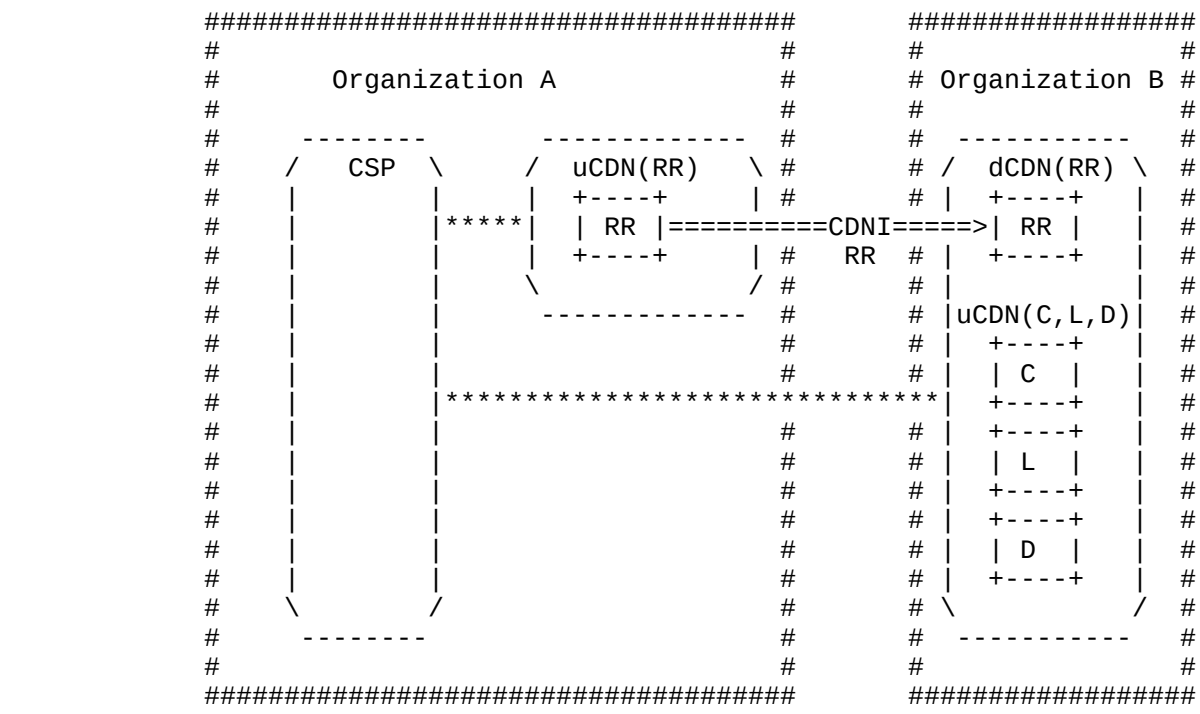


===> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN
**** interfaces outside the scope of CDNI
C Control component of the CDN
L Logging component of the CDN
RR Request Routing component of the CDN
D Distribution component of the CDN

Figure 12: CDNI Deployment Model: Organization combining CSP & uCDN

5.3. CSP using CDNI Request Routing Interface

As another example, a content provider organization may choose to run its own request routing function as a way to select among multiple candidate CDN providers; In this case the content provider may be modeled as the combination of a CSP and of a special, restricted case of a CDN. In that case, as illustrated in Figure 13, the CDNI Request Routing interface can be used between the restricted CDN operated by the content provider Organization and the CDN operated by the full-CDN organization acting as a dCDN in the request routing control plane. Interfaces outside the scope of the CDNI work can be used between the CSP functional entities of the content provider organization and the CDN operated by the full-CDN organization acting as a uCDN) in the CDNI control planes other than the request routing plane (i.e. Control, Distribution, Logging).



===> CDNI Request Routing interface
**** interfaces outside the scope of CDNI

Figure 13: CDNI Deployment Model: Organization combining CSP and partial CDN

5.4. CDN Federations and CDN Exchanges

There are two additional concepts related to, but distinct from CDN Interconnection. The first is CDN Federation. Our view is that CDNI is the more general concept, involving two or more CDNs serving content to each other's users, while federation implies a multi-lateral interconnection arrangement, but other CDN interconnection agreements are also possible (e.g., symmetric bilateral, asymmetric bilateral). An important conclusion is that CDNI technology should not presume (or bake in) a particular interconnection agreement, but should instead be general enough to permit alternative interconnection arrangements to evolve.

The second concept often used in the context of CDN Federation is CDN Exchange--a third party broker or exchange that is used to facilitate a CDN federation. Our view is that a CDN exchange offers valuable machinery to scale the number of CDN operators involved in a multi-

lateral (federated) agreement, but that this machinery is built on top of the core CDNI interconnection mechanisms. For example, as illustrated in Figure 14, the exchange might aggregate and redistribute information about each CDN footprint and capacity, as well as collect, filter, and re-distribute traffic logs that each participant needs for interconnection settlement, but inter-CDN request routing, inter-CDN content distribution (including inter-CDN acquisition) and inter-CDN control which fundamentally involve a direct interaction between an upstream CDN and a downstream CDN--operate exactly as in a pair-wise peering arrangement. Turning to Figure 14, we observe that in this example:

- each CDN supports a direct CDNI Control interface to every other CDN
- each CDN supports a direct CDNI Metadata interface to every other CDN
- each CDN supports a CDNI Logging interface with the CDN Exchange
- each CDN supports both a CDNI request Routing interface with the CDN Exchange (for aggregation and redistribution of dynamic CDN footprint discovery information) and a direct CDNI Request Routing interface to every other CDN (for actual request redirection).

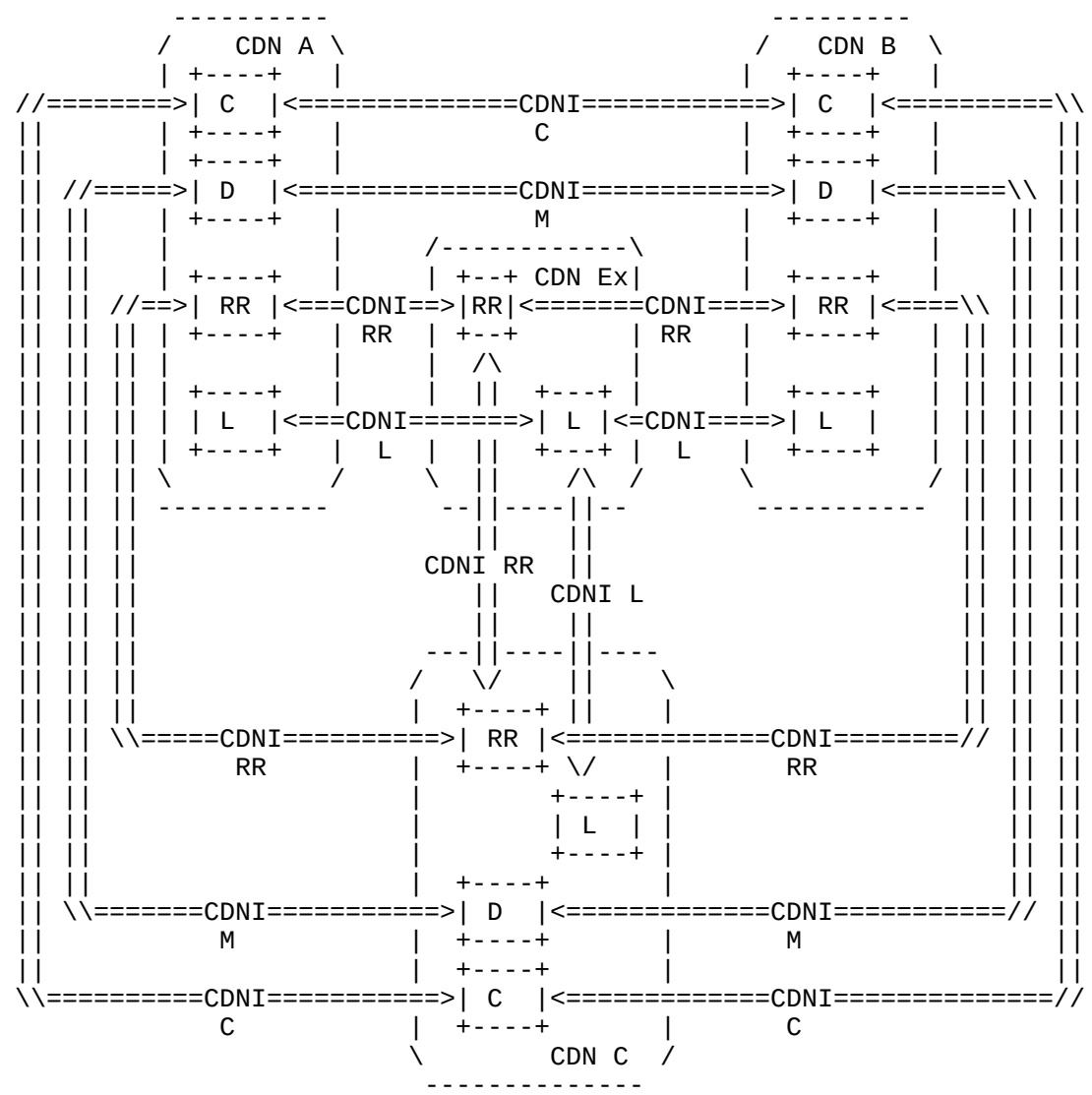


Figure 14: CDNI Deployment Model: CDN Exchange

Note that a CDN exchange may alternatively support a different set of functionality (e.g. Logging only, or Logging and full request

routing, or all the functionality of a CDN including content distribution). All these options are expected to be allowed by the IETF CDNI specifications.

6. Trust Model

There are a number of trust issues that need to be addressed by a CDNI solution. Many of them are in fact similar or identical to those in a simple CDN without interconnection. In a standard CDN environment (without CDNI), the CSP places a degree of trust in a single CDN operator to perform many functions. The CDN is trusted to deliver content with appropriate quality of experience for the end user. The CSP trusts the CDN operator not to corrupt or modify the content. The CSP often relies on the CDN operator to provide reliable accounting information regarding the volume of delivered content. The CSP may also trust the CDN operator to perform actions such as timely invalidation of content and restriction of access to content based on certain criteria such as location of the user and time of day, and to enforce per-request authorization performed by the CSP using techniques such as URI signing.

A CSP also places trust in the CDN not to distribute any information that is confidential to the CSP (e.g., how popular a given piece of content is) or confidential to the end user (e.g., which content has been watched by which user).

A CSP does not necessarily have to place complete trust in a CDN. A CSP will in some cases take steps to protect its content from improper distribution by a CDN, e.g. by encrypting it and distributing keys in some out of band way. A CSP also depends on monitoring (possibly by third parties) and reporting to verify that the CDN has performed adequately. A CSP may use techniques such as client-based metering to verify that accounting information provided by the CDN is reliable. HTTP conditional requests may be used to provide the CSP with some checks on CDN operation. In other words, while a CSP may trust a CDN to perform some functions in the short term, the CSP is able in most cases to verify whether these actions have been performed correctly and to take action (such as moving the content to a different CDN) if the CDN does not live up to expectations.

The main trust issue raised by CDNI is that it introduces transitive trust. A CDN that has a direct relationship with a CSP can now "outsource" the delivery of content to another (downstream) CDN. That CDN may in term outsource delivery to yet another downstream CDN, and so on.

The top level CDN in such a chain of delegation is responsible for ensuring that the requirements of the CSP are met. Failure to do so is presumably just as serious as in the traditional single CDN case. Hence, an upstream CDN is essentially trusting a downstream CDN to perform functions on its behalf in just the same way as a CSP trusts a single CDN. Monitoring and reporting can similarly be used to verify that the downstream CDN has performed appropriately. However, the introduction of multiple CDNs in the path between CSP and end user complicates the picture. For example, third party monitoring of CDN performance (or other aspects of operation, such as timely invalidation) might be able to identify the fact that a problem occurred somewhere in the chain but not point to the particular CDN at fault.

In summary, we assume that an upstream CDN will invest a certain amount of trust in a downstream CDN, but that it will verify that the downstream CDN is performing correctly, and take corrective action (including potentially breaking off its relationship with that CDN) if behavior is not correct. We do not expect that the trust relationship between a CSP and its "top level" CDN will differ significantly from that found today in single CDN situations. However, it does appear that more sophisticated tools and techniques for monitoring CDN performance and behavior will be required to enable the identification of the CDN at fault in a particular delivery chain.

We expect that the detailed designs for the specific interfaces for CDNI will need to take the transitive trust issues into account. For example, explicit confirmation that some action (such as content removal) has taken place in a downstream CDN may help to mitigate some issues of transitive trust.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

While there is a variety of security issues introduced by a single CDN, we are concerned here specifically with the additional issues that arise when CDNs are interconnected. For example, when a single CDN has the ability to distribute content on behalf of a CSP, there may be concerns that such content could be distributed to parties who are not authorized to receive it, and there are mechanisms to deal with such concerns. Our focus in this section is on how CDN interconnection introduces new security issues not found in the

single CDN case.

Many of the security issues that arise in CDNI are related to the transitivity of trust (or lack thereof) described in Section 6. As noted above, the design of the various interfaces for CDNI must take account of the additional risks posed by the fact that a CDN with whom a CSP has no direct relationship is now potentially distributing content for that CSP. The mechanisms used to mitigate these risks may be similar to those used in the single CDN case, but their suitability in this more complex environment must be validated.

Another concern that arises in any CDN is that information about the behavior of users (what content they access, how much content they consume, etc.) may be gathered by the CDN. This risk certainly exists in inter-connected CDNs, but it should be possible to apply the same techniques to mitigate it as in the single CDN case.

CDNs today offer a variety of means to control access to content, such as time-of-day restrictions, geo-blocking, and URI signing. These mechanisms must continue to function in CDNI environments, and this consideration is likely to affect the design of certain CDNI interfaces (e.g. metadata, request routing.)

Just as with a single CDN, each peer CDN must ensure that it is not used as an "open proxy" to deliver content on behalf of a malicious CSP. Whereas a single CDN typically addresses this problem by having CSPs explicitly register content (or origin servers) that is to be served, simply propagating this information to peer downstream CDNs may be problematic because it reveals more information than the upstream CDN is willing to specify. (To this end, the content acquisition step in the earlier examples force the dCDN to retrieve content from the uCDN rather than go directly to the origin server.)

There are several approaches to this problem. One is for the uCDN to encoded a signed token generated from a shared secret in each URL routed to a dCDN, and for the dCDN to validate the request based on this token. Another one is to have each upstream CDN advertise the set of CDN-domains they serve, where the downstream CDN checks each request against this set before caching and delivering the associated object. Although straightforward, this approach requires operators to reveal additional information, which may or may not be an issue.

8.1. Security of CDNI Interfaces

It is noted in [I-D.ietf-cdni-requirements] that all CDNI interfaces must be able to operate securely over insecure IP networks. Since it is expected that the CDNI interfaces will be implemented using existing application protocols such as HTTP or XMPP, we also expect

that the security mechanisms available to those protocols may be used by the CDNI interfaces. Details of how these interfaces are secured will be specified in the relevant interface documents.

8.2. Digital Rights Management

Issues of digital rights management (DRM, also sometimes called digital restrictions management) is often employed for content distributed via CDNs. In general, DRM relies on the CDN to distribute encrypted content, with decryption keys distributed to users by some other means (e.g. directly from the CSP to the end user.) For this reason, DRM is considered out of scope for the CDNI WG [I-D.ietf-cdni-problem-statement] and does not introduce additional security issues for CDNI.

9. Contributors

The following individuals contributed to this document:

- Francois le Faucheur
- Ben Niven-Jenkins
- David Ferguson
- John Hartman

10. Acknowledgements

We thank Aaron Falk and Huw Jones for their helpful input to the draft.

11. Informative References

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-01 (work in progress), October 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-01 (work in progress), October 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

[I-D.previdi-cdni-footprint-advertisement]
Previdi, S., Faucheur, F., Faucheur, L., and J. Medved, "CDNI Footprint Advertisement", draft-previdi-cdni-footprint-advertisement-00 (work in progress), October 2011.

[I-D.seedorf-alto-for-cdni]
Seedorf, J., "ALTO for CDNi Request Routing", draft-seedorf-alto-for-cdni-00 (work in progress), October 2011.

[I-D.vandergaast-edns-client-subnet]
Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client subnet in DNS requests", draft-vandergaast-edns-client-subnet-00 (work in progress), January 2011.

[I-D.xiaoyan-cdni-request-routing-protocol]
He, X., Li, J., Dawkins, S., and G. Chen, "Request Routing Protocol for CDN Interconnection", draft-xiaoyan-cdni-request-routing-protocol-00 (work in progress), October 2011.

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

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

Authors' Addresses

Bruce Davie (editor)
Cisco Systems, Inc.
1414 Mass. Ave.
Boxborough, MA 01719
USA

Email: bsd@cisco.com

Larry Peterson (editor)
Verivue, Inc.
2 Research Way
Princeton, NJ
USA

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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 2, 2012

M.O. van Deventer
R. van Brandenburg
TNO

July 1, 2011

Content Terminology in CDN Interconnection
draft-deventer-cdni-content-terminology-01

Abstract

This internet-draft describes how the term content might take on various meanings in different Content Delivery Network (CDN) interconnection scenarios. In order to solve this ambiguity, some additional terminology to describe content in CDNs is introduced, in alignment with terminology developed by the ETSI MCD CDN-I Working Group.

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 January 2, 2012.

Copyright Notice

Table of Contents

1. Introduction

The goal of this document is to present some terminology to be used when talking about content in the different stages of CDN interconnection. The additional terminology is aligned with CDN interconnect terminology developed by the ETSI MCD Working Group on CDN Interconnection (MCD WG CDN-I). Documents developed by ETSI MCD in the CDN area are publicly available from the ETSI MCD Open Area http://docbox.etsi.org/MCD/Open/.

This document introduces a metaphor for the electronic content delivery ecosystem to provide a better understanding of the different phases of content in this ecosystem. The metaphor of a newspaper distribution ecosystem was chosen, as that ecosystem is well understood and it has many aspects in common with content delivery. Of course, any metaphor should be used with care, as there will always be (subtle) differences.

1.1. Terminology

This document uses the terminology defined in section 1.1 of [I-D.jenkins-cdni-problem-statement] and [I-D.bertrand-cdni-use-cases] and terminology used in [ETSI TS 102 990 v0.0.7].

2. Newspaper metaphor

Using the metaphor of the newspaper distribution ecosystem, this section will describe why the term Content might have a different meaning depending on which phase in the Content Delivery chain one is talking about.

