ETSI TS 182 032 V1.1.1 (2013-04)

CDN Interconnection Architecture

Reference
DTS/NTECH-00002-CDN-I-stage2

Keywords
architecture, CDN, interconnection

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Network Technologies (NTECH).

Introduction

Content Delivery Networks (CDN) are systems for the efficient delivery of digital objects (e.g. files with multimedia content as video on demand or other file types) and multimedia streams (e.g. live television streams) over IP networks to many end points and viewers. Typically, a CDN consists of one or more servers that deliver the digital objects and/or streams, and a management/control system. The management/control system takes care of content distribution, request routing, reporting, metadata and other aspects that make the system work.

Currently implemented CDNs are based on a widespread network of distribution nodes controlled by a functional entity taking care of content distribution decisions. The control functional entity keeps track of all content locations, manages distribution amongst the distribution nodes and/or clusters and also decides which distribution node should serve a client request.

Every CDN may have a specific network footprint in which it can deliver content effectively. This footprint depends on the locations of distribution nodes that the CDN controls. It is often inefficient to build new distribution nodes that are not in the region in which the company running the CDN is operating. A solution is to interconnect two CDNs and share existing infrastructure for content delivery. Interconnection of CDNs is achieved by interconnecting the centralized functional entities that represent the logic behind the decisions in each CDN. This is called CDN-I or Content Delivery Network Interconnection.

The present document defines the architecture and principles of the interconnection between CDNs according to the requirements defined in TS 102 990 [1], which also contains informational use cases for CDN interconnections.

Other sources of relevant information on CDN interconnection topics are the following organizations:

- ETSI (e.g. TS 182 019 [i.6], TR 102 688-9 [i.1])
- IETF: WG CDN-I (e.g. RFC 6707 [i.2])
- ATIS Cloud Services Forum (e.g. ATIS-0200003 [i.3], ATIS-0200004 [i.5])

1 Scope

The present document specifies the architecture and functions of a CDN Interconnection system, implementing the requirements defined in TS 102 990 [1].

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 990: "Media Content Distribution (MCD); CDN Interconnection, use cases and requirements".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 102 688-9: "Media Content Distribution (MCD); MCD framework; Part 9: Content Delivery Infrastructures (CDI)".

[i.2] IETF RFC 6707: "Content Distribution Network Interconnection (CDNI) Problem Statement", September 2012.

NOTE: Available at http://tools.ietf.org/html/rfc6707.

[i.3] ATIS-0200003: "CDN Interconnection use case specification and high-level requirements".

NOTE: Available at http://webstore.ansi.org/RecordDetail.aspx?sku=ATIS-0200003.

[i.4] ISO/IEC 23009-1: "MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH)".

[i.5] ATIS-0200004: "CDN interconnection use cases and requirements for multicast-based content distribution".

NOTE: Available at http://webstore.ansi.org/RecordDetail.aspx?sku=ATIS-0200004.

[i.6] ETSI TS 182 019: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) architecture".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

NOTE: Some of the following definitions are from TS 182 019 [i.6] and TR 102 688-9 [i.1].

CDN Interconnection: interconnection between two CDNs, enabling the controlled distribution of content between those CDNs

content delivery: act of delivering deployed content to a user

Content Delivery Network (CDN): set of functions managing content acquired from content sources, through delivery to the user

content acquisition: act of acquiring content from a content source

content deployment: act of moving ingested content to one or more network entities, based on content deployment policies

content distribution: act of moving content in and between CDNs

content ingestion: act of introducing content (and associated data) into the Content Delivery Infrastructure

content item: uniquely addressable content element in a CDN.

NOTE: A content item is defined by the fact that it has its own Content Metadata associated with it. It is the object of content distribution and request routing operations in a CDN. Example of Content Items are a video file/stream, an audio file/stream, an image file or segmented content together with an associated manifest file.

downstream: side of the CDN interconnection that is closest the Consumer

logging: recording events related to content items, request routing and content distribution

manifest file: file that describes the composition of segmented content

metadata: data about content items and CDN network specifics

reporting: providing access to recorded events related to content items, request routing and content distribution

request routing: exchange of information be two CDNs to aid routing of requests of users for content

segmented content: content composed of multiple files, or content composed of multiple streams, or content composed of one or more files and one or more streams

upstream: side of the CDN interconnection that is closest to the Content Provider

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AAC	Advanced Audio Coding
ALF	Asset Location Function
AMT	Automatic Multicast Tunnelling
AS	Autonomous System
ATIS	Alliance for Telecommunications Industry Solutions
BGP	Border Gateway Protocol
CCF	Cluster Controller Function
CDF	Content Delivery Function

CDN	Content Delivery Network

NOTE: Industry sometimes uses "Content Distribution Network".

CDNCF	Content Delivery Network Control Function
CDN-I	Content Delivery Network Interconnection
CSF	ATIS Cloud Services Forum
dCDN	Downstream Content Delivery Network
DCF	Distribution-of-Content Function
dDCF	Downstream Distribution-of-Content Function
dICF	Downstream Interconnection Control Function
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
dRCF	Downstream Request-routing and Content-control Function
DRM	Digital Rights Management
FLV	Flash video
GPS	Global positioning system
HD	High Definition
HDS	HTTP Dynamic Streaming
HLS	HTTP Live Streaming
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICF	Interconnection Control Function
IP	Internet Protocol
MF	Manifest File
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MSS	Microsoft Media Server
OCEAN	Open ContEnt Aware Networks
RCF	Request-routing and Content-control Function
REST	Representational state transfer
RTMP	Real Time Messaging Protocol
RTSP	Real Time Streaming Protocol
SDO	Standards Developing Organization
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
TISPAN	Telecommunications and Internet converged Services and Protocols for Advanced Networking
uCDN	Upstream Content Delivery Network
uDCF	Upstream Distribution-of-Content Function
UE	User Equipment
uICF	Upstream Interconnection Control Function
uRCF	Upstream Request-routing and Content-control Function
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WMV	Windows Media Video

4 High-level overview

4.1 CDN interconnection services and capabilities

CDNs are in general autonomous networks offering different services to their users. A CDN's primary function is to optimize content distribution and delivery. In addition to this primary function many CDNs chose to implement various other capabilities like content manipulation, digital rights management (DRM), intelligent handling of segmented content and others.

Because of this variation, the present document specifies a basic set of capabilities that every interconnected CDN shall support. In addition to the basic capability set, the present document specifies an extended capability set, describing non-mandatory capabilities available for the CDN interconnection environment. The CDN interconnection shall support a capability exchange mechanism described later in the present document.

Requirements defined in TS 102 990 [1] shall apply.

4.2 CDN interconnection capabilities

Table 4.2.1 provides a table of all the capabilities available for CDN interconnection, grouped into the basic and extended capability set. It also mentions whether those capabilities are mandatory or optional.

Table 4.2.1: Capability Overview Table

	Basic capability set	Extended capability set	Description
Interconnection Control	Mandatory		Responsible for management of the peering relationship between two CDNs
Request Routing	Mandatory		Capabilities responsible for making it possible for CDNs to direct client requests
Content Distribution	Mandatory		Distribute content to other interconnected CDN
Footprint Exchange	Mandatory		Responsible for announcing the network footprints the CDNs are offering to serve
Metadata Exchange	Mandatory		Used to exchange content related information
Content Status Exchange	Mandatory		Used for distribution of real-time status related to the content
Report Exchange	Mandatory		Used for possibly delayed distribution of comprehensive information gathered during the content delivery process
Capability Exchange	Mandatory		Capability to exchange information about the availability of extended capabilities in CDNs
Metadata Defined Reporting		Optional	Capability of creating reports according to content provider parameters defined in metadata
Content Integrity Control		Optional	Capability to maintain content integrity
Content Adaptation		Optional	Capability to make content processing and adaptation within the CDN
Multi-Segment Content Support		Optional	Capability to handle multi-segment content efficiently (for instance logging segment download sessions per session not just per segment)
Content Access Control		Optional	Capability to define advanced content access control rules
Content Security and DRM		Optional	Capability to use cryptographic technologies to maintain content security and DRM
Custom Capability Support		Optional	Capability to support other (not standardized) capabilities

4.3 CDN-I basic capability set

Every CDN that wants to take part in CDN interconnection shall have a basic set of interconnection related capabilities. These capabilities give the CDN the ability to properly respond to requests coming from other CDNs. The capabilities included in the basic service set are listed in figure 4.3.1.

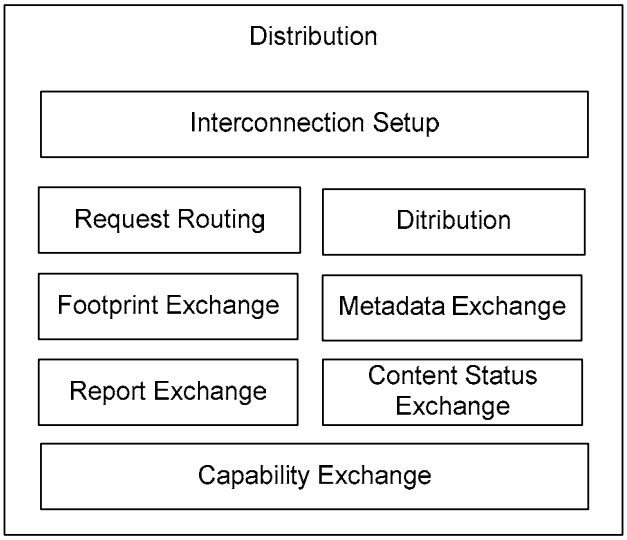

Figure 4.3.1: Basic CDN-I capability set

4.4 CDN-I extended capability set

Whereas the basic capability set is sufficient for basic CDN interconnection many CDNs have additional capabilities that may not be available in all CDNs participating in the CDN federation. The presence of optional capabilities in the CDN federation requires those CDNs that wish to make use of those capabilities to have the capability to exchange information about the availability of extended capabilities in CDNs. This capability is called Capability Exchange and it is mandatory.