2.1. Introduction to newspaper metaphor

A Newspaper company can be compared to a Content Provider (or Content Aggregator); its business is to generate and/or collect news items and other articles, and bring those together into newspapers. The audience for the newspapers are Newspaper Readers (End Users). Just as Content Providers, different Newspaper Companies may have different business models, including subscription-based, advertising-based or other.

Getting newspapers to Newspaper Readers is essential to the Newspaper Company, just as it essential for a Content Provider to bring content to an End User. Some Newspapers Companies may have their own printing press, trucks to move the printed newspapers, outlets for Readers to pick up the newspapers and personnel to get newspapers delivered to mailboxes of Newspaper Readers. However, Newspaper Companies may also decide to outsource some or all of those newspaper-distribution

activities to specialized companies.

- Printing Companies have one or more printing presses to print newspapers.
- Postal Companies have one or more trucks and personnel to move and deliver printed newspapers.
- Newsstand Companies have one or more newsstands where people can pick up newspapers.

These different companies in the Newspaper delivery chain can be compared to the different roles in the Content Delivery Network chain. There is a role for companies to replicate, or transcode, content. There is a role for companies to perform the large scale distribution of content and there is a role for companies to perform the actual delivery of content to End Users. Just as is common in the newspaper world, companies in the Content Delivery world may take on multiple roles.

2.2. Different stages of content

While the contents of the newspaper do not change during the newspaper distribution path from Newspaper Company all the way to Newspaper Reader, the form the newspaper takes during the distribution path does change. The Newspaper Company does not directly create the final newspaper that Readers find on their doorstep. What the Newspaper Company creates and delivers to the Printing Company is a printing plate, a mastercopy, of the newspaper. The Printing Company then uses this printing plate to create a number of replicas (mastercopy replicas) to use in all of its printing facilities. Based on these replicas, the actual newspapers (the consumables) are printed. These printed newspapers are then distributed by a network of trucks to one or more distribution centers. How the newspapers end up in the hands of to the Newspaper Reader depends on the preferences (and location) of that Reader. Some Readers might pick it up at a Newsstand Company, while others will have it delivered by a Postal Company.

A path similar to the one taken by the newspaper is also taken by a piece of Content in the Content Delivery Network chain. The Content Provider does not necessarily create the final piece of content, or the form it takes, that is received by the End User. Along the way the Content might be transcoded to use a different codec, repackaged in a different video container, and delivered using different transport mechanisms and protocols.

In some situations, it might be useful to be able to distinguish

between the different fases and forms content goes through in the content delivery chain. For example, in the case of CDN Interconnection: When talking about how an Upstream CDN ingests content into a Downstream CDN, it is important to know what the term 'content' in this case means. Is it the content in the form as it was first ingested into the Upstream CDN by the Content Provider? Or is it the content in its multiple transcoded forms as stored on the Upstream CDN?

The current terminology does not allow for distinguishing between these different fases and forms of content in the content delivery chain.

3. Content Terminology

As shown in the previous section, the term content might take on a different meaning depending on which phase in the delivery chain one is talking about. It would therefore be useful to create some additional terminology to describe these different content phases. What follows is a first attempt at describing the different forms of content as found in a CDN.

Mastercopy: The content as it is delivered by the Content Provider to the (Upstream) CDN during the Content Ingestion process.

Replica: The content as it is transferred between the Upstream CDN and the Downstream CDN.

Consumable: The content as it is stored on a CDN delivery node directly prior to being delivered to an End User.

Note that, depending on the agreed arrangements with the Content Provider, a Consumable might be a repackaged or transcoded version from the origal Mastercopy or Replica in order to make it suitable for a specific type of end device. Furthermore, a Consumable might be a single file or consist of multiple files/segments that are the result of a segmentation operation having been performed on the content, e.g. to allow for specific transport mechanisms such as HTTP Adapative Streaming (HAS) to be used.

It should be noted that this terminology is purely meant for indicating the role of a particular piece of content in a particular situation; it does not mean that a Consumable and a Mastercopy cannot be bit-for-bit equivalent.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

This memo includes no security considerations.

6. Informative References

[I-D.bertrand-cdni-use-cases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., and P. Eardley, "Use Cases for Content Distribution Network Interconnection", draft-bertrand-cdni-use-cases-01 (work in progress), January 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.

[ETSI TS 102 990 v0.0.7]
ETSI MCD CDN-I Working Group, "Media Content Distribution (MCD); Media CDN Interconnection, use cases and requirements", http://docbox.etsi.org/MCD/Open/Latest_Drafts/ts_102990v000007p.pdf

Authors' Addresses

M. Oskar van Deventer
TNO
Brassersplein 2
Delft,
the Netherlands

Phone: +31 6 51 914 918
Email: oskar.vandeventer@tno.nl

Ray van Brandenburg
TNO
Brassersplein 2
Delft,
the Netherlands

Phone: +31 88 86 63609
Email: ray.vanbrandenburg@tno.nl

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2011

B. Niven-Jenkins
Velocix (Alcatel-Lucent)
F. Le Faucheur
Cisco
N. Bitar
Verizon

March 13, 2011

Content Distribution Network Interconnection (CDNI) Problem Statement
draft-jenkins-cdni-problem-statement-02

Abstract

Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for End Users and increased robustness of delivery. For these reasons they are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an end user regardless of that end user's location or attachment network. This creates a requirement for interconnecting standalone CDNs so they can interoperate as an open content delivery infrastructure for the end-to-end delivery of content from Content Service Providers (CSPs) to end users. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

The goal of this document is to outline the problem area for the IETF with a view towards creating a working group. This working group would work on interoperable and scalable solutions for CDN interconnection.

Requirements Language

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].

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 September 14, 2011.

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for end users and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. However, the footprint of a given CDN in charge of delivering a given content may not expand close enough to the End User's current location or attachment network to realize the cost benefit and user experience that a more distributed CDN would provide. This creates a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

The goal of this document is to outline the problem area for the IETF with a view towards creating a working group. This working group would work on interoperable and scalable solutions for CDN interconnection.

Section 2 discusses the use cases for CDN interconnection. Section 3 presents the CDNI model and problem area to be considered by the IETF. Section 4 discusses how existing protocols can be reused to define the CDNI protocols while Section 5 proposes to focus the scope for the initial charter of a CDNI Working Group to the minimum functional elements necessary for basic CDN interconnection. Section 5 provides a gap analysis of the work of other standards organization and finally Section 5 discusses the relationship with relevant IETF Working Groups.

1.1. Terminology

This document uses the following terms:

Content: Any form of digital data. One important form of Content with additional constraints on Distribution and Delivery is continuous media (i.e. where there is a timing relationship between source and sink).

Metadata: Metadata in general is data about data.

Content Metadata: This is metadata about Content. Content Metadata comprises:

1. Metadata that is relevant to the distribution of the content (and therefore relevant to a CDN involved in the delivery of that content). We refer to this type of metadata as "Content Distribution Metadata". See also the definition of Content Distribution Metadata.
2. Metadata that is associated with the actual Content (and not directly relevant to the distribution of that Content) or content representation. For example, such metadata may include information pertaining to the Content's genre, cast, rating, etc as well as information pertaining to the Content representation's resolution, aspect ratio, etc.

Content Distribution Metadata: The subset of Content Metadata that is relevant to the distribution of the content. This is the metadata required by a CDN in order to enable and control content distribution and delivery by the CDN. In a CDN Interconnection environment, some of the Content Distribution Metadata may have an intra-CDN scope (and therefore need not be communicated between CDNs), while some of the Content Distribution Metadata have an inter-CDN scope (and therefore needs to be communicated between CDNs).

CDNI Metadata: Content Distribution Metadata with inter-CDN scope. For example, CDNI Metadata may include geo-blocking information (i.e. information defining geographical areas where the content is to be made available or blocked), availability windows (i.e. information defining time windows during which the content is to be made available or blocked) and access control mechanisms to be enforced (e.g. URI signature validation). CDNI Metadata may also include information about desired distribution policy (e.g. prepositioned vs dynamic acquisition) and about where/how a CDN can acquire the content. CDNI Metadata may also include content management information (e.g. request for deletion of Content from Surrogates) across interconnected CDNs.

Dynamic content acquisition: Dynamic content acquisition is where a CDN acquires content from the content source in response to an End User requesting that content from the CDN. In the context of CDN Interconnect, dynamic acquisition means that a downstream CDN does not acquire the content from content sources (including upstream CDNs) until a request for that content has been delegated to the downstream CDN by an Upstream CDN.

Dynamic CDNI metadata acquisition: In the context of CDN Interconnect, dynamic CDNI metadata acquisition means that a downstream CDN does not acquire CDNI metadata for content from the

upstream CDN until a request for that content has been delegated to the downstream CDN by an Upstream CDN.

Pre-Positioned content acquisition: Content Pre-positioning is where a CDN acquires content from the content source prior to or independent of any End User requesting that content from the CDN. In the context of CDN interconnect the Upstream CDN instructs the Downstream CDN to acquire the content from content sources (including upstream CDNs) in advance of or indepedent of any End User requesting it.

Pre-positioned CDNI Metadata acquisition: In the context of CDN Interconnect, Metadata Pre-positioning is where the the Downstream CDN acquires distribution metadata for content prior to or independent of any End User requesting that content from the Downstream CDN.

End User (EU): The 'real' user of the system, typically a human but maybe some combination of hardware and/or software emulating a human (e.g. for automated quality monitoring etc.)

User Agent (UA): Software (or a combination of hardware and software) through which the End User interacts with the Content Service. The User Agent will communicate with the CSP's Service for the selection of content and one or more CDNs for the delivery of the Content. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of User Agents (non-exhaustive) are: Browsers, Set Top Boxes (STB), Dedicated content applications (e.g. media players), etc.

Network Service Provider (NSP): Provides network-based connectivity/ services to Users.

Content Service Provider (CSP): Provides a Content Service to End Users (which they access via a User Agent). A CSP may own the Content made available as part of the Content Service, or may license content rights from another party.

Content Service: The service offered by a Content Service Provider. The Content Service encompasses the complete service which may be wider than just the delivery of items of Content, e.g. the Content Service also includes any middleware, key distribution, program guide, etc. which may not require any direct interaction with the CDN.

Content Distribution Network (CDN) / Content Delivery Network (CDN): Network infrastructure in which the network elements cooperate at layers 4 through layer 7 for more effective delivery of Content to

User Agents. Typically a CDN consists of a Request Routing system, a Distribution System (that includes a set of Surrogates), a Logging System and a CDN control system .

CDN Provider: The service provider who operates a CDN. Note that a given entity may operate in more than one role. For example, a company may simultaneously operate as a Content Service Provider, a Network Service Provider and a CDN Provider.

CDN Interconnect (CDNI): The set of interfaces over which two or more CDNs communicate with each other in order to achieve the delivery of content to User Agents by Surrogates in one CDN (the downstream CDN) on behalf of another CDN (the upstream CDN).

Upstream CDN: For a given user request, the CDN (within a pair of directly interconnected CDNs) that redirects the request to the other CDN.

Downstream CDN: For a given user request, the CDN (within a pair of directly interconnected CDNs) to which the request is redirected by the other CDN (the Upstream CDN). Note that in the case of successive redirections (e.g. CDN1-->CDN2-->CDN3) a given CDN (e.g. CDN2) may act as the Downstream CDN for a redirection (e.g. CDN1-->CDN2) and as the Upstream CDN for the subsequent redirection of the same request (e.g. CDN2-->CDN3).

Over-the-top (OTT): A service, e.g. a CDN, operated by a different operator than the NSP to which the users of that service are attached.

Surrogate: A device/function that interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

Request Routing System: The function within a CDN responsible for receiving a content request from a user agent, obtaining and maintaining necessary information about a set of candidate surrogates or candidate CDNs, and for selecting and redirecting the user to the appropriate surrogate or CDN. To enable CDN Interconnect, the Request Routing System must also be capable of handling user agent content requests passed to it by another CDN.

Distribution System: the function within a CDN responsible for distributing Content Distribution Metadata as well as content inside the CDN (e.g. down to the surrogates)

Delivery: the function within CDN surrogates responsible for delivering a piece of content to the User Agent. For example,

delivery may be based on HTTP progressive download or HTTP adaptive streaming.

Logging System: the function within a CDN responsible for collecting measurement and recording of distribution and delivery activities. The information recorded by the logging system may be used for various purposes including charging (e.g. of the CSP), analytics and monitoring.

1.2. CDN Background

Readers are assumed to be familiar with the architecture, features and operation of CDNs. For readers less familiar with the operation of CDNs, the following resources may be useful:

- RFC 3040 [RFC3040] describes many of the component technologies that are used in the construction of a CDN
- Taxonomy [TAXONOMY] compares the architecture of a number of CDNs
- RFC 3466 [RFC3466] and RFC 3570 [RFC3570] are the output of the IETF Content Delivery Internetworking (CDI) working group which was closed in 2003.

Note: Some of the terms used in this document are similar to terms used the above referenced documents. When reading this document terms should be interpreted as having the definitions provided in Section 1.1.

2. CDN Interconnect Use Cases

An increasing number of NSPs are deploying CDNs in order to deal cost-effectively with the growing usage of on-demand video services and other content delivery applications.

CDNs allow caching of content closer to the edge so that a given item of content can be delivered by a CDN Surrogate (i.e. a cache) to multiple User Agents (and their End Users) without transiting multiple times through the network core (i.e from the content origin to the surrogate). This contributes to bandwidth cost reductions for the NSP and to improved quality of experience for the end users. CDNs also enable replication of popular content across many surrogates, which enables content to be served to large numbers of User Agents concurrently. This also helps dealing with situations such as flash crowds and denial of service attacks.

The CDNs deployed by NSPs are not just restricted to the delivery of content to support the Network Service Provider’s own ’walled garden’ services, such as IP delivery of television services to Set Top

Boxes, but are also used for delivery of content to other devices including PCs, tablets, mobile phones etc.

Some service providers operate over multiple geographies and federate multiple affiliate NSPs. These NSPs typically operate independent CDNs. As they evolve their services (e.g. for seamless support of content services to nomadic users across affiliate NSPs) there is a need for interconnection of these CDNs. However there are no open specifications, nor common best practices, defining how to achieve such CDN interconnection.

CSPs have a desire to be able to get (some of) their content to very large number of End Users and/or over many/all geographies and/or with a high quality of experience, all without having to maintain direct business relationships with many different CDN providers (or having to extend their own CDN to a large number of locations). Some NSPs are considering interconnecting their respective CDNs (as well as possibly over-the-top CDNs) so that this collective infrastructure can address the requirements of CSPs in a cost effective manner. In particular, this would enable the CSPs to benefit from on-net delivery (i.e. within the Network Service Provider's own network/CDN footprint) whenever possible and off-net delivery otherwise, without requiring the CSPs to maintain direct business relationships with all the CDNs involved in the delivery. Again, for this requirement, CDN operators (NSPs or over-the-top CDN operators) are faced with a lack of open specifications and best practices.

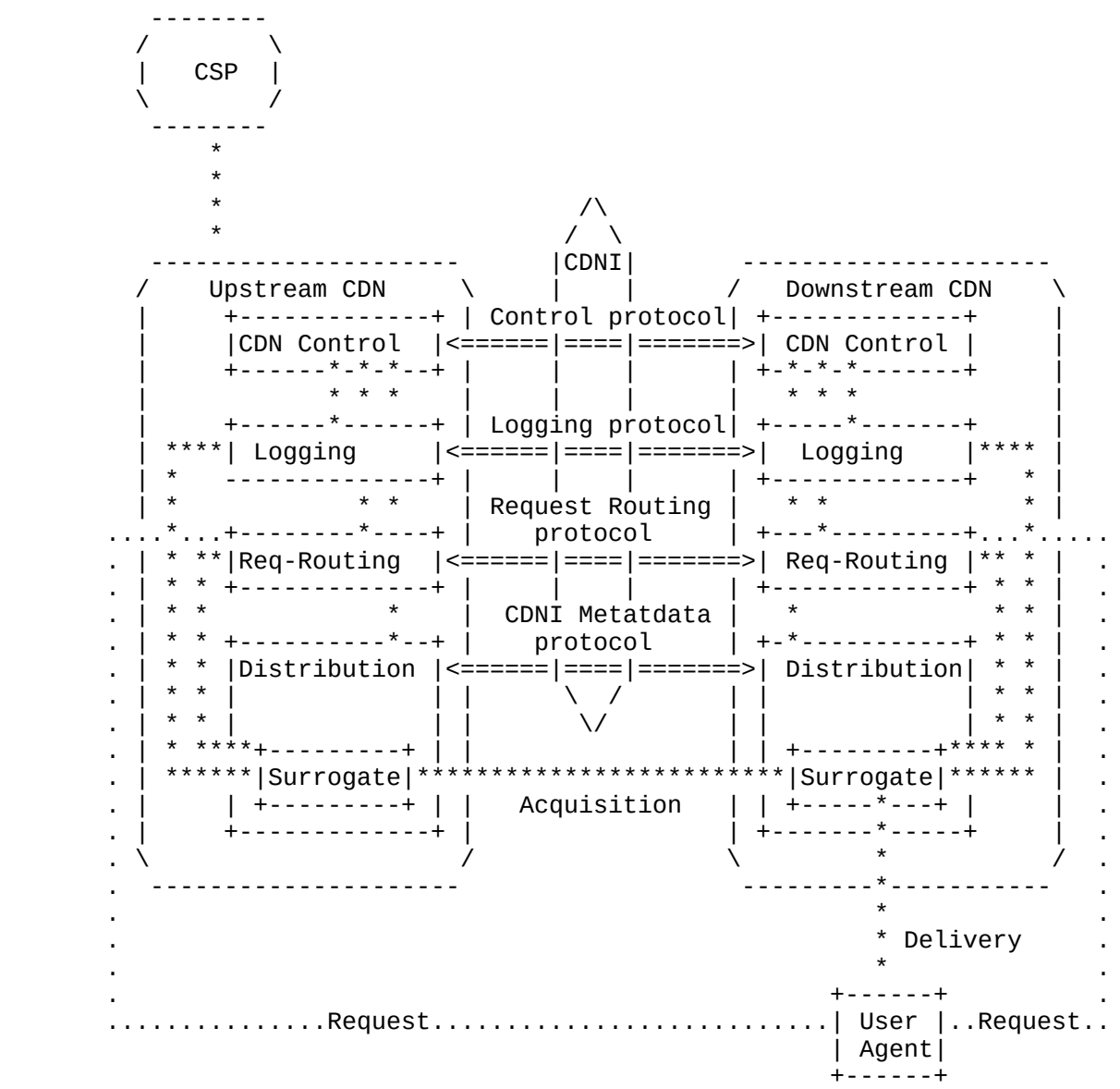
NSPs have often deployed CDNs as specialized cost-reduction projects within the context of a particular service or environment, some NSPs operate separate CDNs for separate services. For example, there may be a CDN for managed IPTV service delivery, a CDN for web-TV delivery and a CDN for video delivery to Mobile terminals. As NSPs integrate their service portfolio, there is a need for interconnecting these CDNs. Again, NSPs face the problem of lack of open interfaces for CDN interconnection.

For operational reasons (e.g. disaster, flash crowd) or commercial reasons, an over-the-top CDN may elect to make use of another CDN (e.g. an NSP CDN with on-net Surrogates for a given footprint) for serving a subset of the user requests (e.g. requests from users attached to that NSP). Again, for this requirement, CDN operators (over-the-top CDN operators or NSPs) are faced with a lack of open specifications and best practices.

Use cases for CDN Interconnection are further discussed in [I-D.bertrand-cdni-use-cases] (which contains a merged set of use cases previously presented in [I-D.watson-cdni-use-cases] and [I-D.bertrand-cdni-use-cases-00]).

3. CDN Interconnect Model & Problem Area for IETF

Interconnecting CDNs involves interactions among multiple different functions and components that form each CDN. Only some of those require standardization. The CDNI model and problem area proposed for IETF work is illustrated in Figure 1. The candidate problem area (and respectively the non-goals) for IETF work on CDN Interconnection are discussed in Section 3.1 (and respectively Section 3.2).



<==> interfaces inside the scope of CDNI

**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Problem Area

3.1. Candidate CDNI Problem Area for IETF

Listed below are the four protocols required to interconnect a pair of CDNs and that constitute the problem space that is proposed to be addressed by a potential CDNI working group in the IETF. The use of the term "protocol" is meant to encompass the protocol over which CDNI data representations (e.g. CDNI Metadata records) are exchanged as well as the specification of the data representations themselves (i.e. what properties/fields each record contains, its structure, etc.). While "interface" may be a more accurate term, the term "protocol" is retained in this document because of its common use.

- o CDNI Control protocol: This protocol allows the "CDNI Control" system in interconnected CDNs to communicate. This protocol may support the following:
  - * Allow bootstrapping of the other CDNI protocols (e.g. protocol address discovery and establishment of security associations).
  - * Allow configuration of the other CDNI protocols (e.g. Upstream CDN specifies information to be reported through the CDNI Logging protocol).
  - * Allow the downstream CDN to communicate static (or fairly static) information about its delivery capabilities and policies.
  - * Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).
  - * Allow upstream CDN to initiate or request specific actions to be undertaken in the downstream CDN. For example, this may include the following capabilities:
    - + Allow an upstream CDN to request that content files and/or CDNI Metadata that it shared, be purged from, or invalidated in, a downstream CDN. Support for content deletion or invalidation from a CDN is a key requirement for some Content Service Providers in order, amongst other use cases for content deletion, to support the content rights agreements they have negotiated. Today’s CDNs use proprietary control interfaces to enable CSPs to remove content cached in the CDN and therefore there is a need to have a similar but standardized content deletion capability between interconnected CDNs.
    - + Allow an upstream CDN to initiate Pre-postioned content acquisition and/or Pre-positioned CDN Metadata acquisition in a downstream CDN.
- o CDNI Request Routing protocol: This protocol allows the Request Routing system in interconnected CDNs to communicate to ensure that an end user request can be (re)directed from an upstream CDN to a surrogate in the downstream CDN, in particular where selection responsibilities may be split across CDNs (for example

the upstream CDN may be responsible for selecting the downstream CDN while the downstream CDN may be responsible for selecting the actual surrogate within that CDN). In particular, the CDN Request Routing protocol, may support the following:

* allow the upstream CDN to query the downstream CDN at request-routing time before redirecting the request to the downstream CDN
* allow the downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents.

- o CDNI Metadata distribution protocol: This protocol allows the Distribution system in interconnected CDNs to communicate to ensure CDNI Metadata can be exchanged across CDNs. See Section 1.1 for definition and examples of CDNI Metadata.
- o CDNI Logging protocol: This protocol allows the Logging system in interconnected CDNs to communicate the relevant activity logs in order to allow log consuming applications to operate in a multi-CDN environments. For example, an upstream CDN may collect delivery logs from a downstream CDN in order to perform consolidated charging of the CSP or for settlement purposes across CDNs. Similarly, an upstream CDN may collect delivery logs from a downstream CDN in order to provide consolidated reporting and monitoring to the CSP.

Note that the actual grouping of functionalities under these four protocols is considered tentative at this stage and may be changed after further study (e.g. some subset of functionality be moved from one protocol into another).

The above list covers a significant potential problem space, in part because in order to interconnect two CDNs there are several 'touch points' that require standardization. However, it is expected that the CDNI protocols need not be defined from scratch and instead can very significantly reuse or leverage existing protocols: this is discussed further in Section 4. Also, it is expected that the items above will be prioritized so that the CDNI Working Group can focus (at least initially) on the most essential and urgent work: this is discussed further in Section 5.

As part of the development of the CDNI protocols and solutions it will also be necessary to agree on common mechanisms for how to identify and name the data objects that are to be interchanged between interconnected CDNs, as well as how to describe which policy should be used when doing so. [I-D.jenkins-cdni-names] presents one view on how CDN data types/objects could be classified such that the problem space of their naming and referencing is not as large as it

might at first appear because there is significant commonality between the different data types/objects required for CDNI.

Some NSPs have started to perform experiments to explore whether their CDN use cases can already be addressed with existing CDN implementations. One set of such experiments is documented in [I-D.bertrand-cdni-experiments]. The conclusions of those experiments are that while some basic limited CDN Interconnection functionality can be achieved with existing CDN technology, the current lack of any standardized CDNI interfaces/protocols such as those discussed in this document is preventing the deployment of production CDN Interconnection solutions with the necessary level of functionality.

3.2. Non-Goals for IETF

Listed below are aspects of content delivery that the authors propose be kept outside of the scope of a potential CDNI working group:

- The interface between Content Service Provider and the Authoritative CDN (i.e. the upstream CDN contracted by the CSP for delivery by this CDN or by its downstream CDNs).
- The delivery interface between the delivering CDN surrogate and the User Agent, such as streaming protocols.
- The request interface between the User Agent and the request-routing system of a given CDN. Existing IETF protocols (e.g. HTTP, RTSP, DNS) are commonly used by User Agents to request content from a CDN and by CDN request routing systems to redirect the User Agent requests. The CDNI working group need not define new protocols for this purpose. Note however, that the CDNI control plane protocol may indirectly affect some of the information exchanged through the request interface (e.g. URI).
- The content acquisition interface between CDNs (i.e. the data plane interface for actual delivery of a piece of content from one CDN to the other). This is expected to use existing protocols such as HTTP or protocols defined in other forums for content acquisition between an origin server and a CDN (e.g. HTTP-based C2 reference point of ATIS IIF CoD). The CDN Interconnection solution may only concern itself with the agreement/negotiation aspects of which content acquisition protocol is to be used between two interconnected CDNs in view of facilitating interoperability.
- End User/User Agent Authentication. End User/User Agent authentication and authorization are the responsibility of the Content Service Provider.
- Content preparation, including encoding and transcoding. The CDNI architecture aims at allowing distribution across interconnected CDNs of content treated as opaque objects. Interpretation and processing of the objects, as well as optimized delivery of these

objects by the surrogate to the end user are outside the scope of CDNI.

- o Digital Rights Management (DRM). DRM is an end-to-end issue between a content protection system and the User Agent.
- o Applications consuming CDNI logs (e.g. charging, analytics, reporting,...).
- o Internal CDN Protocols. i.e. protocols within one CDN.
- o Scalability of individual CDNs. While scalability of the CDNI protocols/approach is in scope, how an individual CDN scales is out of scope.
- o Actual algorithms for selection of CDNs or Surrogates by Request Routing systems (however, some specific parameters required as input to these algorithms may be in scope when they need to be communicated across CDNs).
- o Surrogate algorithms. For example caching algorithms and content acquisition methods are outside the scope of the CDNI work. Content management (e.g. Content Deletion) as it relates to CDNI content management policies, is in scope but the internal algorithms used by a cache to determine when to no longer cache an item of Content (in the absence of any specific metadata to the contrary) is out of scope.
- o Element management interfaces.
- o Commercial, business and legal aspects related to the interconnections of CDNs.

The third bullet in the list above places the acquisition of content between interconnected CDNs as out of scope for CDNI and deserves some additional explanation. The consequence of such a decision is that a CDNI WG would be focussed on only defining the control plane for CDNI; and the CDNI data plane (i.e. the acquisition & distribution of the actual content objects) would not be addressed by a CDNI WG. The rationale for such a decision is that CDNs today typically already use standardized protocols such as HTTP, FTP, rsync, etc. to acquire content from their CSP customers and it is expected that the same protocols could be used for acquisition between interconnected CDNs. Therefore the problem of content acquisition is considered already solved and all that is required from a CDNI WG is describing within the CDNI Metadata where to go and which protocol to use to retrieve the content.

4. Design Approach for Realizing the CDNI APIs

This section expands on how CDNI protocols can reuse and leverage existing protocols. First the "reuse instead of reinvent" design principle is restated, then each protocol is discussed individually with example candidate protocols that can be considered for reuse or leverage. This discussion is not intended to pre-empt any WG

decision as to the most appropriate protocols, technologies and solutions to select to solve CDNI but is intended as an illustration of the fact that these protocols need not be created in a vacuum and that reuse or leverage of existing protocols is likely possible.

4.1. Relationship to the OSI network model

The four CDNI protocols (CDNI Control protocol, CDNI Request Routing protocol, CDNI Metadata protocol, CDNI Logging protocol) described in Section 3.1 within the CDNI problem area are all control plane interfaces operating at the application layer (Layer 7 in the OSI network model). Since it is not expected that these protocols would exhibit unique session, transport or network requirements as compared to the many other existing applications in the Internet, it is expected that the CDNI protocols will be defined on top of existing session, transport and network protocols.

4.2. "Reuse Instead of Reinvent" Principle

Although a new application protocol could be designed specifically for CDNI we assume that this is unnecessary and it is recommended that existing application protocols be reused or leveraged (HTTP [RFC2616], Atom Publishing Protocol [RFC5023], XMPP [RFC3920], for example) to realize the CDNI protocols.

4.3. CDNI Request Routing API

The CDNI Request Routing protocol enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request and to allow the downstream CDN to control what the upstream Request Routing function should return to the User Agent in the redirection message.

The CDNI Request Routing protocol needs to offer a mechanism for an upstream CDN to issue a "Redirection Request" to a downstream CDN. The Request Routing protocol needs to be able to support scenarios where the initial User Agent request to the upstream CDN is received over DNS as well as over a content specific application protocol (e.g. HTTP, RTSP, RTMP, etc.).

Therefore a Redirection Request needs to contain information such as:

- The protocol (e.g. DNS, HTTP) over which the upstream CDN received the initial User Agent request
- Additional details of the User Agent request that are required to perform effective Request Routing by the Downstream CDN. For DNS this would typically be the IP address of the DNS resolver making

the request on behalf of the User Agent. For requests received over content specific application protocols the Redirection Request could contain significantly more information related to the original User Agent request but at a minimum would need to contain the User Agent's IP address, the equivalent of the HTTP Host header and the equivalent of the HTTP abs_path defined in [RFC2616].

It should be noted that, the CDNI architecture needs to consider that a downstream CDN may receive requests from User Agents without first receiving a Redirection Request from an upstream CDN, for example because:

- o User Agents (or DNS resolvers) may cache DNS or application responses from Request Routers.
- o Responses to Redirection Requests over the Request Routing protocol may be cacheable.
- o Some CDNs may want broader policies, e.g. CDN B agrees to always take CDN A's delegated redirection requests, in which case the necessary redirection details are exchanged out of band (of the CDNI protocols), e.g. configured.

On receiving a Redirection Request, the downstream CDN will use the information provided in the request to determine if it is able (and willing) to accept the delegated content request and needs to return the result of its decision to the upstream CDN.

Thus, a Redirection Response from the downstream CDN needs to contain information such as:

- o Status code indicating acceptance or rejection (possibly with accompanying reasons).
- o Information to allow redirection by the Upstream CDN. In the case of DNS-based request routing, this is expected to include the equivalent of a DNS record(s) (e.g. a CNAME) that the upstream CDN should return to the requesting DNS resolver. In the case of application based request routing, this is expected to include the application specific redirection response(s) to return to the requesting User Agent. For HTTP requests from User Agents this could be in the form of a URI that the upstream CDN could return in a HTTP 302 response.

The CDNI Request Routing protocol is therefore a fairly straightforward request/response protocol and could be implemented over any number of request/response protocols. For example, it may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.). This removes the need for a CDNI WG to define a new protocol for the

request/response element of the Request Routing protocol. Thus, a CDNI WG would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics and procedures that are specific to the CDNI Request Routing protocol (e.g. handling of malformed requests/ responses).
- The syntax (i.e representation/encoding) of the redirection requests and responses.
- The semantics (i.e. meaning and expected contents) of the redirection requests and responses.

4.4. CDNI Metadata API

The CDNI Metadata protocol enables the Metadata function in a downstream CDN to obtain CDNI Metadata from an upstream CDN so that the downstream CDN can properly process and respond to:

- Redirection Requests received over the CDNI Request Routing protocol.
- Content Requests received directly from User Agents.

The CDNI Metadata protocol needs to offer a mechanism for an Upstream CDN to:

- distribute/update/remove CDNI Metadata to a Downstream CDN

and/or to allow a downstream CDN to:

- Make direct requests for CDNI Metadata records where the downstream CDN knows the identity of the Metadata record(s) it requires.
- Search for CDNI Metadata records where the downstream CDN does not know the specific Metadata record(s) it requires but does know some property of the record it is searching for. For example, it may know the value of the HTTP Host header received in a HTTP request and it wants to obtain the CDNI Metadata for that host so that it can determine how to further process the received HTTP request.

The CDNI Metadata protocol is therefore similar to the CDNI Request Routing protocol because it is a request/response protocol with the potential addition that CDNI Metadata search may have more complex semantics than a straightforward Request Routing redirection request. Therefore, like the CDNI Request Routing protocol, the CDNI Metadata protocol may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.) or possibly using other existing protocols such as XMPP [RFC3920]. This removes the need for a CDNI WG to define a new protocol for the

request/response element of the Metadata protocol.

Thus, a CDNI WG would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics that are specific to the CDNI Metadata protocol (e.g. handling of malformed requests/responses).
- The syntax (i.e representation/encoding) of the CDNI Metadata records that will be exchanged over the protocol.
- The semantics (i.e. meaning and expected contents) of the individual properties of a Metadata record.
- How the relationships between different CDNI Metadata records are represented.

4.5. CDNI Logging API

The CDNI Logging protocol enables details of logs or events to be exchanged between interconnected CDNs, where events could be:

- Log lines related to the delivery of content (similar to the log lines recorded in a web server's access log).
- Real-time or near-real time events before, during or after content delivery, e.g. content Start/Pause/Stop events, etc.
- Operations and diagnostic messages.

Within CDNs today, logs and events are used for a variety of purposes in addition to real-time and non real-time diagnostics and auditing by the CDN Operator and its customers. Specifically CDNs use logs to generate Call Data Records (CDRs) for passing to billing and payment systems and to real-time (and near real-time) analytics systems. Such use cases place requirements on the CDNI Logging protocol to support guaranteed and timely delivery of log messages between interconnected CDNs. It may also be necessary to be able to prove the integrity of received log messages.

Several protocols already exist that could potentially be used to exchange CDNI logs between interconnected CDNs including SNMP Traps, syslog, ftp, HTTP POST, etc. although it is likely that some of the candidate protocols may not be well suited to meet all the requirements of CDNI. For example SNMP traps pose scalability concerns and SNMP does not support guaranteed delivery of Traps and therefore could result in log records being lost and the consequent CDRs and billing records for that content delivery not being produced as well as that content delivery being invisible to any analytics platforms.

Although it is not necessary to define a new protocol for exchanging logs across the CDNI Logging protocol, a CDNI WG would still need to

specify:

- The recommended protocol to use.
- A default set of log fields and their syntax & semantics. Today there is no standard set of common log fields across different content delivery protocols and in some cases there is not even a standard set of log field names and values for different implementations of the same delivery protocol.
- A default set of events that trigger logs to be generated.

4.6. CDNI Control API

The CDNI Control protocol allows the "CDNI Control" system in interconnected CDNs to communicate. The exact inter-CDN control functionality required to be supported by the CDNI Control protocol is less well defined than the other three CDNI interfaces at this time.

However, as discussed in Section 3.1, the CDNI Control protocol may be required to support functionality similar to the following:

- Allow an upstream CDN and downstream CDN to establish, update or terminate their CDNI interconnection.
- Allow bootstrapping of the other CDNI protocols (e.g. protocol address discovery and establishment of security associations).
- Allow configuration of the other CDNI protocols (e.g. Upstream CDN specifies information to be reported through the CDNI Logging protocol).
- Allow the downstream CDN to communicate information about its delivery capabilities, resources and policies.
- Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).

It is expected that for the Control protocol also, existing protocols can be reused or leveraged. Those will be considered once the requirements for the Control protocol have been refined.

5. Prioritizing the CDNI Work

In order to manage the potential workload of a CDNI WG, it is recommended that the work be prioritized in a "walk before you run" approach.

The CDNI problem area can be categorized into different solution scopes as follows:

- "Base CDNI" Scope: This solution scope comprises the solution elements that can be considered as the ’minimum’ needed to actually deliver any content using interconnected CDNs. For

example, a base CDNI Request Routing protocol and a base CDNI Metadata protocol belong to this scope because without them the upstream CDN is unable to redirect User Agents to the downstream CDN and the downstream CDN is unable to obtain the delivery policies and other CDNI Metadata required to ingest and deliver the content.

- o "Operationalized CDNI" Scope: This solution scope comprises the solution elements that can be considered as the 'minimum' needed to 'operationalize' CDN Interconnects. For example, the CDNI Logging protocol and the base capabilities of the CDNI Control protocol (e.g. content file/metadata deletion) belong to this scope because without them CDN operators are required to substitute for them either with manual processes or proprietary interfaces.
- o "Enhanced CDNI" Scope: This solution scope comprises the solution elements that can be classed as 'enhanced features'. For example, the aspects of the CDNI Control protocol related to automatic bootstrapping and configuration belong to this scope.

It is proposed that these solution scopes be addressed primarily sequentially by a CDNI WG and that the initial charter be centered around the "Base CDNI" scope. However there is obvious benefit from having a solution for the "Base CDNI" scope that is amenable to extension for support of the "Operational" scope and "Enhanced" scope. Therefore it is proposed that the initial CDNI WG charter also includes definition of (at least) the main requirements for the "Operationalized CDNI" scope and "Enhanced CDNI" Scope, so those can be kept in mind when defining the solution for the "Base CDNI" scope.