Some of the capabilities included in the extended capability set are listed in figure 4.4.1.

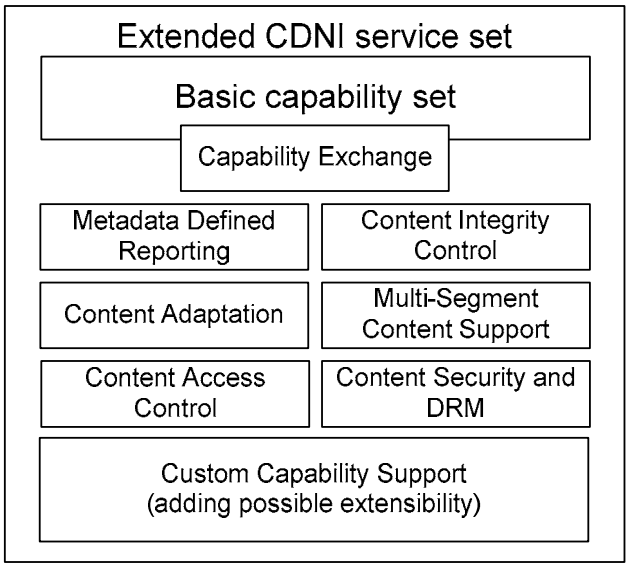

Figure 4.4.1: Extended CDN-I capability set

4.5 Compliance

A CDN interconnection solution is compliant to the present document if the following points are fulfilled:

- All mandatory services and features are implemented as specified in the present document.
- If an optional service or feature is implemented, then it is implemented as specified in the present document.
- If an optional service or feature is implemented, and the present document specifies multiple options, then it is implemented according to at least one or those options.

5 Overview of functional entities

5.1 Functional architecture for CDN interconnection services

The overall functional architecture for CDN interconnection service is shown in figure 5.1.1. The functional architecture is based on a multi-layer architecture that enable separate functionalities used for interconnection of CDNs that may involve up to 3 different level of functionalities and related reference points required. The first layer is responsible for content distribution, the second for controlling of content and request, and the third for the controlling of the interconnection itself. The CDN-I functional architecture should enable CDNs to agreed on a minimal level of interconnection, depending on available capabilities and their needs. In case of CDN peering both interconnected CDNs may play both the role of upstream and/or downstream CDN, depending of direction of content request.

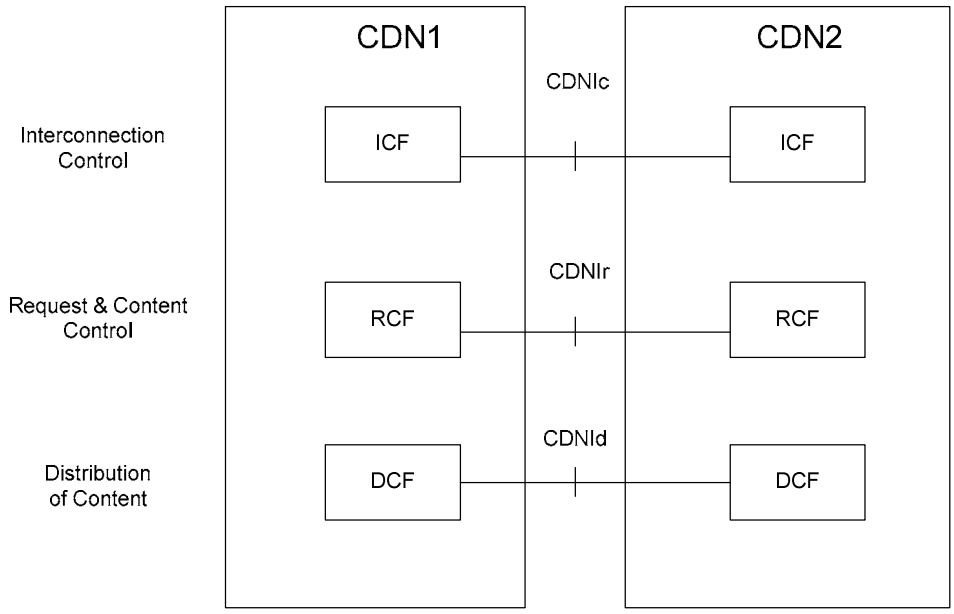

Figure 5.1.1: Functional architecture for CDN interconnection services

5.2 Functional entities

The Content Delivery Network contains one or more Distribution of Content Function (DCF) that can be grouped geographically or administratively in clusters and contains several delivery nodes (that distribute content to other CDN or to end user) hidden to other CDN. CDN shall contain one or more Request and Content Function (RCF) that process requests related to content distribution control and routing request. A CDN Interconnection Control Function (ICF) is responsible for the management of the interconnection. All CDN interconnection entities may support topology hiding and provide abstraction layer from internal CDN architecture.

NOTE: Co-location of interconnection entities with existing CDN entities is possible. For example in case that one of the interconnected CDNs is based on ETSI CDN specification [i.6] there is possibility to collocate ICF with CDNCF, RCF with CCF and DCF with CDF.

5.2.1 CDN Interconnection Control Function (ICF)

A CDN Interconnection Control Function (ICF) shall manage, create, terminate and exchange CDN networks properties, status report required for CDN interconnection between two or more CDNs (CDN peers).

An ICF contains following functionalities:

- Footprint Exchange - enable CDNs exchange footprint information.
- Capability Exchange - the upstream CDN gets information about capabilities of downstream CDN. This information can be used by Request Control Function.
- Network Status Reporting - the upstream CDN gets status of downstream CDN network. This status is used by Request Control Function. If for example downstream network reports problems, upstream CDN will serve request by itself, or via other functional downstream CDN.
- Network Logging - the downstream CDN sends logs to upstream CDN network. These logs are important for content provider and also for CDN administration.

IP interconnection setup and service agreement between CDN providers shall be realized prior to logical CDN interconnection. Details of these activities are out of scope of the present document.

5.2.2 Request and Content Control Function (RCF)

The Request and Content Control Function (RCF) is responsible for content control and request routing as well as exchanging metadata related to content control.

The RCF contains following functionalities:

- Metadata exchange function - Metadata are sent from upstream CDN to downstream CDN. Downstream CDN can then make room for content and can inform upstream CDN about content handling possibility.
- Content request function - one CDN can request content from other CDN. Different request routing models can be used (e.g. push/pull model, chaining or redirecting requests).
- Content status reporting - The Upstream CDN gets status of content from downstream CDN. This status can be than stored local database. Also events can be invoked on status change. Also Downstream CDN can inform upstream CDN when content status has been changed.

5.2.3 Distribution of Content Function (DCF)

Distribution of Content Function (DCF) is responsible for distribution of content between CDNs in form of files, streams, metadata.

The DCF contains following functionalities:

- Transfer of file-based content.
- Publication and streaming of stream-based content.
- Content metadata distribution (if metadata distributed with content).

5.3 Reference points

5.3.1 ICF - ICF (CDN-Ic)

This reference point is between two ICFs and it is used for controlling interconnection peer and transferred over this point information related to CDN capabilities and status, including footprint exchange, capability exchange, interconnection status reporting and network usage/performance logging.

5.3.2 RCF - RCF (CDN-Ir)

This reference point is between two CCFs and it is used for requesting content and to transfer content related information, including content metadata exchange, content requests and content status reporting.

5.3.3 DCF - DCF (CDN-Id)

This reference point is between two DCFs. Content files, content streams and content related data (if distributed as part of content) are transferred over this point.

6 Procedures

This clause specifies CDN-I procedures. Clause 6.1 specifies the 3 main phases of interconnection. The following clauses describe specific procedures related to different capabilities that were agreed during the interconnection phase.

6.1 CDN interconnection phases

The basic process of CDN interconnection may consist of three basic phases:

- Interconnection establishment, during which the CDNs negotiate the interconnection.
- Interconnection phase, during which the CDNs are fully interconnected and able to share their resources.
- Interconnection release, during which the interconnection between the CDNs is released.

The interconnection establishment and release phases may be omitted in systems that prefer manual provisioning and in that case interconnection establishment and configuration is performed statically by both interconnected CDN providers before interconnection phase itself.

The procedure related to the interconnection establishment is described in clause 6.1.1, the procedure related to the interconnection phase is described in clause 6.1.2, and the procedure related to the interconnection release is described in clause 6.1.1.

6.1.1 Interconnection establishment

The **interconnection establishment** is a procedure in which an uCDN and a dCDN begin with no relationship between each other and proceed by exchanging all the information needed to verify each other's identity and thus establish a secure communication channel between each other. This communication is between the ICFs of the CDNs and spans the first three steps of the procedure. The rest of the phase consists of three sub procedures in which the dCDN informs the uCDN about its capabilities, footprint and starts notifying it about its status. After the uCDN receives the first positive status exchange from the dCDN, it can safely assume that the dCDN is ready to process its requests. The conclusion of this phase is the establishment of communication between dRCF and uRCF and the whole CDN-I relationship moves to its **interconnection phase**.