6. Gap Analysis of relevant Standardization and Research Activities

There are a number of other standards bodies and industry forums that are working in areas related to CDN, and in some cases related to CDNI. This section will first outline the key standardization organizations undertaking related work, some related research projects, and will then outline any potential overlap with the proposed CDNI WG and any component that could potentially be reused by CDNI .

6.1. Related standardization activities

6.1.1. IETF CDI Working Group (Concluded)

The Content Distribution Internetworking (CDI) Working Group was formed in the IETF following a BoF in December 2000 and closed in mid 2003.

For convenience, here is an extract from the CDI WG charter [CDI-Charter]:

"

- The goal of this working group is to define protocols to allow the interoperation of separately-administered content networks.
- A content network is an architecture of network elements, arranged for efficient delivery of digital content. Such content includes, but is not limited to, web pages and images delivered via HTTP, and streaming or continuous media which are controlled by RTSP.
- The working group will first define requirements for three modes of content internetworking: interoperation of request-routing systems, interoperation of distribution systems, and interoperation of accounting systems. These requirements are intended to lead to a follow-on effort to define protocols for interoperation of these systems.
- In its initial form, the working group is not chartered to deliver those protocols [...]

"

Thus, the CDI WG touched on the same problem space as the present document.

The CDI WG published 3 Informational RFCs:

- RFC 3466 [RFC3466] - "A Model for Content Internetworking (CDI)".
- RFC 3568 [RFC3568] - "Known Content Network (CN) Request-Routing Mechanisms".
- RFC 3570 [RFC3570] - "Content Internetworking (CDI) Scenarios".

6.1.2. 3GPP

3GPP was the first organization that released a specification related to adaptive streaming over HTTP. 3GPP Release 9 specification on adaptive HTTP streaming was published in March 2010, and there have been some bug fixes on this specification since the publication. In addition, 3GPP is preparing an extended version for Release 10, which is scheduled to be published later in 2011. This release will include a number of clarifications, improvements and new features.

[3GP-DASH] is defined as a general framework independent of the data encapsulation format. It has support for fast initial startup and seeking, adaptive bitrate switching, re-use of HTTP origin and cache servers, re-use of existing media playout engines, on-demand, live and time-shifted delivery. It specifies syntax and semantics of Media Presentation Description (MPD), format of segments and delivery

protocol for segments. It does not specify content provisioning, client behavior or transport of MPD.

The content retrieved by a client using [3GP-DASH] adaptive streaming could be obtained from a CDN but this is not discussed or specified in the 3GPP specifications as it is transparent to [3GP-DASH] operations. Similarly, it is expected that [3GP-DASH] can be used transparently from the CDNs as a delivery protocol (between the delivering CDN surrogate and the User Agent) in a CDN Interconnect environment. [3GP-DASH] could also be a candidate for content acquisition between CDNs in a CDN Interconnect environment.

6.1.3. ISO MPEG

Within ISO MPEG, the Dynamic Adaptive Streaming over HTTP (DASH) ad-hoc group adopted the 3GPP Release 9 [3GP-DASH] specification as a starting point and has made some improvements and extensions. Similar to 3GPP SA4, the MPEG DASH ad-hoc group has been working on standardizing the manifest file and the delivery format. Additionally, the MPEG DASH ad-hoc group has also been working on the use of MPEG-2 Transport Streams as a media format, conversion from/to existing file formats, common encryption, and so on. The MPEG DASH specification could also be a candidate for delivery to the user agent and for content acquisition between CDNs in a CDN Interconnect environment. The Draft International Standard (DIS) version [MPEG-DASH] is currently publicly available since early February 2011.

In the 95th MPEG meeting in January 2011, the DASH ad-hoc group decided to start a new evaluation experiment called "CDN-EE". The goals are to understand the requirements for MPEG DASH to better support CDN-based delivery, and to provide a guidelines document for CDN operators to better support MPEG DASH streaming services. The ongoing work is still very preliminary and does not currently target looking into CDN Interconnect use cases.

6.1.4. ATIS IIF

ATIS ([ATIS]) IIF is the IPTV Interoperability Forum (within ATIS) that develops requirements, standards, and specifications for IPTV.

ATIS IIF is developing the "IPTV Content on Demand (CoD) Service" specification. This includes use of a CDN (referred to in ATIS IIF CoD as the "Content Distribution and Delivery Functions") for support of a Content on Demand (CoD) Service as part of a broader IPTV service. However, this only covers the case of a managed IPTV service (in particular where the CDN is administered by the service provider) and does not cover the use, or interconnection, of multiple

CDNs.

6.1.5. CableLabs

"Founded in 1988 by cable operating companies, Cable Television Laboratories, Inc. (CableLabs) is a non-profit research and development consortium that is dedicated to pursuing new cable telecommunications technologies and to helping its cable operator members integrate those technical advancements into their business objectives." [CableLabs]

CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project.

6.1.6. ETSI MCD

ETSI MCD (Media Content Distribution) is the ETSI technical committee "in charge of guiding and coordinating standardization work aiming at the successful overall development of multimedia systems (television and communication) responding to the present and future market requests on media content distribution".

MCD created a specific work item on interconnection of heterogeneous CDNs ("CDN Interconnection, use cases and requirements") in March 2010. MCD very recently created a working group to progress this work item. However, no protocol level work has yet started in MCD for CDN Interconnect.

6.1.7. ETSI TISPAN

ETSI TISPAN has published two sets of IPTV specifications, one of which is based on IMS. In addition, TISPAN is about to complete the specifications of a CDN architecture supporting delivery of various content services such as time-shifted TV and VoD to TISPAN devices (UEs) or regular PCs. The use cases allow for hierarchically and geographically distributed CDN scenarios, along with multi-CDN cooperation. As a result, the architecture contains reference points to support interconnection of other TISPAN CDNs. The protocol definition phase for the corresponding CDN architecture was kicked-off at the end of 2010. In line with its long history of leveraging IETF protocols, ETSI could potentially leverage CDNI protocols developed in the IETF for their related protocol level work on interconnections of CDNs.

6.1.8. ITU-T

SG13 is developing standards related to the support of IPTV services (i.e.. multimedia services such as television/VoD/audio/text/

graphics/data delivered over IP-based managed networks).

ITU-T Recommendation Y.1910 [Y.1910] provides the description of the IPTV functional architecture. This architecture includes functions and interfaces for the distribution and delivery of content. This architecture is aligned with the ATIS IIF architecture.

Based upon ITU-T Rec. Y.1910, ITU-T Rec. Y.2019 [Y.2019] describes in more detail the content delivery functional architecture. This architecture allows CDN Interconnection: some interfaces (such as D3, D4) at the control level allow relationships between different CDNs, in the same domain or in different domains. Generic procedures are described, but the choice of the protocols is open.

6.1.9. Open IPTV Forum (OIPF)

The Open IPTV Forum has developed an end-to-end solution to allow any OIPF terminal to access enriched and personalized IPTV services either in a managed or a non-managed network[OIPF-Overview]. Some OIPF services (such as Network PVR) may be hosted in a CDN.

To that end, the Open IPTV Forum specification is made of 5 parts:

- Media Formats including HTTP Adaptive Streaming
- Content Metadata
- Protocols
- Terminal (Declarative or Procedural Application Environment)
- Authentication, Content Protection and Service Protection

6.1.10. TV-Anytime Forum

Version 1 of the TV-Anytime Forum specifications were published as ETSI TS 102 822-1 through ETSI TS 102 822-7 "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime")". It includes the specification of content metadata in XML schemas (ETSI TS 102 822-3) which define technical parameters for the description of CoD and Live contents. The specification is referenced by DVB and OIPF.

The TV-anytime Forum was closed in 2005.

6.1.11. SNIA

The Storage Networking Industry Association (SNIA) is an association of producers and consumers of storage networking products whose goal is to further storage networking technology and applications.

SNIA has published the Cloud Data Management Interface (CDMI)

standard ([SNIA-CDMI]).

"The Cloud Data Management Interface defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface."

6.2. Related Research Projects

6.2.1. IRTF P2P Research Group

Some information on CDN interconnection motivations and technical issues were presented in the P2P RG at IETF 77. The presentation can be found in [P2PRG-CDNI].

6.2.2. OCEAN

OCEAN (http://www.ict-ocean.eu/) is an EU funded research project that started in February 2010 for 3 years. Some of its objectives are relevant to CDNI. It aims, among other things, at designing a new architectural framework for audiovisual content delivery over the Internet, defining public interfaces between its major building blocks in order to foster multi-vendor solutions and interconnection between Content Networks (the term "Content Networks" corresponds here to the definition introduced in [RFC3466], which encompasses CDNs).

OCEAN has not yet published any open specifications, nor common best practices, defining how to achieve such CDN interconnection.

6.2.3. Eurescom P1955

Eurescom P1955 was a 2010 research project involving a four European Network operators, which studied the interests and feasibility of interconnecting CDNs by firstly elaborating the main service models around CDN interconnection, as well as analyzing an adequate CDN interconnection technical architecture and framework, and finally by providing recommendations for telcos to implement CDN interconnection. The Eurescom P1955 project ended in July 2010.

The authors are not aware of material discussing CDN interconnection protocols made publically available as a deliverable of this project.

6.3. Gap Analysis

A number of standards bodies have produced specifications related to CDNs, namely:

- TISPAN has a dedicated specification for CDN.
- OIPF and ATIS specify the architecture and the protocols of an IPTV solution. Although OIPF and ATIS specifications include the interaction with a CDN, the CDN specifications are coupled with their IPTV specifications.
- <TODO: Add a sentence on ITU>
- IETF CDN WG (now concluded) touched on the same problem space as the present document. However, in accordance with its initial charter, the CDI WG did not define any protocols or interfaces to actually enable CDN Interconnection and at that time (2003) there was not enough industry interest and real life requirements to justify rechartering the WG to conduct the corresponding protocol work.

Although some of the specifications describe multi-CDN cooperation or include reference points for interconnecting CDNs, none of them specify in sufficient detail all the CDNI protocols and CDNI Metadata representations required to enable even a base level of CDN Interconnect functionality to be implemented.

The following sections will summarize the existing work described in Section 6.1 against the CDNI problem space.

6.3.1. Content Acquisition across CDNs and Delivery to End User (Data plane)

A number of standards bodies have completed work in the areas of content acquisition interface between a CSP and a CDN, as well as as on the delivery interface between the surrogate and the User Agent. Some of this work is summarized below.

TISPAN, OIPF and ATIS have specified IPTV and/or CoD services, including the data plane aspects (typically different flavors of RTP/RTCP and HTTP) to obtain content and deliver it to User Agents. For example, :

- The OIPF data plane includes both RTP and HTTP flavors (HTTP progressive download, HTTP Adaptive streaming [3GP-DASH],...).
- ATIS specification "IPTV Content on Demand (CoD) Service" [ATIS-COD] defines a reference point (C2) and the corresponding HTTP-based data plane protocol for content acquisition between an authoritative origin server and the CDN.

While these protocols have not been explicitly specified for content acquisition across CDNs, they are suitable (in addition to others

such as standard HTTP) for content acquisition between CDNs in a CDN Interconnect environment. Therefore for the purpose of a CDNI WG there are already multiple existing data plane protocols that can be used for content acquisition across CDNs.

Similarly, there are multiple existing standards (e.g. OIPTF data plane mentioned above, HTTP adaptive streaming [3GP-DASH]) or public specifications (e.g. vendor specific HTTP Adaptive streaming specification) so that content delivery is considered already solved (or at least sufficiently addressed in other forums).

Thus, specification of the content acquisition interface between CDNs and the delivery interface between the surrogate and the User Agent are out of scope for CDNI. CDNI may only concern itself with the negotiation/selection aspects of the acquisition protocol to be used in a CDN interonnect scenario.

6.3.2. CDNI Metadata

CableLabs, ITU, OIPF and TV-Anytime have work items dedicated to the specification of content metadata:

- CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project. "The VOD Metadata project is a cable television industry and cross-industry-wide effort to specify the metadata and interfaces for distribution of video-on-demand (VOD) material from multiple content providers to cable operators." [CableLabs-Metadata]. However, while the CableLabs work specifies an interface between a content provider and a service provider running a CDN, it does not include an interface that could be used between CDNs.
- ITU Study Group 16 has started work on a number of draft Recommendations (H.IPTV-CPMD, H.IPTV-CPMD, HSTP.IPTV-CMA, HSTP.IPTV-UMCI) specifying metadata for content distribution in IPTV services.
- An Open IPTV Terminal receives the technical description of the content distribution from the OIPF IPTV platform before receiving any content. The Content distribution metadata is sent in the format of a TV-Anytime XSD including tags to describes the location and program type (on demand or Live) as well as describing the time availability of the on demand and live content.

However the specifications outlined above do not include metadata specific to the distribution of content within a CDN or between interconnected CDNs, for example geo-blocking information, availability windows, access control mechanisms to be enforced by the surrogate, how to map an incoming content request to a file on the

origin server or acquire it from the upstream CDN etc.

The CDMI standard ([SNIA-CDMI]) from SNIA defines metadata that can be associated with data that is stored by a cloud storage provider. While the metadata currently defined do not match the need of a CDN Interconnect solution, it is worth considering CDMI as one of the existing pieces of work that may potentially be leveraged for the CDNI Metadata protocol (e.g by extending the CDMI metadata to address more specific CDNI needs).

7. Relationship to relevant IETF Working Groups

7.1. ALTO

As stated in the ALTO Working Group charter [ALTO-Charter]:

"The Working Group will design and specify an Application-Layer Traffic Optimization (ALTO) service that will provide applications with information to perform better-than-random initial peer selection. ALTO services may take different approaches at balancing factors such as maximum bandwidth, minimum cross-domain traffic, lowest cost to the user, etc. The WG will consider the needs of BitTorrent, tracker-less P2P, and other applications, such as content delivery networks (CDN) and mirror selection."

In particular, the ALTO service can be used by a CDN Request Routing system to improve its selection of a CDN surrogate to serve a particular User Agent request (or to serve a request from another surrogate). See [I-D.penno-alto-cdn] for a detailed discussion on how CDN Request Routing can be used as an integration point of ALTO into CDNs. It is possible that the ALTO service could be used in the same manner in a multi-CDN environment based on CDN Interconnect. For example, an upstream CDN may take advantage of the ALTO service in its decision for selecting a downstream CDN to which a user request should be delegated.

However, the work of ALTO is complementary to and does not overlap with the work proposed in this document because the integration between ALTO and a CDN would fall under "algorithms for selection of CDN or Surrogate by Request-Routing systems" in Section 3.2 and is therefore out of scope for a CDNI WG. One area for further study is whether additional information should be provided by an ALTO service to facilitate CDNI CDN selection.

7.2. DECADE

The DECADE Working Group [DECADE-Charter] is addressing the problem of reducing traffic on the last-mile uplink, as well as backbone and transit links caused by P2P streaming and file sharing applications. It addresses the problem by enabling an application endpoint to make content available from an in-network storage service and by enabling other application endpoints to retrieve the content from there.

Exchanging data through the in-network storage service in this manner, instead of through direct communication, provides significant gain where:

- The network capacity/bandwidth from in-network storage service to application endpoint significantly exceeds the capacity/bandwidth from application endpoint to application endpoint (e.g. because of an end-user uplink bottleneck); and
- Where the content is to be accessed by multiple instances of application endpoints (e.g. as is typically the case for P2P applications).

While, as is the case for any other data distribution application, the DECADE architecture and mechanisms could potentially be used for exchange of CDNI control plane information via an in-network-storage service (as opposed to directly between the entities terminating the CDNI protocols in the neighbor CDNs), we observe that:

- CDNI would operate as a "Content Distribution Application" from the DECADE viewpoint (i.e. would operate on top of DECADE).
- There does not seem to be obvious benefits in integrating the DECADE control plane responsible for signaling information relating to control of the in-network storage service itself, and the CDNI control plane responsible for application-specific CDNI interactions (such as exchange of CDNI metadata, CDNI request redirection, transfer of CDNI logging information).
- There would typically be limited benefits in making use of a DECADE in-network storage service because the CDNI protocols are expected to be terminated by a very small number of CDNI clients (if not one) in each CDN, and the CDNI clients are expected to benefit from high bandwidth/capacity when communicating directly to each other (at least as high as if they were communicating via an in-network storage server).

The DECADE in-network storage architecture and mechanisms may theoretically be used for the acquisition of the content objects themselves between interconnected CDNs. It is not expected that this would have obvious benefits in typical situations where a content object is acquired only once from an Upstream CDN to a Downstream CDN

(and then distributed as needed inside the Downstream CDN). But it might have benefits in some particular situations. Since the acquisition protocol between CDNs is outside the scope of the CDNI work, this question is left for further study.

The DECADE in-network storage architecture and mechanisms may potentially also be used within a given CDN for the distribution of the content objects themselves among surrogates of that CDN. Since the CDNI work does not concern itself with operation within a CDN, this question is left for further study.

Therefore, the work of DECADE may be complementary to but does not overlap with the CDNI work proposed in this document.

7.3. PPSP

As stated in the PPSP Working Group charter [PPSP-Charter]:

"The Peer-to-Peer Streaming Protocol (PPSP) working group develops two signaling and control protocols for a peer-to-peer (P2P) streaming system for transmitting live and time-shifted media content with near real-time delivery requirements." and "The PPSP WG designs a protocol for signaling and control between trackers and peers (the PPSP "tracker protocol") and a signaling and control protocol for communication among the peers (the PPSP "peer protocol"). The two protocols enable peers to receive streaming data within the time constraints required by specific content items."

Therefore PPSP is concerned with the distribution of the streamed content itself along with the necessary signaling and control required to distribute the content. As such, it could potentially be used for the acquisition of streamed content across interconnected CDNs. But since the acquisition protocol is outside the scope of the work proposed for CDNI, we leave this for further study. Also, because of its streaming nature, PPSP is not seen as applicable to the distribution and control of the CDNI control plane and CDNI data representations.

Therefore, the work of PPSP may be complementary to but does not overlap with the work proposed in this document for CDNI.

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

Distribution of content by a CDN comes with a range of security considerations such as how to enforce control of access to the content by users in line with the CSP policy. These security aspects are already dealt with by CDN Providers and CSPs today in the context of standalone CDNs. However, interconnection of CDNs introduces a new set of security considerations by extending the trust model (i.e. the CSP "trusts" a CDN that "trusts" another CDN).

Maintaining the security of the content itself, its associated metadata (including distribution and delivery policies) and the CDNs distributing and delivering it, are critical requirements for both CDN Providers and CSPs and any work on CDN Interconnection must provide sufficient mechanisms to maintain the security of the overall system of interconnected CDNs as well as the information (content, metadata, logs, etc) distributed and delivered through any CDN Interconnects.

10. Acknowledgements

The authors would like to thank Andre Beck, Mark Carlson, Bruce Davie, David Ferguson, Yiu Lee, Kevin Ma, Julien Maisonneuve, Guy Meador, Emile Stephan, Oskar van Deventer and Mahesh Viveganandhan for their review comments and contributions to the text.

11. References

11.1. Normative References

[I-D.bertrand-cdni-experiments]
Bertrand, G., Faucheur, F., and L. Peterson, "Content Distribution Network Interconnection (CDNI) Experiments", draft-bertrand-cdni-experiments-00 (work in progress), February 2011.

[I-D.bertrand-cdni-use-cases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., and P. Eardley, "Use Cases for Content Distribution Network Interconnection", draft-bertrand-cdni-use-cases-01 (work in progress), January 2011.

[I-D.bertrand-cdni-use-cases-00]
Bertrand, G. and E. Stephan, "Use Cases for Content Distribution Network Interconnection - draft-bertrand-cdni-use-cases-00 (superseded)",

January 2011.

[I-D.jenkins-cdni-names]
Niven-Jenkins, B., "Thoughts on Naming and Referencing of Data Objects within Content Distribution Network Interconnection (CDNI) solutions", draft-jenkins-cdni-names-00 (work in progress), February 2011.

[I-D.watson-cdni-use-cases]
Watson, G., "CDN Interconnect Use Cases", draft-watson-cdni-use-cases-00 (work in progress), January 2011.

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

11.2. Informative References

[3GP-DASH]
"Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) http://www.3gpp.org/ftp/Specs/html-info/26247.htm".

[ALTO-Charter]
"IETF ALTO WG Charter (http://datatracker.ietf.org/wg/alto/charter/)".

[ATIS] "ATIS (http://www.atis.org/)".

[ATIS-COD]
"ATIS IIF: IPTV Content on Demand Service, January 2011 ht tp://www.atis.org/iif/_Com/Docs/Task_Forces/ARCH/ ATIS-0800042.pdf".

[CDI-Charter]
"IETF CDI WG Charter (http://www.ietf.org/wg/concluded/cdi)".

[CableLabs]
"CableLabs (http://www.cablelabs.com/about/)".

[CableLabs-Metadata]
"CableLabs VoD Metadata Project Primer (http://www.cablelabs.com/projects/metadata/primer/)".

[DECADE-Charter]

"IETF DECADE WG Charter
(http://datatracker.ietf.org/wg/decade/charter/)".

[I-D.penno-alto-cdn]
Penno, R., Raghunath, S., Medved, J., Alimi, R., Yang, R., and S. Previdi, "ALTO and Content Delivery Networks", draft-penno-alto-cdn-02 (work in progress), October 2010.

[MPEG-DASH]
"Information technology - MPEG systems technologies - Part 6: Dynamic adaptive streaming over HTTP (DASH), (DIS version), February 2011
http://mpeg.chiariglione.org/working_documents.htm#MPEG-B".

[OIPF-Overview]
"OIPF Release 2 Specification Volume 1 - Overview", September 2010.

[P2PRG-CDNI]
Davie, B. and F. Le Faucheur, "Interconnecting CDNs aka "Peering Peer-to-Peer" (http://www.ietf.org/proceedings/77/slides/P2PRG-2.pdf)", March 2010.

[PPSP-Charter]
"IETF PPSP WG Charter
(http://datatracker.ietf.org/wg/ppsp/charter/)".

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3040] Cooper, I., Melve, I., and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", RFC 3040, January 2001.

[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.

[RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence

Protocol (XMPP): Core", RFC 3920, October 2004.

[RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.

[SNIA-CDMI] "SNIA CDMI (http://www.snia.org/tech_activities/standards/curr_standards/cdmi)".

[TAXONOMY] Pathan, A., "A Taxonomy and Survey of Content Delivery Networks (http://www.gridbus.org/reports/CDN-Taxonomy.pdf)", 2007.

[Y.1910] "ITU-T Recomendation Y.1910 "IPTV functional architecture"", September 2008.

[Y.2019] "ITU-T Recomendation Y.2019 "Content delivery functional architecture in NGN"", September 2010.

Authors' Addresses

Ben Niven-Jenkins
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: ben@velocix.com

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

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

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
USA

Email: nabil.bitar@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 10, 2012

K. Leung
Cisco
Y. Lee
Comcast
F. Le Faucheur
M. Viveganandhan
Cisco
G. Watson
BT

July 9, 2011

Content Distribution Network Interconnection (CDNI) Requirements
draft-lefaucheur-cdni-requirements-02

Abstract

Content Delivery Networks (CDNs) are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. There is a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to end users. The Content Distribution Network Interconnection (CDNI) working group has been chartered to develop an interoperable and scalable solution for such CDN interconnection.

The goal of the present document is to outline the requirements for the solution and interfaces to be specified by the CDNI working group.

Requirements Language

The key words "Must", "Should" and "May" in this document are to be interpreted in the following way:

- "Must" indicates requirements that are to be supported by the CDNI protocols in the stated scope (aka "within initial CDNI scope" or "beyond initial scope"). A requirement is stated as a "Must" when it is established by that it can be met without compromising the targeted schedule for WG deliverables, or when it is established that specifying a solution without meeting this requirement would not make sense and would justify re-adjusting the WG schedule, or both.

- "Should" indicates requirements that are to be supported by the CDNI protocols in the stated scope (aka "within initial CDNI scope" or "beyond initial scope") unless the WG realizes at a

later stage that attempting to meet this requirement would compromise the overall WG schedule (for example it would involve complexities that would result in significantly delaying the deliverables).

- o "May" indicates requirements that are to be supported by the CDNI protocols in the stated scope (aka "within initial CDNI scope" or "beyond initial scope") provided that dedicating WG resources to this work does not prevent addressing "Should" and "Must" requirements and that attempting to meet this requirement would not compromise the overall WG schedule.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for end users, and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. However, the footprint of a given CDN in charge of delivering a given content may not expand close enough to the End User's current location or attachment network to realize the cost benefit and user experience that a more distributed CDN would provide. This creates a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

[I-D.jenkins-cdni-problem-statement] outlines the problem area that the CDNI working group is chartered to address. [I-D.bertrand-cdni-use-cases] discusses the use cases for CDN Interconnection. [I-D.davie-cdni-framework] discusses the technology framework for the CDNI solution and interfaces.

The goal of the present document is to document the requirements for the CDNI solution and interfaces. In accordance with the working group charter, the work is prioritized in a "walk before you run" approach: the present document separates the CDNI requirements into a set of more urgent requirements that are within the initial scope of the CDNI working group, and a set of less urgent additional requirements that are left to potential future rechartering of the working group.

1.1. Terminology

This document uses the terminology defined in section 1.1 of [I-D.jenkins-cdni-problem-statement].

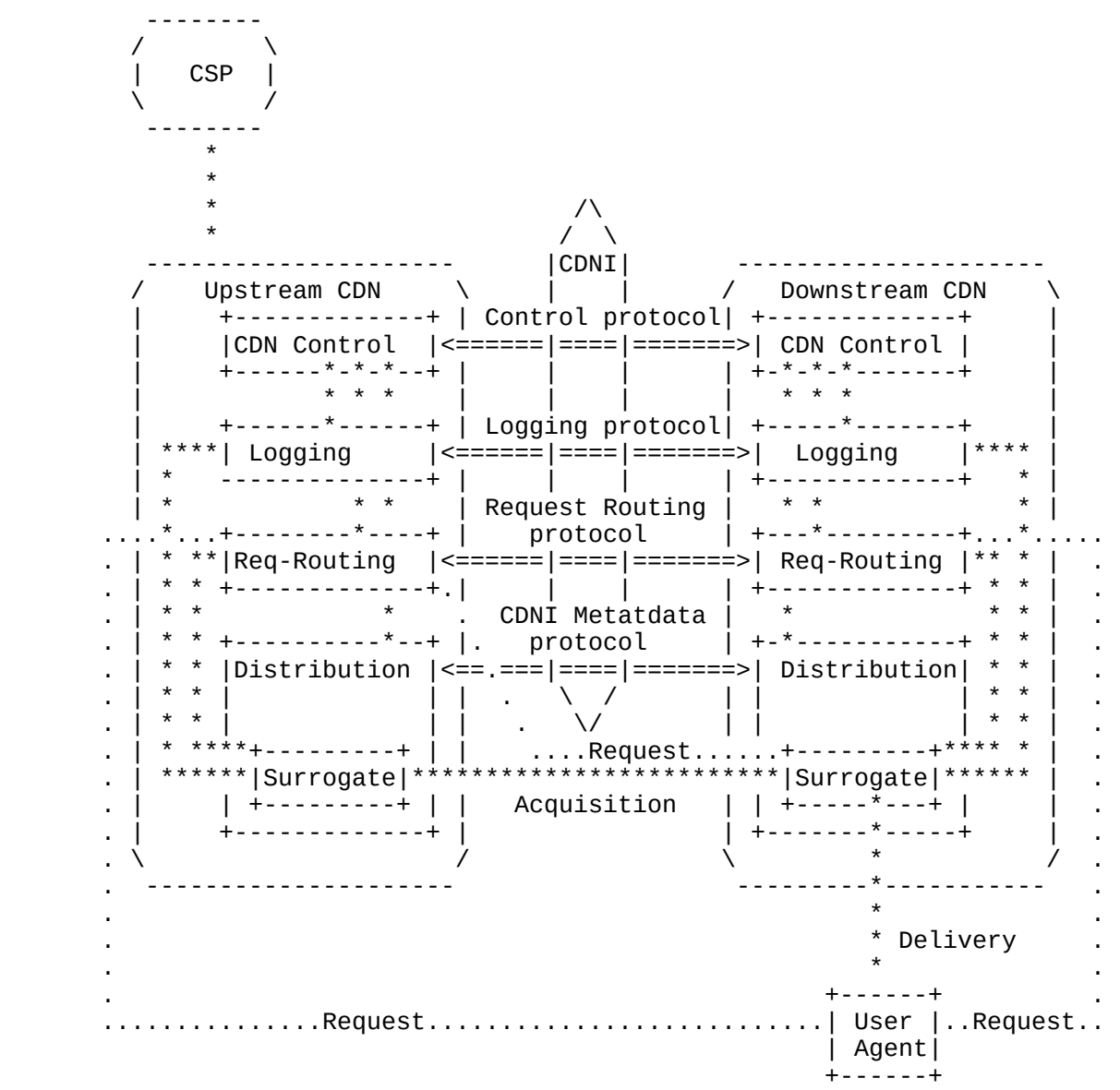
This also defined the following additional terms [Editor's Note: these definitions may be better located in another document such as the Problem Statement]:

- Recursive CDNI request routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can query the Downstream CDN Request Routing system via the CDNI Request Routing protocol (or use information cached from earlier similar queries) to find out how the Downstream CDN wants the request to be redirected, which allows the Upstream CDN to factor in the Downstream CDN response when redirecting the user agent. This approach is referred to as "recursive" CDNI request routing. Note that the Downstream CDN may elect to have the request redirected directly to a Surrogate inside the Downstream CDN, to the Request-Routing System of the Downstream CDN, to another CDN, or to any other system that the Downstream CDN sees as fit for handling the redirected request.

- Iterative CDNI Request Routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can base its redirection purely on a local decision (and without attempting to take into account how the Downstream CDN may in turn redirect the user agent). In that case, the Upstream CDN redirects the request to the request routing system in the Downstream CDN, which in turn will decide how to redirect that request: this approach is referred to as "iterative" CDNI request routing.

2. CDNI Model and CDNI protocols

For convenience Figure 1 from [I-D.jenkins-cdni-problem-statement] illustrating the CDNI problem area and the CDNI protocols is replicated below.



<==> interfaces inside the scope of CDNI

**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Model and CDNI APIs

3. Generic Requirements

This section identifies generic requirements independent of the individual CDNI protocols. Some of those are expected to affect multiple or all protocols.

3.1. Within Initial CDNI Scope

R1 Wherever possible, the CDNI protocols Should reuse or leverage existing IETF protocols.

R2 The CDNI solution Must not require a change, or an upgrade, to the User Agent to benefit from content delivery through interconnected CDNs.

R3 The CDNI solution Must not require intra-CDN information to be exposed to other CDNs for effective and efficient delivery of the content. Examples of intra-CDN information include surrogate topology, surrogate status, cached content, etc.

R4 The CDNI solution Must support delivery to the user agent based on HTTP [RFC2616]. [Note that while delivery and acquisition "data plane" protocols are out of the CDNI solution scope, the CDNI solution "control plane" protocols are expected to participate in enabling, selecting or facilitating operations of such acquisition and delivery protocols. Hence it is useful to state requirements on the CDNI solution in terms of which acquisition and delivery protocols].

R5 The CDNI solution Must support acquisition across CDNs based on HTTP [RFC2616].

R6 The CDNI solution May support delivery to the user agent based on protocols other than HTTP.

R7 The CDNI solution May support acquisition across CDNs based on protocols other than HTTP.

R8 The CDNI solution Should support cascaded CDN redirection (CDN1 redirects to CDN2 that redirects to CDN3) to an arbitrary number of levels.

R9 The CDNI solution Should support an arbitrary topology of interconnected CDNs (i.e. the CDN topology cannot be restricted to a tree, a loop-free topology, etc.).

R10 The CDNI solution Must prevent looping of any CDNI information exchange.

R11 When making use of third party reference, the CDNI solution Must consider the potential issues associated with the use of various format of third-party references (e.g. NAT or IPv4/IPv6 translation potentially breaking third-party references based on an IP addresses such as URI containing IPv4 or IPv6 address litterals, split DNS situations potentially breaking third-party references based on DNS fully qualified domain names) and wherever possible avoid, minimize or mitigate the associated risks based on the specifics of the environments where the reference is used (e.g. likely or unlikely presence of NAT in the path). In particular, this applies to situations where the CDNI solution needs to construct and convey uniform resource identifiers for directing/redirecting a content request, as well as to situations where the CDNI solution needs to pass on a third party reference (e.g. to identify a User Agent) in order to allow another entity to make a more informed decision (e.g. make a more informed request routing decision by attempting to derive location information from the third party reference).

3.2. Beyond Initial CDNI Scope

R12 The CDNI solution Must support cascaded CDN redirection (CDN1 redirects to CDN2 that redirects to CDN3) to an arbitrary number of levels. [Note: this "Must" requirement appeared as a "Should" requirement in Section 3.1]

R13 The CDNI solution Must support an arbitrary topology of interconnected CDNs (i.e. the CDN topology cannot be restricted to a tree, a loop-free topology, etc.). [Note: this "Must" requirement appeared as a "Should" requirement in Section 3.1]

R14 The CDNI solution Should support virtualization of the Downstream CDN, so that the Downstream CDN can appear as multiple logical Downstream CDNs.

4. CDNI Control Protocol Requirements

The primary purpose of the CDNI Control protocol is to initiate the interconnection across CDNs, bootstrap the other CDNI interfaces and trigger actions into the Downstream CDN by the Upstream CDN (such as delete object from caches or trigger pre-positioned content acquisition). We observe that while the CDNI Control protocol is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized

over a single interface and protocol, or over multiple interfaces and protocols.

4.1. Within Initial CDNI Scope

R15 The CDNI Control protocol Must allow the Upstream CDN to request that the Downstream CDN (and, if cascaded CDNs are supported by the solution, that the potential cascaded Downstream CDNs) perform the following actions on an object or object set:

* Mark an object(s) and/or its CDNI metadata as "stale" and revalidate them before they are delivered again

* Delete an object(s) and/or its CDNI metadata from the CDN surrogates and any storage.

R16 The CDNI Control protocol Must allow the downstream CDN to report on the completion of these actions (by itself, and if cascaded CDNs are supported by the solution, by potential cascaded Downstream CDNs), in a manner appropriate for the action (e.g. synchronously or asynchronously).

R17 The CDNI Control protocol Must support initiation and control by the Upstream CDN of pre-positioned CDNI metadata acquisition by the Downstream CDN.

R18 The CDNI Control protocol Should support initiation and control by the Upstream CDN of pre-positioned content acquisition by the Downstream CDN.[Editor’s Note: how much influence the Upstream CDN ought to have on pre-positioning of the content on surrogates inside the Downstream CDN is TBD].

4.2. Beyond Initial CDNI Scope

R19 The CDNI Control protocol Must support support initiation and control by the Upstream CDN of pre-positioned content acquisition.[Editor’s Note: how much influence the Upstream CDN ought to have on pre-positioning of the content on surrogates inside the Downstream CDN is TBD]. [Note: this "Must" requirement appeared as a "Should" requirement in Section 4.1]

R20 The CDNI Control protocol Must allow a CDN to establish, update and terminate a CDN interconnection with another CDN whereby one CDN can act as a Downstream CDN for the other CDN (that acts as an Upstream CDN).

R21 The CDNI Control protocol Must allow control of the CDNI interconnection between any two CDNs independently for each direction (i.e. For the direction where CDN1 is the Upstream CDN and CDN2 is the Downstream CDN, and for the direction where CDN2 is the Upstream CDN and CDN1 is the Downstream CDN).

R22 The CDNI Control protocol Should allow bootstrapping of the Request-Routing protocol. For example, this can potentially include:

* negotiation of the Request-Routing method (e.g. DNS vs HTTP, if more than one method is specified)

* discovery of the Request-Routing protocol endpoints

* information necessary to establish secure communication between the Request-Routing protocol endpoints.

R23 The CDNI Control protocol Should allow bootstrapping of the Metadata Signaling protocol. This information could, for example, include:

* discovery of the Metadata Signaling protocol endpoints

* information necessary to establish secure communication between the Metadata Signaling protocol endpoints.

R24 The CDNI Control protocol Should allow bootstrapping of the Content Acquisition protocol. This could, for example, include exchange and negotiation of the Content Acquisition protocols to be used across the CDNs (e.g. HTTP, HTTPS, FTP, ATIS C2).

R25 The CDNI Control protocol Should allow exchange and negotiation of delivery authorization mechanisms to be supported across the CDNs (e.g. URI signature based validation).

R26 The CDNI Control protocol Should allow bootstrapping of the CDNI Logging protocol. This information could, for example, include:

* discovery of the Logging protocol endpoints

* information necessary to establish secure communication between the Logging protocol endpoints

* negotiation/definition of the log file format and set of fields to be exported through the Logging protocol, with some granularity (e.g. On a per content type basis).