The interconnection establishment is an optional procedure, which is not used in case both CDN providers agree on static manual pre-provisioning of CDN interconnection, but if dynamic interconnection establishment is supported by both CDNs it should follow one of these procedures.

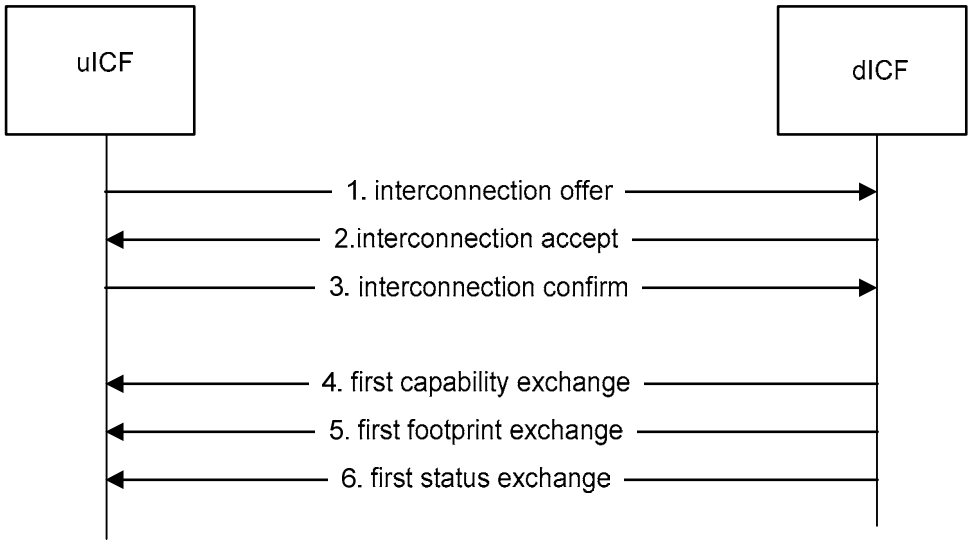

Figure 6.1.1.1: Interconnection establishment

NOTE 1: Prior to interconnection establishment procedure, IP interconnection and service level agreement between interconnected CDN providers and/or content provider are needed.

The interconnection process should begin with the **interconnection establishment**. This interconnection establishment is defined by a procedure consisting of following steps:

1) The uCDN sends an interconnection request to the dCDN. This message is sent between the ICFs of the CDNs and consists of all the information the dCDN needs to decide whether to accept or deny the establishment of peering relationship with the uCDN. This information may include uCDN's CDN identifier, authentication data, required peering parameters and others.

2) After the dCDN receives an interconnection offer from an uCDN it analyses its contents and decides whether to deny it (sending an interconnection deny message and terminating the procedure) or accept it. If it decides to accept the offer then it sends an interconnection accept message that contains information that the uCDN can use to make a final decision about establishing the peering relationship with the dCDN. This information may include dCDN's CDN identifier, authentication data, required peering parameters and others.

3) After the uCDN receives an interconnection accept from a dCDN it analyses its contents and decides whether to deny it (sending an interconnection deny message and terminating the procedure) or confirm it. If it decides to confirm it then it sends an interconnection accept message indicating that the peering can begin.

4) After the interconnection is confirmed by the uCDN then the dCDN shall begin its first capability exchange procedure.

5) After the first capability exchange procedure is finished the dCDN shall begin its first footprint exchange procedure.

6) After the first footprint exchange procedure is finished the dCDN shall begin its first status exchange procedure.

NOTE 2: This procedure may be repeated if there are multiple ICFs to be interconnected between the two CDNs.

Both the CDNs should then proceed to the interconnection phase.

6.1.2 Interconnection phase

The second phase is simply called **interconnection phase.** In interconnection phase shall be both CDNs already fully interconnected and through their interconnection interfaces ICFs and RCFs. Interconnection interfaces are fully useable by both CDNs and agreed capabilities accessible over CDN interconnection.

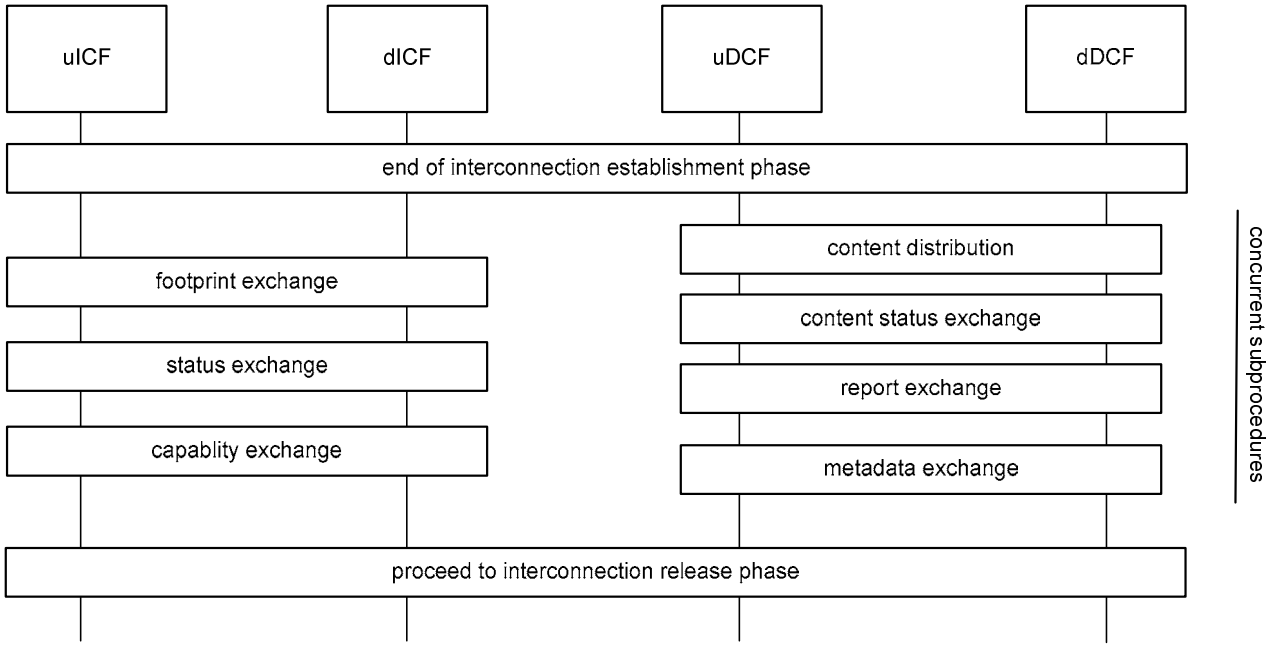

Figure 6.1.2.1: Interconnection phase

The general procedure of the interconnection phase consists of concurrent executions of multiple separate procedures. These procedures are described in other sub-clauses of this clause, beginning with clause 6.2.

6.1.3 Interconnection release

The interconnection release is a phase that begins when an uCDN or a dCDN decides to release a CDN peering relationship with each other (for a specific ICF-ICF relation, and associated RCF-RCF relations). This means that the procedure has two variants, depending on whether it was initiated by the uCDN or the dCDN. In both cases it concludes the interconnection relationship by cleanly finishing all outstanding actions between the two CDNs.

The interconnection release is an optional procedure, which is not used in case CDN providers agree on static manual pre-provisioning of CDN interconnection, but if dynamic interconnection establishment is supported by both CDNs it should follow one of these procedures.

The uCDN initiated procedure for this consists of following steps, shown in figure 6.1.3.1.

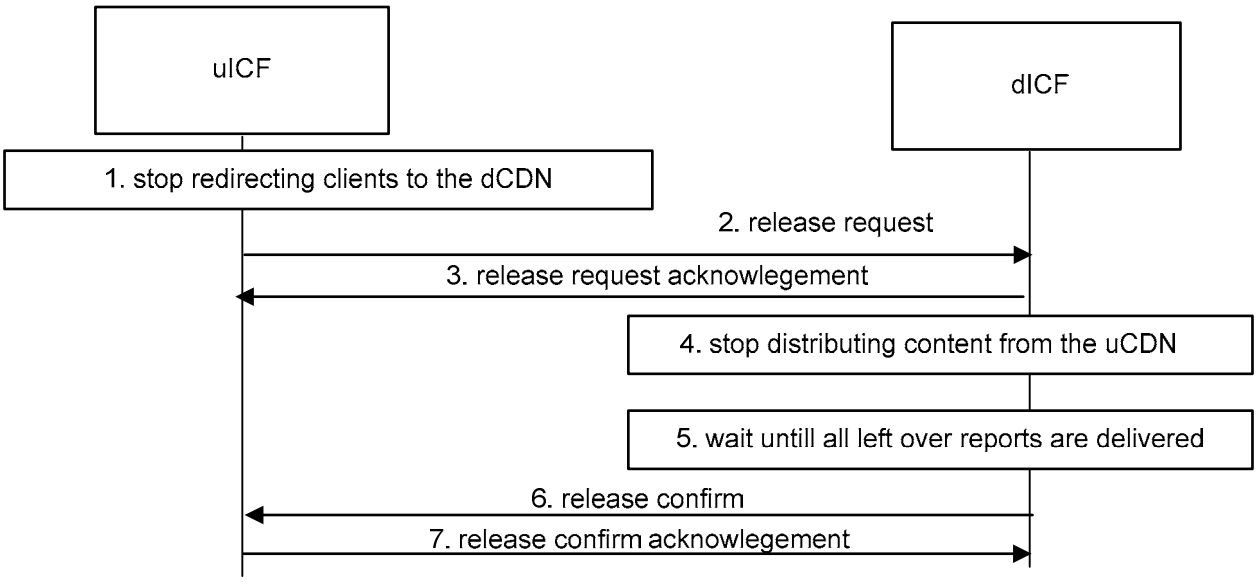

Figure 6.1.3.1: uCDN-initiated interconnection release

1) The uCDN stops redirecting clients to the dCDN.
2) After the last client redirect is finished it sends the release request message to the dCDN.
3) The dICF immediately sends an acknowledgement.
4) After receiving the release request, the dCDN also stops distributing content from the uCDN.
5) The dCDN then waits until all outstanding reports are delivered.
6) The dCDN informs the uCDN about this event using the release confirm message.
7) The uCDN sends back acknowledgment message.