* negotiation/definition of parameters related to transaction Logs export (e.g., export protocol, file compression, export frequency, directory).

5. CDNI Request Routing Protocol Requirements

5.1. Within Initial CDNI Scope

The main function of the Request Routing protocol is to allow the Request-Routing systems in interconnected CDNs to communicate to facilitate redirection of the request across CDNs.

R27 The CDNI Control protocol Must allow the Downstream CDN to communicate to the Upstream CDN coarse information about the Downstream CDN ability and/or willingness to handle requests from the Upstream CDN. For example, this could potentially include a binary signal ("Downstream CDN ready/not-ready to take additional requests from Upstream CDN") to be used in case of excessive load or failure condition in the Downstream CDN.

R28 The CDNI Request-Routing protocol Should allow the Downstream CDN to communicate to the Upstream CDN aggregate information to facilitate CDN selection during request routing, such as Downstream CDN capabilities, resources and affinities (i.e. Preferences or cost). This information could, for example, include:

* supported content types and delivery protocols

* footprint (e.g. layer-3 coverage)

* a set of metrics/attributes (e.g. Streaming bandwidth, storage resources, distribution and delivery priority)

* a set of affinities (e.g. Preferences, indication of distribution/delivery fees)

* information to facilitate request redirection (e.g. Reachability information of Downstream CDN Request Routing system).

[Note: Some of this information - such as supported content types and delivery protocols- may also potentially be taken into account by the distribution system in the Upstream CDN for pre-positioning of content and/or metadata in the Downstream CDN in case of pre-positioned content acquisition and/or pre-positioned CDNI metadata acquisition.]

R29 If cascaded redirection is supported by the CDNI solution, the CDNI Request-Routing protocol Must allow the Downstream CDN to also include in the information communicated to the Upstream CDN, information on the capabilities, resources and affinities of CDNs to which the Downstream CDN may (in turn) redirect requests received by the Upstream CDN. In that case, the CDNI Request-Routing protocol Must prevent looping of such information exchange.

R30 The CDNI Control protocol May allow the Downstream CDN to communicate to the Upstream CDN aggregate information on CDNI administrative limits and policy. This information can be taken into account by the Upstream CDN Request Routing system in its CDN Selection decisions. This information could, for example, include:

* maximum number of requests redirected by the Upstream CDN to be served simultaneously by the Downstream CDN

* maximum aggregate volume of content (e.g. in Terabytes) to be delivered by the Downstream CDN over a time period.

R31 The CDNI Request-Routing architecture and protocol Must support efficient request-routing for small objects. This may, for example, call for a mode of operation (e.g. DNS-based request routing) where freshness and accuracy of CDN/Surrogate selection can be traded-off against reduced request-routing load (e.g. Via lighter-weight queries and caching of request-routing decisions).

R32 The CDNI Request-Routing architecture and protocol Must support efficient request-routing for large objects. This may, for example, call for a mode of operation (e.g. HTTP-based request routing) where freshness and accuracy of CDN/Surrogate selection justifies a per-request decision and a per-request CDNI Request-Routing protocol call.

R33 The CDNI Request-Routing architecture Must support recursive CDNI request routing.

R34 The CDNI Request-Routing architecture Must support iterative CDNI request routing.

R35 In case of detection of a request redirection loop, the CDNI Request-Routing loop prevention mechanism Should allow routing of the request (as opposed to the request loop being simply interrupted without routing the request).

R36 The CDNI Request-Routing protocol Should support an optional mechanism allowing enforcment of a limit on the number of successive CDN redirections for a given request.

R37 The CDNI Request-Routing protocol May support an optional mechanism allowing an upstream CDN to avoid redirecting a request to a downstream CDN if that is likely to result in the total redirection time exceeding some limit.

R38 The CDNI Request-Routing protocol Must allow the Upstream CDN to include, in the query to the Downstream CDN, the necessary information to allow the Downstream CDN to process the redirection query. This could, for example, include:

* information from which the location of the user-agent that originated the request can be inferred (e.g. User Agent fully qualified domain name in case of HTTP-based Request Routing, DNS Proxy fully qualified domain name in case of DNS-based Request Routing)

* requested resource information (e.g. Resource URI in case of HTTP-based Request Routing, Resource hostname in case of DNS-based Request Routing)

* additional available request information (e.g. request headers in case of HTTP-based Request Routing).

R39 The CDNI Request-Routing protocol May also allow the Upstream CDN to convey information pointing to CDNI metadata applicable (individually or through inheritance) to the requested content. For illustration, the CDNI metadata pointed to could potentially include metadata that is applicable to any content, metadata that is applicable to a content collection (to which the requested content belongs) and/or metadata that is applicable individually to the requested content.

R40 The CDNI Request-Routing protocol Must allow the Downstream CDN to include the following information in the response to the Upstream CDN:

* status code, in particular indicating acceptance or rejection of request (e.g. Because the Downstream CDN is unwilling or unable to serve the request). In case of rejection, an error code is also to be provided, which allows the Upstream CDN to react appropriately (e.g. Select another Downstream CDN, or serve the request itself)

* redirection information (e.g. Resource URI in case of HTTP-based Request Routing, equivalent of a DNS record in case of DNS-based Request Routing).

5.2. Beyond Initial CDNI Scope

R41 The CDNI Request-Routing protocol Must allow the Downstream CDN to communicate to the Upstream CDN aggregate information to facilitate CDN selection during request routing, such as Downstream CDN capabilities, resources and affinities (i.e. Preferences or cost). This information could, for example, include:

* supported content types and delivery protocols

* footprint (e.g. layer-3 coverage)

* a set of metrics/attributes (e.g. Streaming bandwidth, storage resources, distribution and delivery priority)

* a set of affinities (e.g. Preferences, indication of distribution/delivery fees)

* information to facilitate request redirection (e.g. Reachability information of Downstream CDN Request Routing system).

[Note: this "Must" requirement appeared as a "Should" requirement in Section 5.1]

R42 The CDNI Request-Routing protocol Must allow the Downstream CDN to also include in the information communicated to the Upstream CDN, information on the capabilities, resources and affinities of CDNs to which the Downstream CDN may (in turn) redirect requests received by the Upstream CDN. The CDNI Control protocol Must prevent looping of such information exchange. [Note: this "Must" requirement appeared as a conditional "Must" requirement in Section 5.1]

R43 The CDNI Request-Routing protocol Should allow the Downstream CDN to communicate to the Upstream CDN aggregate information on CDNI administrative limits and policy. This information can be taken into account by the Upstream CDN Request Routing system in its CDN Selection decisions. This information could, for example, include:

* maximum number of requests redirected by the Upstream CDN that to be served simultaneously by the Downstream CDN

* maximum aggregate volume of content (e.g. in Terabytes) to be delivered by the Downstream CDN over a time period

[Note: this "Should" requirement appeared as a "May" requirement in Section 5.1]

R44 The CDNI Request-Routing loop prevention mechanism Must allow routing of the request (as opposed to the request loop being simply interrupted without routing the request). [Note: this "Must" requirement appeared as a "Should" requirement in Section 5.1]

R45 The CDNI Request-Routing protocol Must support optional enforcement of a limit on the number of successive CDN redirections for a given request. [Note: this "Must" requirement appeared as a "Should" requirement in Section 5.1]

6. CDNI Metadata Distribution Protocol Requirements

The primary function of the CDNI Metadata Distribution protocol is to allow the Distribution system in interconnected CDNs to communicate to ensure Content Distribution Metadata with inter-CDN scope can be exchanged across CDNs. We observe that while the CDNI Metadata Distribution protocol is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols. For example, a subset of the CDNI metadata might be conveyed in-band along with the actual content acquisition across CDNs (e.g. content MD5 in HTTP header) while another subset might require an out-of-band interface & protocol (e.g. geo-blocking information).

6.1. Within Initial CDNI Scope

R46 The CDNI Metadata Distribution protocol Must allow the Upstream CDN to provide the Downstream CDN with content distribution metadata of inter-CDN scope.

R47 The CDNI Metadata Distribution protocol Must support exchange of CDNI metadata for both the dynamic content acquisition model and the pre-positioning content acquisition model.

R48 The CDNI Metadata Distribution protocol Must/Should/May? support a mode where no, or a subset of, the Metadata is initially communicated to the Downstream CDN along with information about how/where to acquire the rest of the CDNI Metadata (i.e. Dynamic CDNI metadata acquisition).

R49 The CDNI Metadata Distribution protocol Must/Should/May? support a mode where all the relevant Metadata is initially communicated to the Downstream CDN (i.e. Pre-positioned CDNI metadata acquisition).

R50 Whether in the pre-positioned content acquisition model or in the dynamic content acquisition model, the CDNI Metadata Distribution protocol Must provide the necessary information to allow the Downstream CDN to acquire the content from an upstream source (e.g. Acquisition protocol and Uniform Resource Identifier in Upstream CDN- or rules to construct this URI).

R51 The CDNI metadata Must allow signaling of one or more upstream sources, where each upstream source can be in the Upstream CDN, in another CDN, the CSP origin server or any arbitrary source designated by the Upstream CDN. Note that some upstream sources (e.g. the content origin server) may or may not be willing to serve the content to the Downstream CDN, if this policy is known to the upstream CDN then it may omit those sources when exchanging CDNI metadata.

R52 The CDNI Metadata Distribution protocol Must allow the Upstream CDN to request addition and modification of CDNI Metadata into the Downstream CDN.

R53 The CDNI Metadata Distribution protocol Must allow removal of obsolete CDNI Metadata from the Downstream CDN (this could, for example, be achieved via an explicit removal request from the Upstream CDN or via expiration of a Time-To-Live associated to the Metadata).

R54 The CDNI Metadata Distribution protocol Must allow association of CDNI Metadata at the granularity of individual object. This is necessary to achieve fine-grain Metadata distribution at the level of an individual object when necessary.

R55 The CDNI Metadata Distribution protocol Must allow association of CDNI Metadata at the granularity of an object set. This is necessary to achieve scalable distribution of metadata when a large number of objects share the same distribution policy.

R56 The CDNI Metadata Distribution protocol Must support multiple levels of inheritance with precedence to more specific metadata. For example, the CDNI Metadata Distribution protocol may support metadata that is applicable to any content, metadata that is applicable to a content collection and metadata that is applicable to an individual content where content level metadata overrides content collection metadata that overrides metadata

for any content.

R57 The CDNI Metadata Distribution protocol Must ensure that conflicting metadata with overlapping scope are prevented or deterministically handled.

R58 The CDNI Metadata Distribution protocol Must provide indication by the Downstream CDN to the Upstream CDN of whether the CDNI metadata (and corresponding future request redirections) is accepted or rejected. When rejected, the CDNI Metadata Distribution protocol Must allow the Downstream CDN to provide information about the cause of the rejection.

R59 The CDNI Metadata Distribution protocol Must allow signaling of content distribution control policies. For example, this could potentially include:

* geo-blocking information (i.e. Information defining geographical areas where the content is to be made available or blocked)

* availability windows (i.e. Information defining time windows during which the content is to be made available or blocked)

* delegation whitelist/blacklist (i.e. Information defining which downstream CDNs the content may/may not be delivered through)

R60 The CDNI Metadata Distribution protocol Must allow signaling of authorization checks and validation that are to be performed by the surrogate before delivery. For example, this could potentially include:

* need to validate URI signed information (e.g. Expiry time, Client IP address).

6.2. Beyond Initial CDNI Scope

R61 The CDNI Metadata Distribution protocol Must support a mode where no, or a subset of, the Metadata is initially communicated to the Downstream CDN along with information about how/where to acquire the rest of the CDNI Metadata (i.e. Dynamic CDNI metadata acquisition). [Note: this "Must" requirement appeared as a "Must/Should/May?" requirement in Section 6.1]

R62 The CDNI Metadata Distribution protocol Must support a mode where all the relevant Metadata is initially communicated to the Downstream CDN (i.e.Pre-positioned CDNI metadata acquisition). [Note: this "Must" requirement appeared as a "Must/Should/May?" requirement in Section 6.1]

R63 The CDNI Metadata Distribution protocol Must allow signaling of CDNI-relevant surrogate cache behavior parameters. For example, this could potentially include:

* control of whether the query string of HTTP URI is to be ignored by surrogate cache

* content revalidation parameters (e.g. TTL)

7. CDNI Logging Protocol Requirements

This section identifies the requirements related to the CDNI Logging protocol. We observe that while the CDNI Logging protocol is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols.

7.1. Within Initial CDNI Scope

R64 The CDNI logging architecture and protocol Must ensure reliable logging of CDNI events.

R65 The CDNI Logging protocol Must provide logging of deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN.

R66 If cascaded CDNs are supported, the CDNI logging protocol Must allow the Downstream CDN to report to the Upstream CDN logging for deliveries performed by the Downstream CDN itself as well as logging for deliveries performed by cascaded CDNs on behalf of the Downstream CDN.

R67 The CDNI Logging protocol Must provide logging of distribution performed by the Upstream CDN as a result of acquisition request by the Downstream CDN.

R68 The CDNI Logging protocol Must support batch/offline exchange of logging records.

R69 The CDNI Logging protocol Should also support additional timing constraints for some types of logging records (e.g. near-real time for monitoring and analytics applications)

R70 The CDNI Logging protocol Must define a log file format and a set of fields to be exported through the Logging protocol, with some granularity (e.g. On a per content type basis).

R71 The CDNI Logging protocol Must define a transport mechanisms to exchange CDNI Logging files.

[Editor's note: should we add a requirement for support of aggregate/ summarized logs (e.g. total bytes delivered for a content regardless of individual USer Agents to which it was delivered)]

7.2. Beyond Initial CDNI Scope

R72 The CDNI logging protocol Must allow the Downstream CDN to report to the Upstream CDN logging for deliveries performed by the Downstream CDN itself as well as logging for deliveries performed by cascaded CDNs on behalf of the Downstream CDN. [Note: this "Must" requirement appeared as a conditional "Must" requirement in Section 7.1]

R73 The CDNI Logging protocol Must support real-time exchange of some types of logging records (e.g. For real-time monitoring of deliveries across CDNs). [Note: this "Must" requirement appeared as a "Should" requirement in Section 7.1]

R74 The CDNI Logging protocol Must allow a CDN to query another CDN for relevant current logging records (e.g. For on-demand access to real-time logging information).

8. CDNI Security Requirements

This section identifies the requirements related to the CDNI security. Some of those are expected to affect multiple or all protocols.

8.1. Within Initial CDNI Scope

R75 All the CDNI protocols Must support secure operation over unsecured IP connectivity (e.g. The Internet). This includes authentication, confidentiality, integrity protection as well as protection against spoofing and replay.

R76 The CDNI solution Must provide sufficient protection against Denial of Service attacks. This includes protection against spoofed delivery requests sent by user agents directly to a Downstream CDN attempting to appear as if they had been redirected by a given Upstream CDN when they have not.

R77 The CDNI solution Should be able to ensure that for any given request redirected to a Downstream CDN, the chain of CDN Delegation (leading to that request being served by that CDN) can be established with non-repudiation.

R78 The CDNI solution Should be able to ensure that the Downstream CDN cannot spoof a transaction log attempting to appear as if it corresponds to a request redirected by a given Upstream CDN when that request has not been redirected by this Upstream CDN. This ensures non-repudiation by the Upstream CDN of transaction logs generated by the Downstream CDN for deliveries performed by the Downstream CDN on behalf of the Upstream CDN.

R79 The CDNI solution May provide a mechanism allowing an Upstream CDN that has credentials to acquire content from the CSP origin server (or another CDN), to allow establishment of credentials authorizing the Downstream CDN to acquire the content from the CSP origin server (or the other CDN) (e.g. In case the content cannot be acquired from the Upstream CDN).

8.2. Beyond Initial CDNI Scope

R80 The CDNI solution Must provide a mechanism allowing an Upstream CDN that has credentials to acquire content from the CSP origin server (or another CDN), to allow establishment of credentials authorizing the Downstream CDN to acquire the content from the CSP origin server (or the other CDN) (e.g. In case the content cannot be acquired from the Upstream CDN). [Note: this "Must" requirement appeared as a "May" requirement in Section 8.1]

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

This document discusses CDNI security requirements in Section 8.

11. Acknowledgements

This document leverages the earlier work of the IETF CDI working group in particular as documented in [I-D.cain-request-routing-req], [I-D.amini-cdi-distribution-reqs] and [I-D.gilletti-cdnp-aaa-reqs].

The authors would like to thank Gilles Bertrand, Christophe Caillet, Bruce Davie, Phil Eardly, Agustin Schapira and Emile Stephan for their input. We also want to thank Ben Niven-Jenkins for his review and comments.

12. References

12.1. Normative 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.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (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.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

12.2. Informative References

[I-D.amini-cdi-distribution-reqs]
Amini, L., "Distribution Requirements for Content Internetworking", draft-amini-cdi-distribution-reqs-02 (work in progress), November 2001.

[I-D.cain-request-routing-req]
Cain, B., "Request Routing Requirements for Content Internetworking", draft-cain-request-routing-req-03 (work in progress), November 2001.

[I-D.gilletti-cdnp-aaa-reqs]
"CDI AAA Requirements,
draft-gilletti-cdnp-aaa-reqs-01.txt", June 2001.

Authors' Addresses

Kent Leung
Cisco Systems
3625 Cisco Way
San Jose 95134
USA

Phone: +1 408 526 5030
Email: kleung@cisco.com

Yiu Lee
Comcast

Email: yiu_lee@cable.comcast.com

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

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

Mahesh Viveganandhan
Cisco Systems
375 East Tasman Drive
San Jose 95134
USA

Email: mvittal@cisco.com

Grant Watson
BT

Email: grant.watson@bt.com

CDNI
Internet-Draft
Intended status: Informational
Expires: January 5, 2012

M. Stiemerling
NEC Europe Ltd.

July 4, 2011

Considerations on Request Routing for CDNI
draft-stiemerling-cdni-routing-cons-00

Abstract

Request routing in CDNs and also in the case of interconnecting multiple CDNs requires to match against a set functional requirements but also operational requirements. This memo discusses a few operational requirements and the impact to the current request routing proposals.

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 January 5, 2012.

Copyright Notice

Table of Contents

1. Introduction

Request routing in CDNs and also in the case of interconnecting multiple CDNs requires to match against a set functional requirements (see [I-D.lefaucheur-cdni-requirements]) but also operational requirements. This memo discusses a few operational requirements and the impact to the current request routing proposals [I-D.peterson-cdni-strawman] and [I-D.xiaoyan-cdni-requestrouting]. This draft presents some initial considerations about request routing in CDN interconnect scenarios.

Comments and discussions about this memo should be directed to the CDNI WG: cdni@ietf.org.