After this both CDNs can fully disband the interconnection relationship and release all related resources.

The dCDN initiated procedure for this consists of following steps, shown in figure 6.1.3.2.

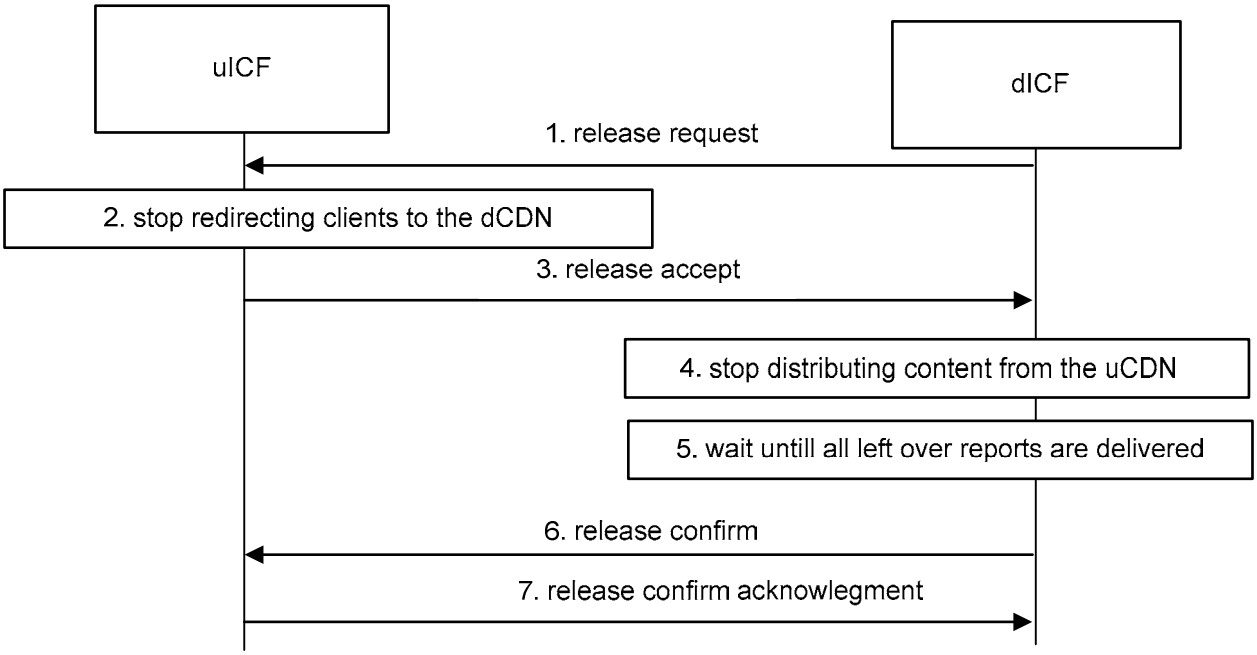

Figure 6.1.3.2: dCDN-initiated interconnection release

The dCDN initiated procedure for this consists of following steps:

1) The dICF sends a release request message to the uICF.
2) The uCDN stops redirecting clients to the dCDN.
3) After the last client redirect is finished uICF sends the release accept message to the dICF.
4) After receiving the release request, the dCDN stops distributing content from the uCDN.
5) The dICF then waits until all outstanding reports are delivered.
6) The dICF informs the uICF about this event using the termination confirm message.
7) The uCDN sends an acknowledgement.

After this both CDNs can fully disbanded the interconnection relationship and release all related resources.

6.2 Content distribution

6.2.1 General

This clause provides procedures for content distribution:

- Content distribution control, the initiation of content exchange.
- Content exchange, the actually transfer or streaming of content between the interconnecting CDNs.

6.2.2 Content distribution control

6.2.2.1 General

This clause provides procedures for content distribution control:

- Content distribution.
- Content deletion.

6.2.2.2 Upstream-initiated content distribution

Figure 6.2.2.2.1 shows the procedure for upstream-initiated content distribution from the uCDN to the dCDN. The two CDNs should be interconnected.

NOTE 1: Upstream-initiated content distribution is typically used for the prepositioning of files or streams.

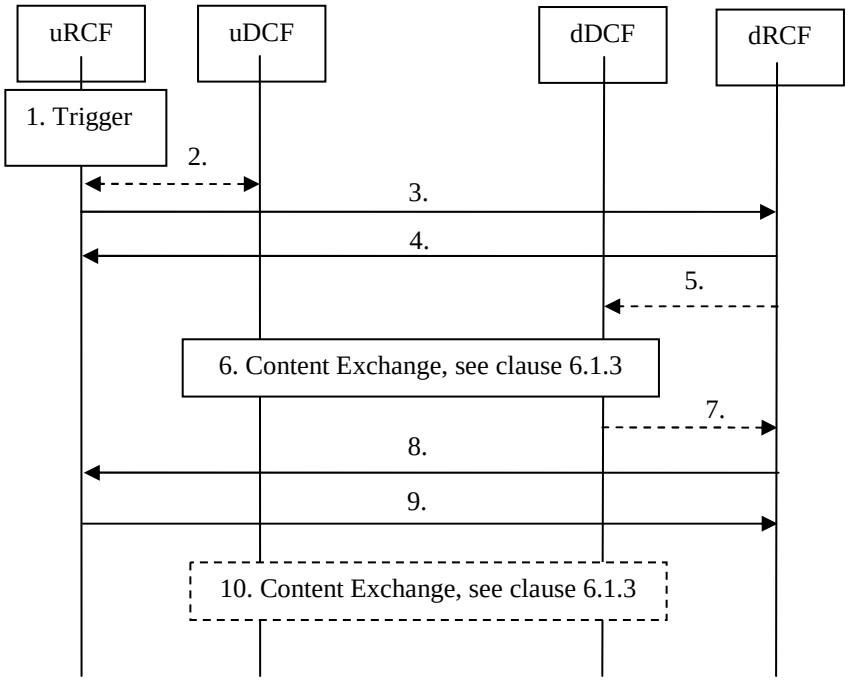

Figure 6.2.2.2.1: Upstream-initiated content distribution

The procedure has the following steps:

1) The uRCF is triggered to initiate content distribution. It selects a dCDN and makes sure that the capabilities of the dCDN match with the capability requirements of the content that the uCDN wants to distribute.

2) The uRCF selects a uDCF. It may communicate about this with the selected uDCF.

NOTE 2: The communication between uRCF and uDCF is CDN internal. It is not specified in the present document.

NOTE 3: The uDCF may be located in the Upstream CDN domain or in the Content Provider domain. The latter option saves resources in the Upstream CDN domain.

3) The uRCF sends a request to the dRCF for content distribution. The request contains the contentID of the content item that is to be distributed, the address of the selected uDCF and optionally other information, like a token for authentication purposes, or a DRM flag to indicate that DRM procedures apply (see also clause 6.6.2).

4) The dRCF returns a response, acknowledging the request. The response should contain an address or URI to which future UE request for the content item should be redirected. The dRCF could also reject the request.

5) The dRCF selects one or more dDCFs. It instructs the selected dDCF(s) to perform a content exchange.

NOTE 4: The communication between dRCF and dDCF is CDN internal. It is not specified in the present document.

6) The selected dDCF performs a content exchange with the selected uDCF. The content exchange is specified in clause 6.2.3.

7) The dDCF informs the dRCF when the content exchange has completed.

NOTE 5: The communication between dDCF and dRCF is CDN internal. It is not specified in the present document.

8) The dRCF informs the uRCF that the content is available for delivery by the Downstream CDN.

9) The uRCF returns an acknowledgement.

10) There may be subsequent content exchanges for the identified content item, where other DCF(s) retrieve the same content item.

After step 9, the uCDN can start directing UE requests to the address or URI provided in step 4.

NOTE 6: Communication with the UE is not described in the present document.

NOTE 7: This procedure can be used in a way that the dCDN retrieves a content item only once, and it takes care of its internal distribution. This procedure can also be used in a way that the same or other dDCFs retrieve a content item multiple times, e.g. because of implemented caching strategies (least recently used, least frequently used, etc).

NOTE 8: The dCDN can hide its internal topology towards the uCDN by always using the same limited group of dDCFs to retrieve content items from the uCDN, and performing further content distribution within the dCDN internally. Similarly, the uCDN can hide its internal topology by always using the same limited group of uDCFs to distribute content items to the dCDN.

6.2.2.3 Downstream-initiated content distribution

Figure 6.2.2.3.1 shows the procedure for downstream-initiated content distribution from the uCDN to the dCDN. The procedure is used when the dCDN needs to know from where to retrieve an identified content item from the uCDN.

NOTE 1: Downstream-initiated content distribution is typically used for cases where the distribution of content files or streams between the two CDNs is postponed until there is an actual user request.

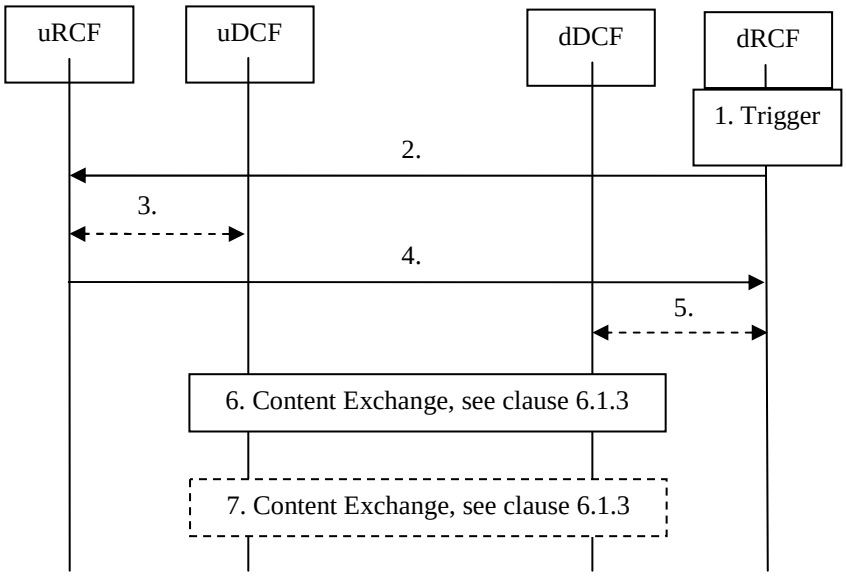

Figure 6.2.2.3.1: Downstream-initiated content distribution

The procedure has the following steps:

1) The dRCF is triggered to initiate content distribution for an identified content item.

2) The dRCF sends a request to the uRCF. There includes the contentID of the content item.

3) The uRCF selects a uDCF. It may communicate about this with the selected uDCF.

NOTE 2: The communication between uRCF and uDCF is CDN internal. It is not specified in the present document.

4) The selected uRCF responds to the dRCF. The response includes the location of the selected uDCF and optionally other information, like a token for authentication purposes, or a DRM flag to indicate that DRM procedures apply (see also clause 6.6.2).
The uRCF could also reject the request.

5) The dRCF selects a dDCF and triggers it for a content exchange.

NOTE 3: The communication between uRCF and uDCF is CDN internal. It is not specified in the present document.

6) The content exchange is specified in clause 6.2.3.

7) There may be subsequent content exchanges for the identified content item, where other DCF(s) retrieve the same content item.

After step 6, the dCDN can start delivering the content item to Ues that have been redirected to the dCDN.

NOTE 4: Communication with the UE is not described in the present document.

6.2.2.4 Upstream-initiated content deletion

Figure 6.2.2.4.1 shows the procedure for upstream-initiated content deletion in the dCDN.

NOTE 1: Upstream-initiated content deletion is typically performed at the request of a Content Provider to make sure that content is no longer available after a point in time.

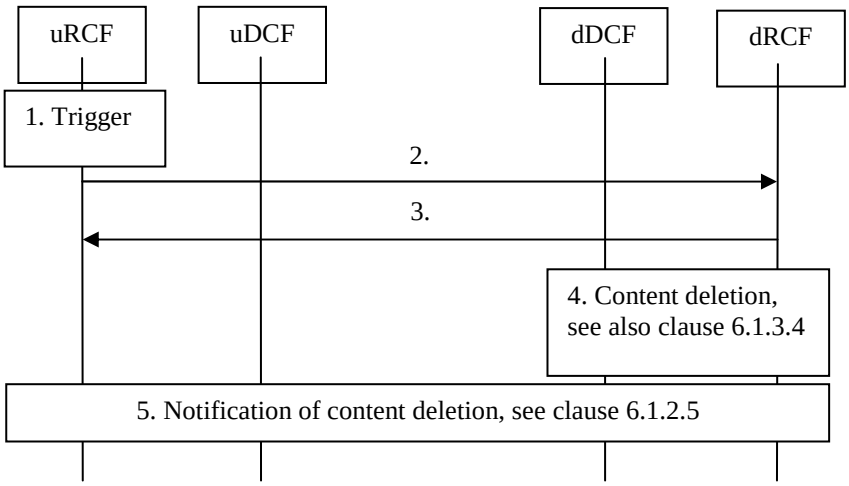

Figure 6.2.2.4.1: Upstream-initiated content deletion

The procedure has the following steps:

1) The uRCF is triggered to initiate content deletion. It should stop the request routing for the content item to the dCDN.

2) The uRCF sends a request to the dRCF to delete content. The request includes the contentID of the to-be-deleted content and a time-out parameter.

3) The dRCF returns an acknowledgement.

4) There are dCDN-internal communications and action to delete the identified content item. The dCDN should stop accepting new delivery requests for the content item. After the time-out, specified by the time-out parameter, it should actively stop/release all still on-going deliveries of the content item. If the content item is a stream, then clause 6.2.3.4 applies.

NOTE 2: The dCDN-internal communications and actions are not specified in the present document.

5) The dRCF conforms the successful deletion of the content to the uRCF, see clause 6.2.2.5.

After completion of this procedure, there shall be no further content exchanges for the deleted content.

6.2.2.5 Downstream notification of content deletion

Figure 6.2.2.5.1 shows the procedure for downstream notification of content deletion.

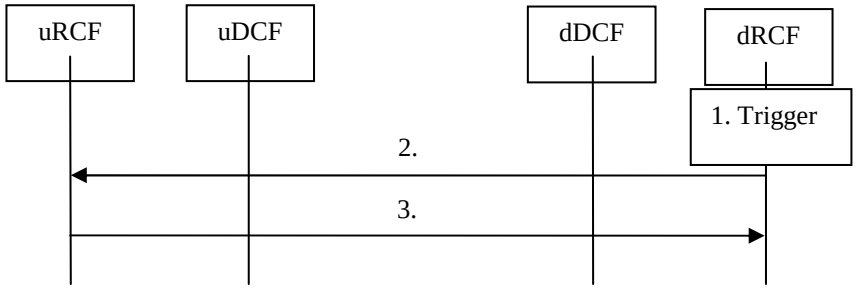

Figure 6.2.2.5.1: Downstream notification of content deletion

The procedure has the following steps:

1) The dRCF is triggered to send a notification of content deletion. The dRCF may be triggered when all instances of a particular content file or stream are deleted in the dCDN. The dRCF shall be triggered if the content deletion is the result of an upstream-initiated content deletion as specified in clause 6.2.2.4.

2) The dRCF sends a notification of content deletion to the uRCF. The notification includes the contentID of the deleted content item.

3) The uRCF returns an acknowledgement.

6.2.3 Content exchange

6.2.3.1 General

This clause provides procedures for content exchange. The following types of content may be exchanged:

- Files.
- Streams.
- Segmented content.

6.2.3.2 File transfer

Figure 6.2.3.2.1 shows the content-exchange procedure for file transfer.

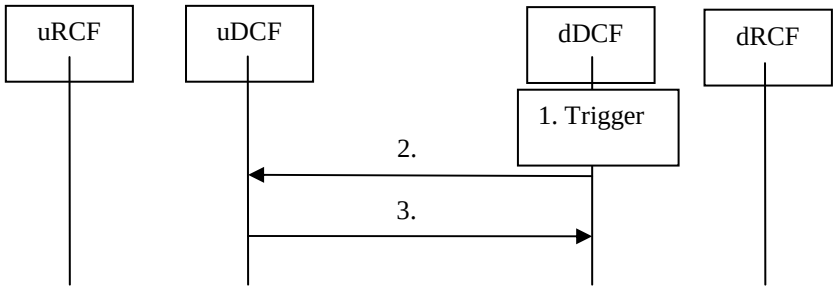

Figure 6.2.3.2.1: File transfer

The procedure has the following steps:

1) The dDCF is triggered retrieve a file. The trigger includes information that enables the dDCF to identify and locate a selected uDCF.

2) The dDCF sends a retrieval request to the uDCF.

3) The uRCF authenticates the request and returns the requested file.

NOTE: There may be more messages exchanged between the dDCF and uDCF, depending on the used protocol(s).

6.2.3.3 Stream set-up

Figure 6.2.3.3.1 shows the content-exchange procedure for stream set-up.

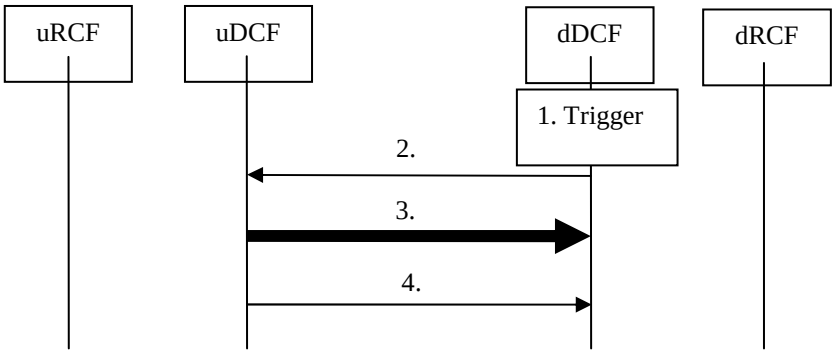

Figure 6.2.3.3.1: Stream set-up

The procedure has the following steps:

1) The dDCF is triggered to set up a stream. The trigger includes information that enables the dDCF to identify and locate a selected uDCF.