2. Speed Matters

The speed of content delivery matters a lot in CDNs for the actual content consumer, but also for the content provider. The content consumer does not like to wait for too long until the content is being displayed on its device. The content provider wants to achieve a fast and reliable content delivery. However, the current request routing proposals suggest that the time from the first content request to the actual start of delivery does not matter much.

For instance, method 1 of [I-D.peterson-cdni-strawman], shown in Figure 1, requires 4 "transactions", as seen from the End-User to start the delivery of content. The total time from the first request to the actual data delivery is: t_start = t_I + t_II + t_III + t_IV.

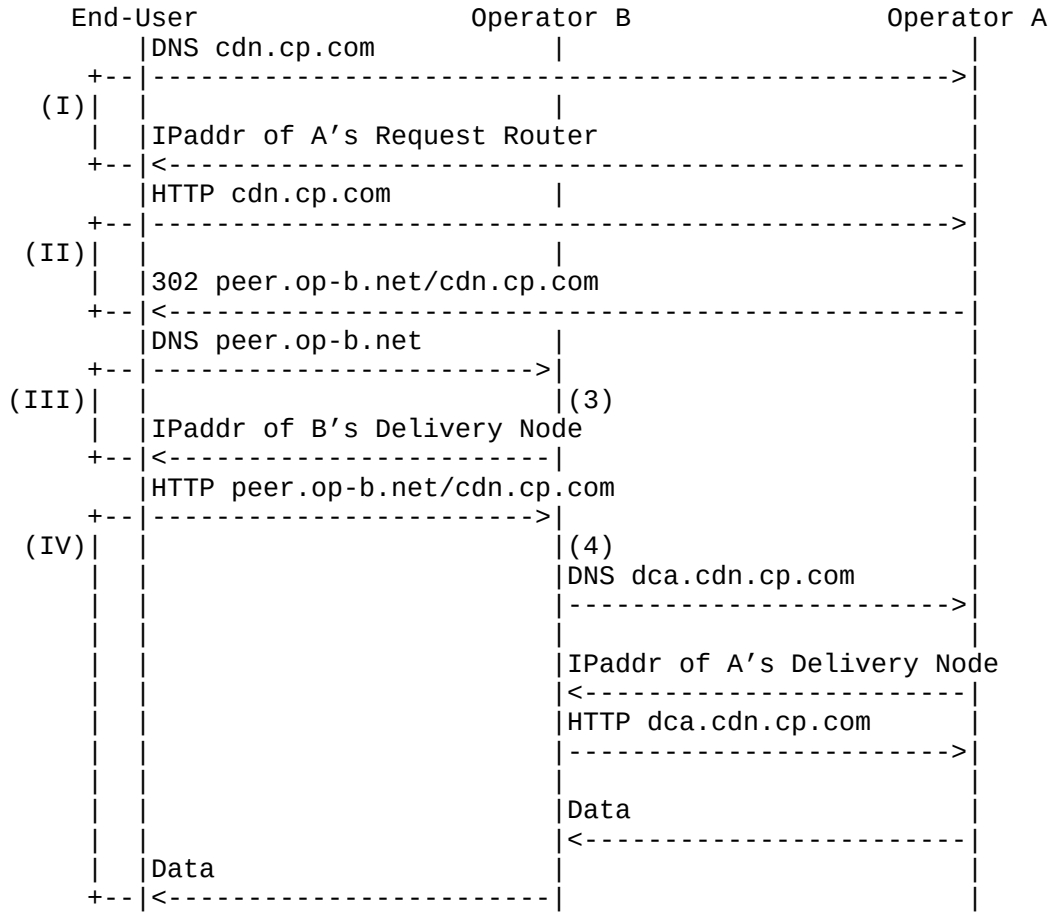


Figure 1: Method 1 of CDNI Strawman proposal

To give an impression about the times to expect for t_total here are some values for a DNS resolution process and also round trip times (RTT). NB: These numbers are measured from Germany via an ADSL access, German research network (DFN) access, and via a 3G network operator in Germany (Eplus). NB: Those numbers are not representative as they are out of single runs and not multiple runs spread over a larger time windows. They are rather given to illustrate the challenge.

Access-Type	DNS resolution (gmx.de,cached)	RTT (Germany)
ADSL (11 MBit/s)	26 ms	30 ms
DFN (32 MBit/s)	12 ms	10 to 20 ms
3G (EPlus)	200 to 300 ms	200 to 300 ms

Table 1: Some time measurements

As compared to a popular video site (e.g., youtube.com) where the video is delivered after roughly 2 seconds (19 ms to resolve FQDN of cache in charge, 2 ms for TCP 3-way handshake, and yet 2 seconds until the content is actually requested). This has been test from the DFN network connection. The 2 seconds time seem to come from the usage of the Flash player, as the network connection is up,running, and already being used for yet another request to the cache.

Here are the times, assuming that steps I, II, and III are consuming the RTT, but neglect the processing times:

- ADSL: t_total = 3*30ms + t_IV = 90ms + t_IV
- DFN: t_total = 3*20ms + t_IV = 60ms + t_IV
- 3G: t_total = 3*300ms + t_IV = 900ms + t_IV

The numbers for step I to III may look reasonable low (except for the 3G case), but t_IV isn’t know yet, leaving still potential to prolong the long process anyhow.

3. Failure Detection and Recovery

It is hard to foresee from the current request routing specifications how a failure in the downstream CDN is detected by the upstream CDN and such a failure case can be corrected.

4. Security Considerations

TBD

5. Conclusion

This draft lists some operational challenges for the request routing for the interconnection of CDNs and is more than incomplete. The current proposals ([I-D.peterson-cdni-strawman] and [I-D.xiaoyan-cdni-requestrouting]) for request routing seem not to consider the overall time from the first service request to the start of the content delivery.

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.lefaucheur-cdni-requirements]
Faucheur, F., Viveganandhan, M., Watson, G., and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-lefaucheur-cdni-requirements-01 (work in progress), March 2011.

[I-D.peterson-cdni-strawman]
Peterson, L. and J. Hartman, "A Simple Approach to CDN Interconnection", draft-peterson-cdni-strawman-01 (work in progress), May 2011.

[I-D.xiaoyan-cdni-requestrouting]
He, X., Li, J., Dawkins, S., and G. Chen, "Request Routing for CDN Interconnection", draft-xiaoyan-cdni-requestrouting-01 (work in progress), June 2011.

Author's Address

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Fax: +49 6221 4342 155
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Network Working Group
Internet Draft
Intended status: Informational
Expires: December 2011

Xiaoyan He
Jincheng Li
Spencer Dawkins
Huawei
Ge Chen
China Telecom

June 30, 2011

Request Routing for CDN Interconnection
draft-xiaoyan-cdni-requestrouting-01.txt

Status of this Memo

Copyright Notice

Abstract

This document describes recursive request routing procedures for CDN interconnection, it also describes content acquisition procedure using the HTTP protocol when cache miss occurs. The goal of the present document is to spur discussion about these procedures and to achieve a consensus on request routing procedures of CDNI.

Table of Contents

1. Introduction

CDNI request routing includes the following two scenarios:

Scenario A: request routing for user initiated requests, where once the upstream CDN receives the user request (DNS or HTTP), upstream CDN communicates with a downstream CDN to get the address of a delivery node to serve the user.

Scenario B: request routing for content acquisition, where the selected delivery node in downstream CDN receives the

user's request and encounters a cache miss, and communicates with upstream CDN to select a delivery node to acquire the content.

From procedure perspective, both iterative and recursive request routing are to be supported by CDNI solution as per requirement draft [I-D.draft-lefaucheur-cdni-requirements] (requirements R33 and R34 in version -01).

From protocol perspective, either DNS or HTTP[RFC2616] can be used for request routing.

Therefore, we can have four solution combinations for each scenario, see table below.

Case No.	Protocol		Procedure	
	DNS	HTTP	Iterative	Recursive
1	Y		Y	
2	Y			Y
3		Y	Y	
4		Y		Y

Table 1: solution combinations for CDNI scenarios

Document [I-D.draft-peterson-cdni-strawman] gives a discussion on iterative routing using DNS. The present document discusses recursive request routing procedures using DNS or HTTP.

Note: According to the proposed charter and current (individual) requirement document for CDNI, interfaces inside CDN are out of scope of CDNI, i.e. interactions

between a delivery node and RR in the downstream CDN during content acquisition should not be specified in CDNI; hence this document does not describe that interface in detail.

2. Conventions used in this document

The two terms "Iterative CDNI request routing" and "Recursive CDNI request routing" in this document are to be interpreted as defined in [I-D .draft-lefaucheur-cdni-requirements].

3. Request routing

3.1. Scope definition

This document focuses on Recursive CDNI request routing. For convenience, definition for recursive CDNI request routing from [I-D .draft-lefaucheur-cdni-requirements] is copied as below.

> o Recursive CDNI request routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can query the Downstream CDN Request Routing system via the CDNI Request Routing protocol (or use information cached from earlier similar queries) to find out how the Downstream CDN wants the request to be redirected, which allows the Upstream CDN to factor in the Downstream CDN response when redirecting the user agent.
> This approach is referred to as "recursive" CDNI request routing. Note that the Downstream CDN may elect to have the request redirected directly to a Surrogate inside the Downstream CDN, to the Request-Routing System of the Downstream CDN, to another CDN, or to any other system that the Downstream CDN sees as fit for handling the redirected request.

Other "recursive" mechanisms also utilized in CDNI, but not for interactions of RRs during request routing procedures such as the local DNS’s role in DNS lookup, are not in the scope of this document.

3.2. Protocol considerations on routing procedure

Two main protocols can be used for CDNI recursive request routing, i.e. HTTP or DNS. The present document obeys a principle that the protocol utilized between two RR systems keeps same as that utilized between the end user and the first touched RR to simplify the RR implementation.

3.3. Domain Name examples

There are several domain names involved in this document, below are their examples and explanations.

1. video.cp.example

In this document, a content provider with domain name cp.example contracts with a CDN provider. Domain name video.cp.example represents the specific sub domain to be accelerated by the contracted CDN.

Generally, the authoritative DNS server of domain video.cp.example has a record pointing to a domain of the contracted CDN,in order to direct the relevant DNS requests to that CDN.

A contractual CDN may ingest content from the contracted content provider in advance or dynamically. However, the ingestion procedures are out of scope of CDNI and therefore not described in this draft.

2. cdn.op-x.example

Operator X with domain name op-x.example provides CDN services using sub domain name cdn.op-x.example. This CDN-domain augmented with a prefix "peer" used to identify the request it received is from a peer CDN rather than from end users.

3.4. Operation codes considerations

In case of HTTP signaling used between RRs of connected CDNs, as one CDN may act as a downstream peer and an upstream peer for different CPs at the same time, it’s possible that a CDN receives requests from other interconnected CDNs for different purposes. e.g.selecting a delivery node for an end user or selecting a delivery node holding the content for a downstream peer. Therefore the CDN needs to distinguish these requests to act correspondingly.

Operation code "CDNIRoutereq" and "CDNIContentlocate" contained in the URL of HTTP requests is used for this purpose in the present document.

3.5. Client location's availability

To select the most appropriate delivery node ultimately to serve for an end user, usually the user's location is taken into account by the downstream CDN. Therefore, the user's IP address or FQDN should be conveyed to the downstream CDN from the upstream peer during CDNI request routing.

In case of HTTP redirection, the upstream CDN can get the specific IP address of the end user. It is proposed the upstream CDN convey this IP address or construct a FQDN to the downstream CDN for optimal routing decision.

In case of DNS redirection, the upstream CDN may obtain the end user's IP address through the DNS client subnet extension or by embedding the client IP address in the DNS name [I-D.vandergaast-edns-client-subnet]. It is proposed that the upstream CDN forwards this info to the downstream CDN directly for optimal routing decision. If the UE does not support this DNS extension, the upstream CDN therefore will not convey sort of this extension to its down peer. It is proposed that the downstream CDN returns the request router's address. After that the specific HTTP request for content will be sent to the request router,at this point, the request router can elect a more accurate delivery node for the user based on its IP address.

4. Configuration summary

Interconnection CDNs must exchange the following information to peer with each other:

- The IP address of the entry point of the CDN or distinguished CDN domain name; and
- Set of IP prefixes for which the CDN is prepared to deliver to end-users.
- Set of CP domain names for which the CDN is served.

When a Request Router in an upstream sees an end-user IP address best served by a downstream peer, upstream CDNs needs to forward the request to the configured entry point of the downstream peer, or to result of DNS lookup for the configured downstream CDN's domain name.

When a delivery node in a downstream receives content requests from end users and result in cache miss, it verifies that the CP domain

contained in the URL is served by a known CDN peer based on configuration, and if so, issues a content acquisition request to that peer.

5. Request routing approaches

5.1. First approach

This alternative utilizes HTTP redirection between the end user and the upstream CDN. On reception of HTTP request, the upstream CDN communicates directly with selected downstream CDN to get a delivery node.

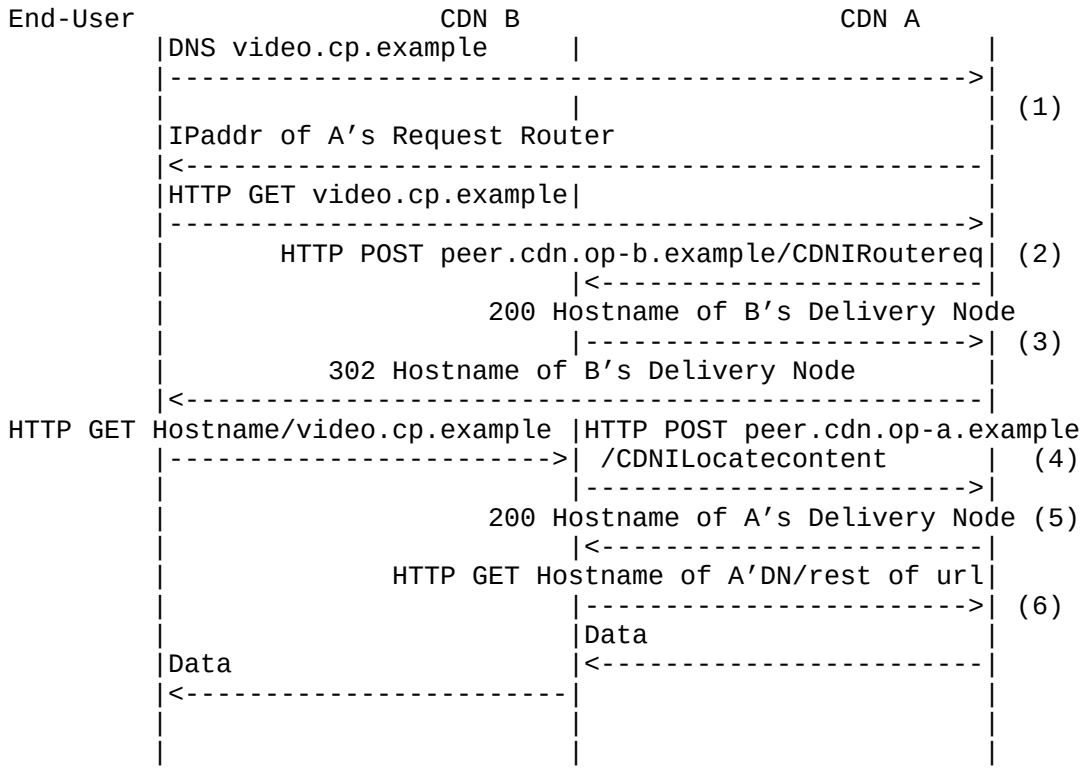


Figure 1: Request routing for approach one

1. A Request Router of CDN A processes the DNS request for its customer and recognizes that the end-user is best served by another CDN. CDN A returns the IP address of its Request Router to get the HTTP request of the user.

Note: The domain name in step1 of the CP e.g.video.cp.example should be changed to the CDN A's when the request is redirected to CDN A. For simplicity, this is not shown in the figure.

2. A Request Router of CDN A processes the HTTP request and again recognizes that the end-user is best served by another CDN, so based on the configuration and result of a potential DNS lookup, CDN A issues an HTTP POST to CDN B's Request Router, with the distinguished CDN b's domain name and a command code "CDNIRoutreq" contained in the URL. The command code is used to explicitly indicate this is a request from a peer for delivery node selection. Other parameters e.g.the end user's IP address and the CP's domain name shall be included in the body of the HTTP POST.

3. CDN B's Request Router recognizes the request is from a peer CDN, if the end user's IP is included, it selects and returns a suitable delivery node for the user based on its location. The address of this node is returned to the end user via CDN A.
   Based on local policy, CDN B may return the request router's address instead of that of a delivery node.

4. The end-user requests the content from CDN B's delivery node, potentially resulting in a cache miss. From the URL CDN B verifies that this CP-domain served by a known peer. It then sends a HTTP POST with a command code "CDNILocatecontent" and CDN A's distinguished domain name contained in the URL to CDN A's Request Router. Other parameters such as playback URL of the content shall be included in the body of the HTTP POST.
   If the CDN B's Request Router address is returned in step 3, the content request will first arrive at CDN B's Request Router and a 302 response with the address of the selected delivery node will be sent back to the end user.

5. CDN A recognizes that the HTTP request is from a peer CDN rather than an end-user based on the command code and so returns address of a delivery node.

6. CDN A serves content for the requested CDN-domain.

5.2. Second approach

This approach utilizes DNS redirection between an end user and the upstream CDN. On reception of DNS lookup request, the upstream CDN communicates directly with the selected downstream CDN to locate delivery node.

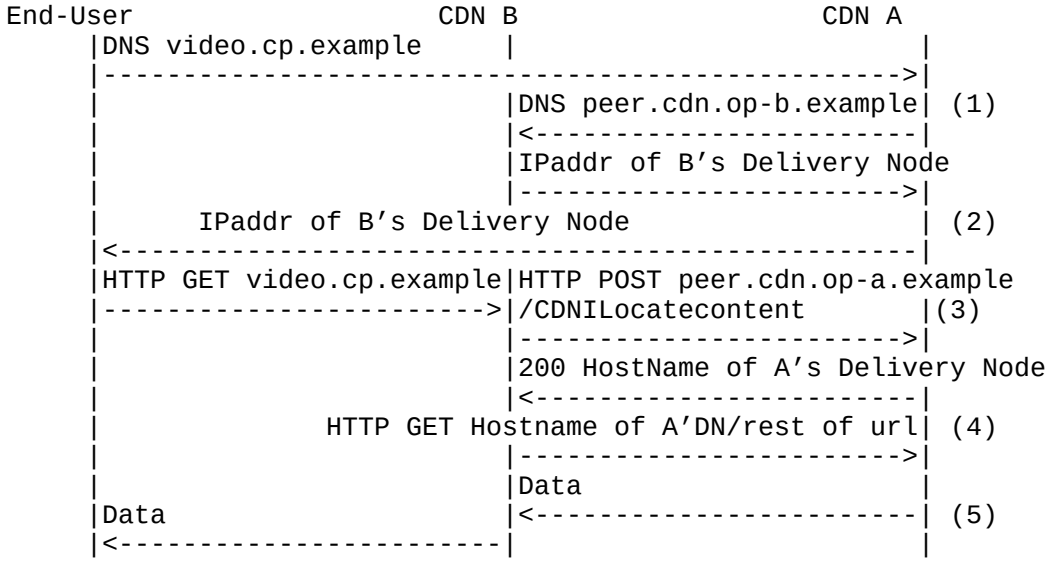


Figure 2: Request routing for approach two

1. A Request Router of CDN A processes the DNS request for its customer and recognizes that the end-user is best served by another CDN. The client IP used in this determination is obtained either through the DNS client subnet extension or by embedding the client IP in the DNS name [I-D.vandergaast-edns-client-subnet]. Based on configuration, the Request Router changes the domain name to CDN B's distinguished name and forwards the DNS request to CDN B's Request Router.