2) The dDCF sends a request to the uDCF to set up the stream. The request includes the contentID of the requested stream.

3) The uDCF authenticates the request and starts sending the requested stream to the dDCF.

4) The uDCF confirms to the dDCF that the stream has been set-up.

NOTE: There may be more messages exchanged between the dDCF and uDCF, depending on the used protocol(s).

6.2.3.4 Stream release

Figure 6.2.3.4.1 shows the content-exchange procedure for stream release.

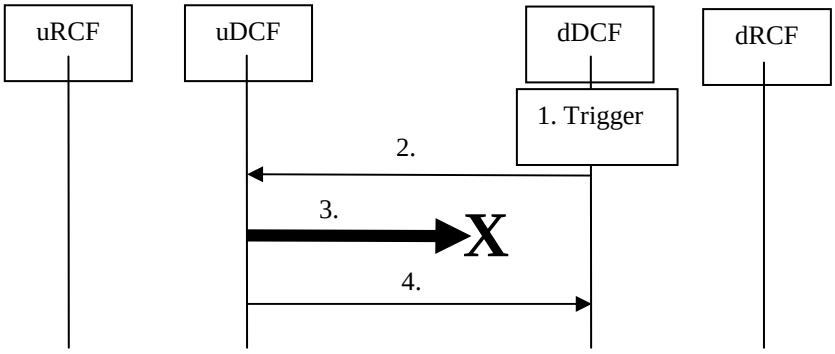

Figure 6.2.3.4.1: Stream release

The procedure has the following steps:

1) The dDCF is triggered to release a stream.

2) The dDCF sends a request to the uDCF to release the stream.

3) The uRCF stops sending the stream to the dDCF.

4) The uRCF confirms to the dDCF that the stream has been stopped.

NOTE: There may be more messages exchanged between the dDCF and uDCF, depending on the used protocol(s).

6.2.3.5 Segmented content

Segmented content is content composed of multiple files, or content composed of multiple streams, or content composed of one or more files and one or more streams. The composition of segmented content is described in an associated manifest file (MF).

NOTE 1: The term "manifest file" has several synonyms, depending on the technology used. MPEG DASH [i.4], for example, uses the term Media Presentation Description (MPD).

Figure 6.2.3.5.1 shows the content-exchange procedure for segmented content.

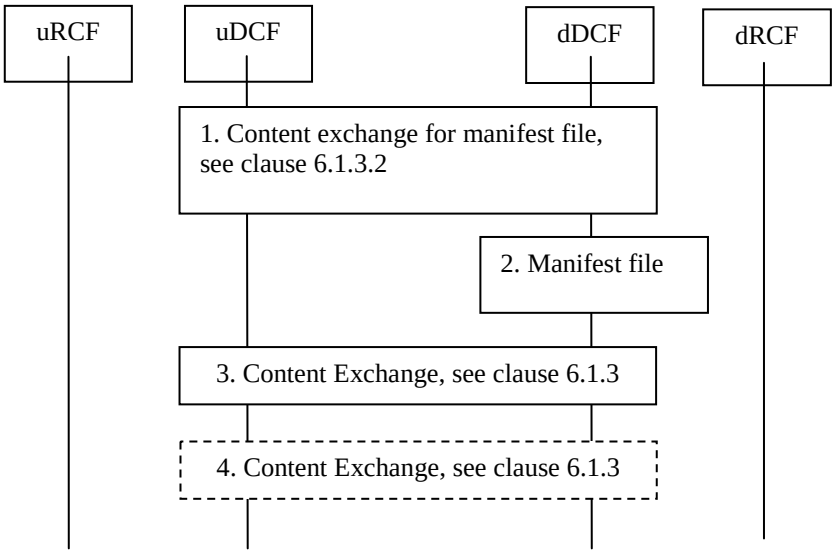

Figure 6.2.3.5.1: Segmented content

The procedure has the following steps:

1) The dDCF retrieves an MF from the uDCF, using the procedure of clause 6.2.3.2. The uCDN may decide to keep some URLs in the MF empty, in order to prevent the dCDN from retrieving the associated segments.

NOTE 2: There may be several reasons why an uCDN would want to do this. If the dCDN is a mobile CDN, then it makes no sense to populate the dCDN with high-bitrate high-definition (HD) segments, that would never be delivered by the dCDN anyway. There may be cost considerations, where the uCDN would outsource the delivery of popular segments to the dCDN, and deliver the less popular segments from the uCDN itself. Especially for video clips, the early parts are much better watched than the later parts. Finally, the uCDN may want to deliver some segments itself for monitoring and logging purposes.

NOTE 3: An MF with some empty URLs would typically only be used for content distribution and not be delivered to UEs.

2) The dDCF parses the manifest file and decides to initiate further content exchanges to retrieve the identified segments.

3) Further content exchange, see clause 6.2.3.

4) Optional further content exchange(s), see clause 6.2.3.

NOTE 4: Whereas this clause shows how an MF (Manifest File) is distributed as a special type of content item, the MF could also be handled in a different way depending on bilateral agreements between the uCDN and dCDN. For example, depending on the type of MF (containing relative URLs, absolute URLs with redirection, or absolute URL without redirection), the MF could be distributed like any generic content item, the MF could be distributed like a special type of metadata, or the MF is not distributed to the dCDN at all.

6.3 Request routing

6.3.1 General

This clause provides procedures for request routing between two interconnected CDNs.

6.3.2 Single-delivery request routing

Figure 6.3.2.1 shows the procedure for request routing for a single content-item delivery.

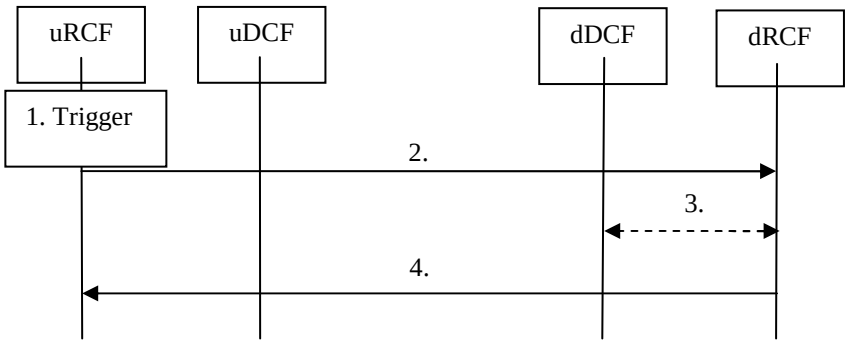

Figure 6.3.2.1: Single-delivery request routing

The procedure has the following steps:

1) The uRCF is triggered to initiate request routing, e.g. by an incoming content delivery request from a User Equipment (UE).

2) The uRCF send a request to the dRCF, with a contentID and optionally information (e.g. location or IP address) of the UE to which the identified content is intended to be delivered.

3) The dRCF may select a dDCF. It may communicate about this with the selected uDCF.

NOTE 1: The communication between dRCF and dDCF is CDN internal. It is not specified in the present document.

4) The dRCF sends a response to the uRCF, with the dDCF address to which the UE content delivery request should be redirected.

NOTE 2: No assumptions are made how that redirection takes place in the Upstream CDN domain. It may use HTTP redirect, DNS response, combination of both, or other.

6.3.3 Multiple-delivery request routing

Figure 6.3.3.1 shows the procedure for request routing for multiple content items simultaneously. This procedure increases efficiency when there is a high intensity of request routing between the uCDN and dCDN. A specific case is segmented content, where the uCDN wants to fill-in a manifest file (MF) for directing individual segment requests directly to the proper dDCF. As the resulting MF, filled-in by the uCDN, is typically relatively small and UE-specific, it could be delivered to the UE by the uCDN directly, instead of having the MF delivered by the dCDN.

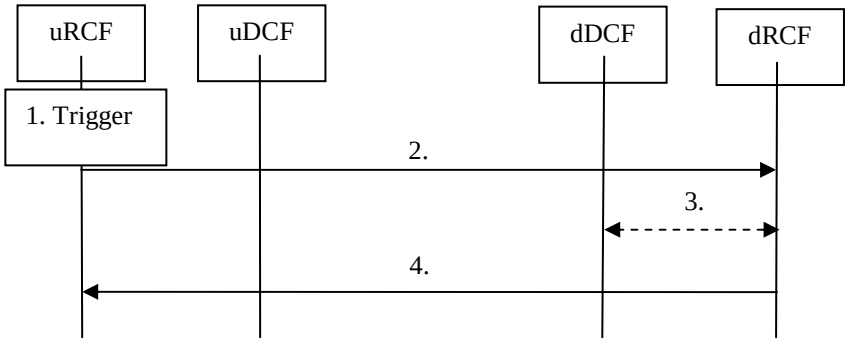

Figure 6.3.3.1: Multiple-delivery request routing

The procedure has the following steps:

1) The uRCF is triggered to initiate request routing, e.g. by an incoming content delivery request from a User Equipment (UE) for segmented content.

2) The uRCF sends a request to the dRCF, with multiple contentIDs and optionally information (e.g. location or IP address) of the UE to which the identified content items are intended to be delivered.

3) The dRCF may select one or multiple dDCF(s) for the content delivery. It may communicate about this with the selected uDCF.

NOTE: The communication between dRCF and dDCF is CDN internal. It is not specified in the present document.

4) The dRCF sends a response to the uRCF, with the dDCF address(es) to which the UE content delivery request should be redirected for the different identified content items. The uCDN can use this information for UE redirection and/or for filling in an MF.

6.4 Reporting

6.4.1 General

The reporting between CDNs is realized by two different procedures. The differences between them are explained below:

- **Upstream-initiated reporting** is used by the upstream CDN to retrieve any information from the downstream CDN. This information may include up-to date delivery statistics and status , comprehensive historical logs or other information. The structure of these reports is not strictly defined by the present document. It only provides a way for the uCDN to inform the dCDN about the kind of data that it requests.

- **Downstream-initiated reporting** is used when a downstream CDN needs to immediately inform the upstream CDN about a recent event related to a specific content item. The format and contents of these reports are strictly defined by the present document.

6.4.2 Upstream-initiated reporting

The main role of the upstream-initiated reporting procedure is to make it possible for the uCDN to get access to the content-related information gathered by the dCDN while delivering the content to clients. This procedure shall be initiated by the uRCF to inform the dRCF about the fact that the uRCF requests specific information. The request sent by the uRCF shall include the description of the type of information required. Depending on the nature of the information the dRCF shall either respond either by directly returning the requested information or it shall return a URL that the uRCF will be able to use to access the requested information. This procedure is visualized in figure 6.4.2.1.

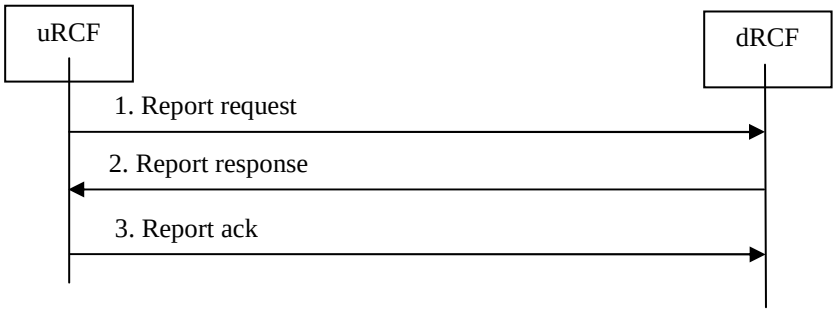

Figure 6.4.2.1: Upstream-initiated reporting procedure

The procedure consists of following steps:

1) The uRCF sends a report request to the dRCF. This request indicates the description of information that the uRCF requires.

2) The dRCF shall respond to the uRCF with either the required information, or information about where it can be obtained.

3) The uRCF shall acknowledge the reception of the response.

6.4.3 Downstream-initiated reporting

The role of the downstream-initiated reporting procedure is to inform the uCDN about a content-related event that happened within the dCDN. This procedure is visualized in figure 6.4.3.1.

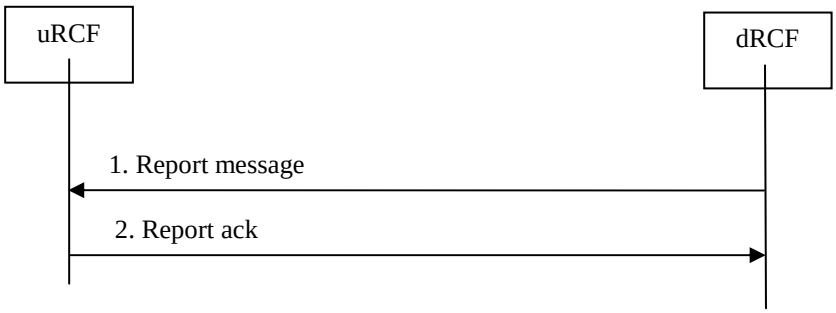

Figure 6.4.3.1: Downstream-initiated reporting procedure

The procedure consists of following steps:

1) The dRCF sends a report message to the uRCF.

2) The uRCF shall acknowledge the reception of the report message.

6.5 Interconnection Control Function Procedures

6.5.1 General

These procedures allow to inform the uCDN about changes of specified capabilities/footprint/status of a dCDN. The information contained in these updates shall match the data models defined in clause 7. Examples of specific messages can be found in annex B.

6.5.2 Capabilities exchange

The capability exchange is a procedure whose main goal is to inform the uCDN about the capabilities of a dCDN. The capability list is a data structure describing the list of capabilities (or features, services and their parameters) that the dCDN is willing to provide to uCDN.

The capability exchange procedure is primarily used to exchange information about CDN capabilities over CDN interconnection. It can also be used for exchanging information about capabilities related to content delivery. These capabilities description are used to achieve interoperability between the CDNs compliant with the present document and enable interconnection with other CDNs that support just a subset or different sets of capabilities (e.g. comply only with basic capabilities specified in the IETF CDN-I standards [i.2]). If there is a CDN that does not support one of the methods of content delivery for example HTTP, it will still be able to interconnect and provide the delivery of content using other delivery methods that it does support, and will just need to communicate its capabilities over the capability exchange. For this reason CDNs compliant with the present document shall understand the meaning of the identifiers that represent these capabilities.

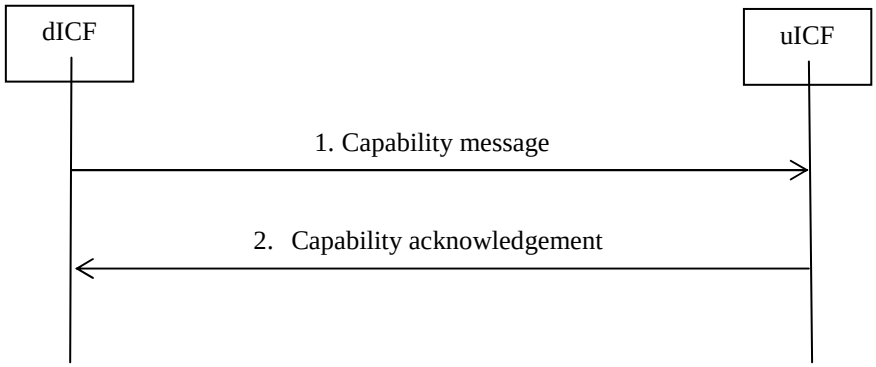

Figure 6.5.2.1: Capability Exchange

The capability exchange procedure is always initiated by the dCDN sending information about its capabilities to the uCDN. It consists of following steps:

1) The dCDN sends a capability message to the uCDN. This message contains the list of capabilities describing the dCDN. Examples of possible capabilities can be found in clause B.1 Metadata structure.

2) The uCDN acknowledges the reception of the capability message by sending a capability acknowledgement.

6.5.3 Footprint exchange

The footprint exchange is a procedure whose main goal is to inform the uCDN about the footprint of a dCDN. The footprint is a data structure containing a list of network segments for which the dCDN can deliver content items for the uCDN.

There are multiple methods that can be used to describe a network segment. The most basic one is an IP prefix. An IP prefix is a network number accompanied by a prefix length number. Geolocation, AS numbers and others are also viable candidates for network segment definitions. The basic rule is that any segment description can be used as long as there is a definitive agreement between the interconnected CDNs about how to map them to IP addresses.

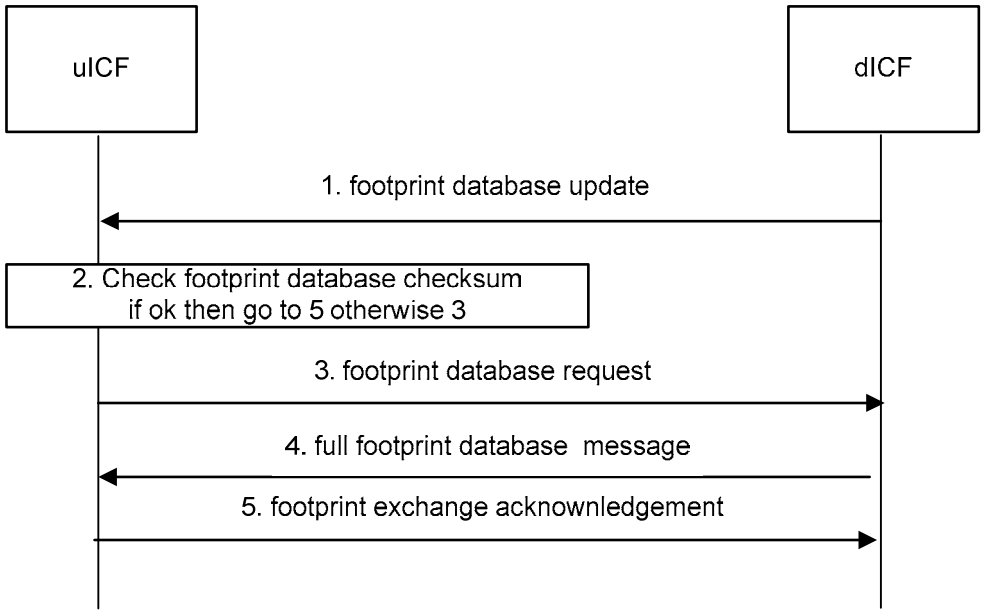

Figure 6.5.3.1: Footprint Exchange

The footprint exchange procedure consists of following steps:

1) The dCDN sends a footprint database update to the uCDN. This message is triggered by a change of the footprint that dCDN wants to serve for uCDN. The update message contains only the changes from the previous version of the database and a checksum of the full database it is supposed to converge to after merging with previous version.