2. CDN B's Request Router recognizes the request is from a peer CDN based on the distinguished domain name, it returns a suitable delivery node. The address of this node is returned to the end user via CDN A. Or based on local policy or CDN B did not get the address of the client in step 1, CDN B may return the address of the Request Router.

3. The end-user requests the content from CDN B's delivery node, potentially resulting in a cache miss. From the URL CDN B verifies that this CP is served by a known peer. It then issues an HTTP POST with command code "CDNILocatecontent" and CDN A's distinguished domain name contained in the URL to CDN A's Request Router. Other parameters such as playback URL of the content shall be included in the body of the HTTP POST. If the CDN B's Request Router address is returned in step2, the content request will first arrive at CDN B's Request Router and a 302 response with the address of selected delivery node will be sent back to the end user.

4. CDN A recognizes that the HTTP request is from a peer CDN rather than an end-user from the command code and so returns the address of a delivery node.

5. CDN A serves content for the requested CDN-domain.

6. Conclusions

Priorities for proposed approaches in present document will be worked out later based on discussion on the email list.

7. Security Considerations

For the recursive request routing approach described in Section 5.2, the user agent may determine to convey its IP address to the CDN by including its IP address in the DNS client subnet extension or embedding the IP address in the DNS name [I-D.vandergaast-edns-client-subnet] for optimizing the routing.

In case of CDNI, if [RFC1918] address space is utilized by the end user, the [RFC1918] IP address should not be paased across the addressing boundary to the downstream CDN, for several reasons,which include the possibility that the downstream CDN might also use overlapping [RFC1918] address space.

A possible approach for this is that when the upstream CDN recognizes that an [RFC1918] address is used by the user agent, it shall change this address to the global unique IP address of the local DNS for the end user from the IP packet header and transfer this address to the downstream peer. The downstream CDN then may select a delivery node for the user based on the local DNS's address or just return the RR's address to the upstream CDN.

8. IANA Considerations

If the approach described in this document is adopted, we would request that IANA allocate the command codes "CDNIRoutereq" and "CDNIContentlocate" in the HTTP Parameters registry.

9. References

9.1. Normative References

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC1918] "Address Allocation for Private Internets", Y.Rekhter, B.Moskowitz,D.Karrenberg, G.J.de Groot,E.Lear, February 1996.

[I-D.draft-lefaucheur-cdni-requirements] "Content Distribution Network Interconnection (CDNI) Requirements", Francois Le Faucheur, Mahesh Viveganandhan, Grant Watson, Yiu Lee, 14-Mar-11, <draft-lefaucheur-cdni-requirements-01.txt>

[I-D.draft-peterson-cdni-strawman] "A Simple Approach to CDN Interconnection", Larry Peterson, John Hartman, 18-May-11,<draft-peterson-cdni-strawman-01.txt,.pdf>

[I-D.vandergaast-edns-client-subnet] Contavalli, C., van der Gaast, W., Leach, S., and D. Rodden, "Client subnet in DNS requests", January 2011.

9.2. Informative References

Authors' Addresses

Xiaoyan He
Huawei
B2, Huawei Industrial Base, 518129
China

Phone: +86 158 2938 5137
Email: hexiaoyan@huawei.com

Jincheng Li
Huawei
B2, Huawei Industrial Base, 518129
China

Phone: +86 139 9195 3074
Email: lijincheng@huawei.com

Spencer Dawkins
Huawei
1700 Alma Drive,Suite 100 Plano,TX 75075 USA

Phone: +1 469 229 5397
Email: spencer@wonderhamster.org

Ge Chen
China Telecom
109 West Zhongshan Ave,Tianhe District,Guangzhou,P.R.C

Phone: +86 133 1609 0408
Email: cheng@gsta.com

IETF cdni
Internet-Draft
Intended status: Informational
Expires: February 18, 2012

ZP.Zhou
Huawei Technologies, Inc.

Aug 17, 2011

CDNI use case
draft-zhou-cdni-use-case-01

Abstract

Industry needs the CDN interconnection to provide the CDN service that may cover a wider geographical area and a better service quality. In the real pragmatic operation, the relationship between two CDNs should concern many factors(for instance, CDN or CP may select the downstream CDN based on some principals or conditions or service authorizations), hence the CDN interconnection is defacto a sort of constrained interconnection. This document will give the use cases to indicate the models of how the CDN interconnection may be used and the associated examples for the use cases will be listed subsequently.

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 February 18, 2012.

Copyright Notice

Table of Contents

1. Introduction

[I- D.bertrand-cdni-use-cases] has provided some basic use cases for CDN Interconnection. While it has not talked about how CDN-A may interconnect CDN-B as its downstream CDN(e.g. How the downstream CDN is selected; what operation rule(business rule) should be taken into account on service of CDN interconnection).

This draft gives some use cases concerning the operation of multiple CDN federation and as a result some examples are provides.

2. Terminology

Roughly, this document may refer the terminology defined in section 1.1 of [I-D.jenkins-cdni-problem-statement] and [I-D.bertrand-cdni-use-cases].

3. Architecture for CDN-I Service

Following, a general service architecture of CDN interconnection is listed as Figure 1:

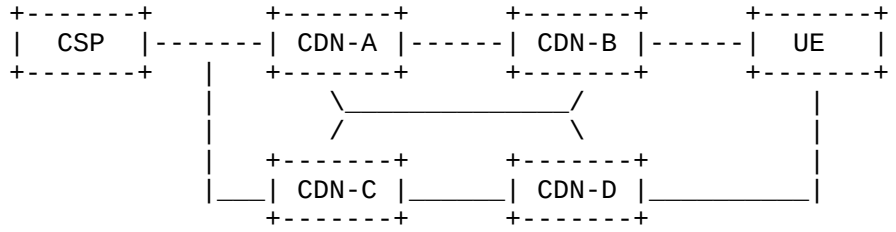


Figure 1. Typical Architecture for CDN interconnection

This is just a typical architecture of CDN interconnection. The CSP may connect multiple CDNs and a upstream CDN may connect multiple downstream CDNs(such as CDN-A may connect either CDN-B or CDN-D; similarly to CDN-C)

This figure is used for the following discussion of use cases while in reality it may be possible that the CSP only select one delegate CDN or a upstream CDN may only have one downstream CDN.

4. Operating Requirements on CDN Interconnection

In the real operation, some basic requirements of CDN-I service based

on service operating should be abided, including:

Authorization area for CDN provider: the CDN provider may only provide CDN service in a specific area, such as CDN-A may only serve in Beijing and CDN-B may only serve in Shanghai. Such authorization may be determined by CSP or by upstream CDN for the downstream CDN. And such authorization is because of the business model(for example to protect the business of authorized CDN service provider). Therefore the associated policy should be supported in the CDN interconnection.

Constraint of service by content: the access area or access period of specific content may be constrained in content service. For example the movie in a Chinese video website may only be accessed by the terminals located in China mainland and if being overseas the request for the movie service will be rejected. The rule in this use case comes from the content rights operation and should be conducted by the CDN service provider.

Content adaptation: the format of content service may be converted according to the terminal’s capability or consumer’s application. For example, convert multicast (or unicast) RTP streams to HTTP streaming and then deliver the adaptive stream to terminals(refer to MCD CDN-I use case and requirement). As for implementation of CDN service, it is optimum maintaining and delivering one format of content within CDN system(including CDN to CDN) while providing the adaptive streaming service on the interface between CDN and terminals. The content adaptation may be deployed on a node of the CDN or CDN-I system as a functionality.

Content priority: towards the content service, one way to provide the differ-service of content delivery is to set priority for the content. Such priority mechanism may be supported and extended between CDNs. The downstream CDN may inherit the content priority from CSP or upstream CDN or the priority may even be updated at the downstream CDN depending on the service.

Totally the requirements above are usual rules of content service operating and are also applicable for CDN-I use cases.

5. Use Cases

In this section, three Use Cases to realize the requirements are described along with examples.

5.1. CDN authorization

In this use case, the CP may make a list of pairs of CDN and its authorized service area. For example:

CDN Provider	Service Area
CDN-A	Beijing
CDN-B	Shanghai
CDN-C	Guangdong
....	....
CDN-Z	Tianjing
CDN-HK	Hongkong

Figure 2: Example List of CDN Authorization Areas

The list of CDN authorization areas may be created by the CP and be kept by all CDNs. When delivering content, CDN may refer to this list to determine with which CDN it has to connect. Here gives two sub-cases:

Sub-Case one: for the unicast service, when the CP receives a content request from a UE located in Beijing, CP will redirect that request to CDN-A. CDN-A then will check whether the requested content is available locally. If yes, then provide the content to the requesting UE, else if the content at that time is only stored in Tianjing and Guangdong, CDN-A will contact CDN-Z or CDN-C getting that content and at last issue that content to the UE.

Sub-Case two: for the broadcast service, the content broadcast route may be designed based on the geographical relationship of each area. For example Tianjing is very near to Beijing. If the CP is also located in Beijing, for the content broadcast, the content may be delivered from CP to CDN-A first and CDN-A will then forward the content to CDN-Z. In the same way, the content to CDN-HK may be forwarded from CDN-C since Hongkong is near to Guangdong. To design such transport route should consider the transport efficiency, neither overlap nor neglect an area.

Further, if concern there are possibly multiple CDNs in an area, for example CDN-Ax and CDN-Ay both serve in Beijing, there should be an entity coordinating these two CDNs(CDN-Ax and CDN-Ay) and contacting

other CDNs outside of Beijing. Or probably each CDN in the same area is responsible for different services, e.g. CDN-Ax serves for Mobile request and CDN-Bx serves for request of IPTV service.

The arrangement above may be seen as a typical deployment policy(may be beneficial for both efficiency and business cooperation). The core issue is to keep all CDNs complying with the same policy.

5.2. Constraint for Content

The arrangement of content publishing/content service is another typical policy controlled by CP. For example, the rights of OlympicGame in 2008 for live broadcast in China Mainland was purchased by CCTV and all related content may only be accessed in China Mainland. Hereby a terminal from outside Mainland of China such as Hongkong is prohibited to access the live Olympic broadcast or video on demand issued from CCTV during Olympic period(Hongkong users may access Olympic content from another content provider locally in Hongkong). Hence for all the CDNs within China during Olympic period, the content forward should abide this constriction. If broadcast both Movie-A and OlympicGame B, CDN-C should deliver content of Movie-A to CDN-HK but should not deliver content of OlympicGame B to CDN-HK.

Content Name	Service Area	Service Period
Movie-A	China	2009.01--2009.12
OlympicGame B	China Mainland	2008.08--2008.10

Figure 3: Example List of Constraint for Content

When the service period expires, the CDNs should remove the related content.

This constraint in fact has controlled the broadcast service as a limited broadcast. This is another typical use case of broadcast or multicast over CDN interconnection.

5.3. Content Adaptation

To support the diversity of terminals and the fluctuant network band width, adaptation measures may be conducted for the content services. While for the CDN operator, it is better only to maintain minimum(only one) content format(s). So, one solution is to transform the content format prior to delivery. It means the content forwarded between CDNs may be RTP streams of best quality(e.g. HD

video and Hi-Fi audio) while the RTP stream may be transformed to adaptive streaming to serve for user. One possible realization is that the delivering CDN which is nearest to the UE performs such delivery format adaptation according to the UE's capabilities. The adaptation may include, but is not limited to, conversion to adaptive streaming, the transport format conversion, codec type conversion, content encapsulation conversion and content metadata conversion (e.g. content description, program information, etc). For example, content is delivered from Content Provider through CDN-A to CDN-B in the form of normal RTP streams. The delivering CDN, i.e. CDN-B, determines that a UE requires the content to be delivered using adaptive HTTP streaming, CDN-B now transforms the RTP streams to an adaptive HTTP stream that can be delivered to the UE. See as figure 4:

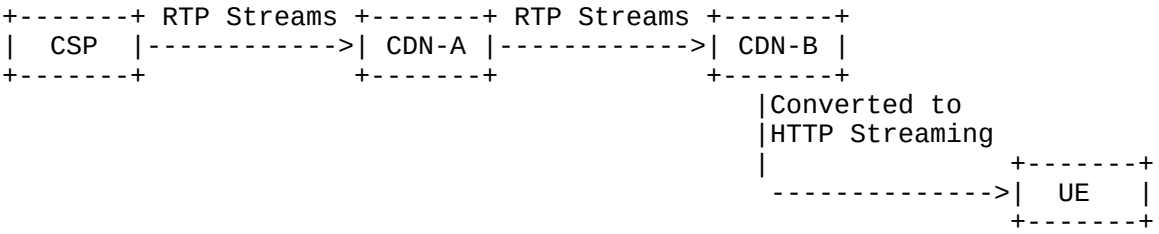


Figure 4. Example of Content Adaptation

This use case is applicable for either broadcast or unicast service; for either live Broadcast or VOD service. The content may be transformed and stored as different files with different formats/ rates. According to the UE capability(media resolution, media types, bandwidth, etc), the content request will be forwarded to specific content link of different content files/live broadcast sources.

5.4. Content Priority

For the differ-service of content delivery, one way is to set priority for the content. For example the content for more users may be set a high priority and will be served first.

For example, when CDN-A receives two tasks to deliver two contents to its downstream CDNs, CDN-A will check the priority of these two contents. Since content-A is a live broadcast of football game to millions of audiences and has been set a very high priority, CDN-A will handle the task of content-A with high priority.

In general, the content should be set a priority before it is delivered. The content priority may be set by CSP, Upstream CDN or the UE(the user) and the priority may be set whether it may be

updated or not in terms of service implementation. For instance, the user may set his subscribed content with a high priority with high price and may change the priority sometimes, but the user may not set the priority of some contents such as the advertisement content.

With the content priority, the CDN may also provide a VAS for the user. If the user has subscribed two contents and when there is a time conflict of these two content services, CDN will automatically deliver the content service with high priority to the UE. One example is, the user has subscribed two broadcast content services: the News program from 6 to 10 o'clock and a football game from 7 o'clock but the football game is set a higher priority. At 6 o'clock, the CDN starts to deliver News content to UE and when time arrives at 7 o'clock, the CDN will automatically stop the News content delivery but start to deliver the content of football game. Accordingly, the UE will alter to play the football game from the News. When the football game finishes at 10 after 9 o'clock, the CDN will automatically deliver again the News to the UE.

This use case is applicable for either broadcast or unicast service; for either live Broadcast or VOD service. The core requirement is the mechanism of priority management(e.g. set, exchange, update).

6. Application Example

Taking TISPAN CDN architecture as example in this section, a full implementary instance for broadcast service performing the operating requirements is given as below:

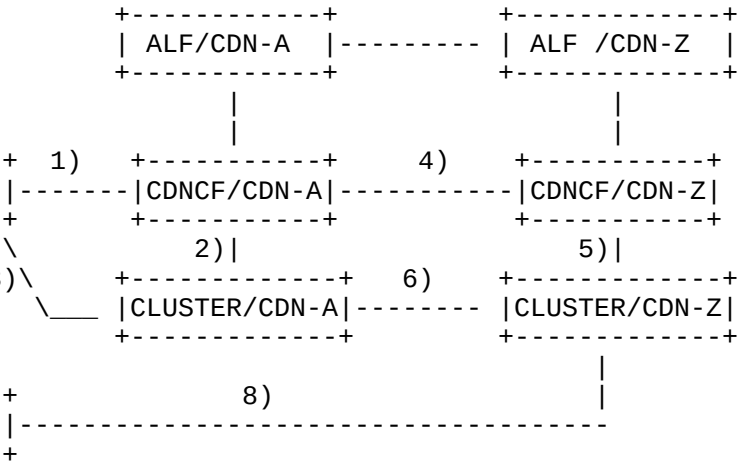


Figure 5: Example of Broadcast Service via CDN Interconnection

Execution steps for this procedure may be as following:

step 1: CSP informs to broadcast the content.

step 2: CDNCF in CDN-A informs its clusters to receive/get the content(sports game with HD video format).

step 3: Cluster in CDN-A receives/gets the content from CSP.

step 4: CDNCF in CDN-A informs another CDNCF in CDN-Z to receive/get the content from the cluster in CDN-A.

step 5: CDNCF in CDN-Z informs its clusters to receive/get the content from CDN-A.

step 6: cluster in CDN-Z receives/gets the content from cluster in CDN-A.

step 7: A UE in Tianjing subscribes the broadcast service. Through a general CDN procedure, the UE is informed the address of cluster under CDN-Z where UE may receive the broadcast content. Additionally, CDN-Z may provide HTTP adaptive streaming and the content received from CDN-A is transformed in adaptive streaming for the UE.

step 8: UE receives the broadcast service from CDN-Z.

7. Security Considerations

The involved service information should be guaranteed unchanged. More detail may be provided later.

8. IANA Considerations

This document requires no actions from IANA.

9. Acknowledgements

Many thanks to all the members of Huawei Software CDN project team and all the friends met in the CDN standard meetings.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[TISPAN] ETSI TISPAN, "CDN Architecture(ETSI TS 182 019)".

[MCD] ETSI MCD, "Media CDN Interconnection, use cases and requirements(ETSI TS 102 990)".

[IETF-CDNI-usecase]
IETF CDNI, "[Draft-bertrand-cdni-use-cases]".

[IETF-CDNI-problem-statement]
IETF CDNI, "[Draft-jenkins-cdni-problem-statement]".

Author's Address

Zhipeng Zhou
Huawei Technologies, Inc.
No.101, Software Avenue, Yuhuatai District
Nanjing 210012
P.R.China

Phone: +86-25-56620690
Email: zhouzhipeng@huawei.com