2) The uCDN merges the update with its footprint database related to the dCDN, calculates its checksum and compares it to a checksum received as part of the update message. If the checksum is ok then it skips over to step 5, otherwise it proceeds on step 3.

3) The uCDN requests the full footprint database from the source CDN. The full footprint database contains the whole data structure containing information about dCDN's footprint.

4) The dCDN sends the full footprint database to the uCDN.

5) The uCDN confirms the reception of the footprint update or database by sending a footprint exchange acknowledgement message.

The footprint exchange process can start on step 1, 3 or 4 of the described procedure. If the process is triggered by a change in dCDN footprint database then it starts on step 1. If the uCDN triggers the process then it starts on step 3. If the process is triggered as the first footprint exchange then it starts on step 4.

6.5.4 Services status exchange

The status exchange is a procedure whose main goal is to inform the uCDN about the status of a dCDN. This procedure is specific in the fact that it can be triggered not only in reaction to a change of status in the dCDN but is also used as a keep alive mechanism. This means that even when the dCDN's status did not change it has to send a status message within a defined amount of time to let the uCDN know that the status is still valid.

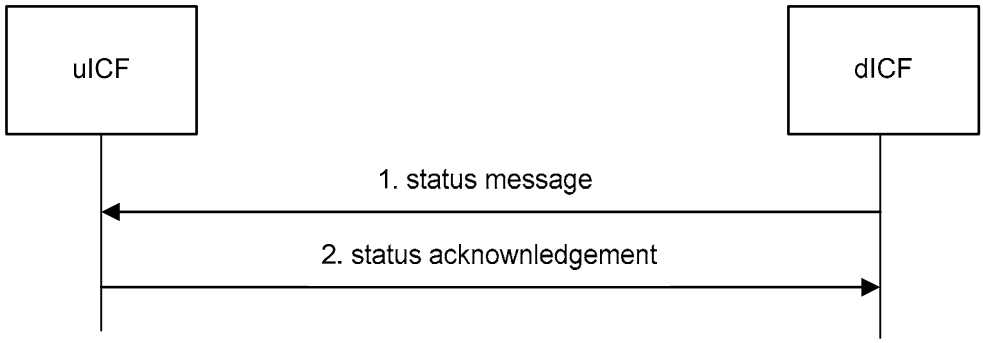

Figure 6.5.4.1: Service Status Exchange

The status exchange procedure is always initiated by the dCDN sending information about its status to the uCDN. It consists of following steps:

1) The dCDN sends a status message to the target CDN.

2) The target CDN acknowledges reception of the status message by sending a status acknowledgement.

6.6 DRM Procedures

6.6.1 General

These procedures are used when the CDN interconnection is involved in the DRM for content delivery.

6.6.2 Flagging CDN content for DRM

An uCDN can flag to a dCDN that DRM procedures should be applied for a specific content item. This is flagged during content distribution.

In case of upstream-initiated content distribution, step 3 of clause 6.2.2.2 includes that DRM flag.

In case of downstream-initiated content distribution, step 4 of clause 6.2.2.3 includes that DRM flag.

6.6.3 Key exchange for DRM

These procedures are relevant if the dCDN would need to encrypt a content item for delivery to a specific UE. Different encryption for different UE provides an additional layer of protection against signal theft. The content item could be stored in plain format. There exist also DRM systems that enable the content to be stored in encrypted format, and where a key is used to re-encrypt or partially decrypt it into the UE-specific encryption without fully decrypting the content in the process. This protects the content item against content theft at the CDN.

6.6.3.1 uCDN-initiated key exchange for DRM

Figure 6.6.3.1.1 shows the procedure for uCDN-initiated key exchange for DRM. It may be combined with the single-delivery request-routing procedure of clause 6.3.2.

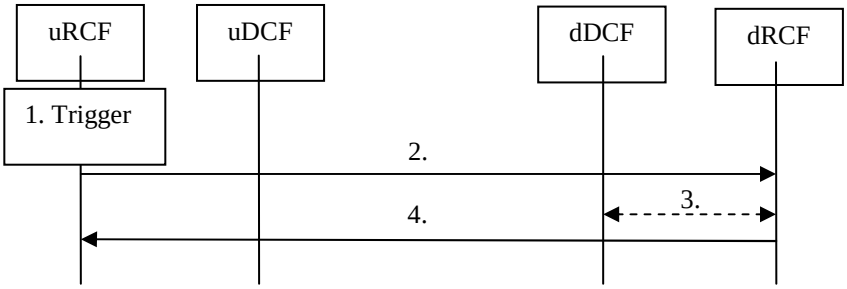

Figure 6.6.3.1.1: uCDN-initiated key exchange for DRM

The procedure has the following steps:

1) The uRCF is triggered to perform a key exchange procedure. The trigger may be a delivery request from a UE for DRM-protected content.

2) The uRCF sends a key exchange message to the uDCF. This message contains a DRM-instruction identifier, a DRM key, a contentID identifying the content item and an identifier of the UE and/or transaction (e.g. IP address or token). The semantics of the DRM instruction identifier are bilaterally agreed between the uCDN and the dCDN. It is used to instruct the dCDN which DRM system and DRM procedures need to be applied.

3) The dRCF may communicate with one or more dDCF to pass the received information.

NOTE: The communication between dRCF and dDCF is CDN internal. It is not specified in the present document.

4) The dRCF returns an acknowledgement.

As the triggers and procedure are similar with single-delivery request routing (clause 6.3.2), the two procedures may be combined.

6.6.3.2 dCDN-initiated key exchange for DRM

Figure 6.6.3.2.1 shows the procedure for dCDN-initiated key exchange for DRM. If the content distribution to the dCDN has not yet taken place, then this procedure may be combined with the procedure for downstream-initiated content distribution in clause 6.2.2.3.

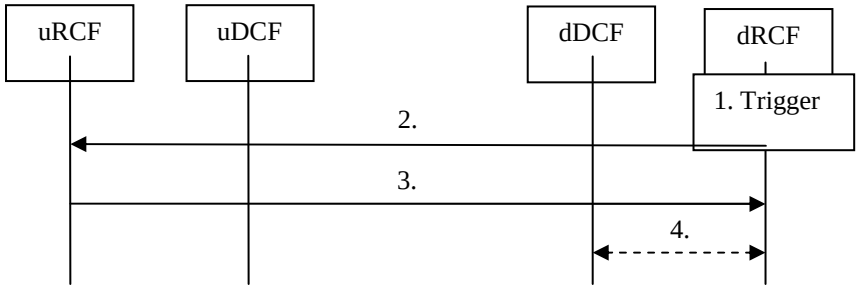

Figure 6.6.3.2.1: dCDN-initiated key exchange for DRM

The procedure has the following steps:

1) The dRCF is triggered to perform a key exchange procedure. The may be a delivery request from a UE for DRM-protected content.

2) The dRCF sends a request message to the uRCF. This message contains a contentID identifying the content item and an identifier of the UE and/or transaction (e.g. IP address or token).

3) The uRCF responds with a message that contains a DRM key and optionally a DRM-instruction identifier. A DRM-instruction identifier would be passed, if it cannot be derived from the UE request. UE requests typically contain tokens for authentication purposes, and a DRM instruction could also have been encoded in the token.

4) The dRCF may communicate with one or more dDCF to pass the received information.

NOTE: The communication between dRCF and dDCF is CDN internal. It is not specified in the present document.

7 Data models

There are two categories of data models.

1) CDN related data models

   This category of data represents the information about a specific CDN network from the interconnection perspective.

   - **CDN information** contains basic data that is relevant to specific CDN.
   - **CDN footprint** defines a single footprint segment that is related to a specific CDN.
   - **CDN capabilities** describe the capabilities of a specific CDN within a specific footprint segment.

2) Content related data models

   This category describes the data models relevant to specific content items and their delivery.

   - **Content related metadata data model** includes all the basic information describing the content.
   - **Content distribution reporting data model** includes all the data related to the distribution status of a specific content.
   - **Content request source data model** includes all the data related to a source of content delivery request.
   - **Content delivery reporting data model** includes all the data related to delivery of a specific content.

Annex D provides examples of such data models.

NOTE: There were no IETF-specified inter-CDN data models available at the time of writing the present document.

8 Security

This clause specifies all the security aspects used for CDN interconnection. Their main purpose is to make it provide a complete tool-set that can be used to make CDN interconnection into a secure environment while maintaining scalability and flexibility.

Scalability is guaranteed by adhering to the architectural principles defined by the high-level architecture as described in clause 5 of the present document. It is also reinforced by the use of encryption protocols and techniques that can be easily accelerated by available hardware.

Flexibility is achieved by making all the elements of CDN interconnection security optional and by not using any proprietary protocol or complicated solutions. All the technologies used are usually already implemented in production CDN environments so they will not have to be re-engineered for CDN-I environments.

8.1 Security feature interoperability

Because all of the security features in CDN-I environments are optional it is important to define the means of interoperability between CDNs with different CDN-I security feature sets.

This kind of interoperability is achieved by the means of capabilities, see clause 6.5.2. This means that every security feature is represented by a corresponding capability that is advertised to peer CDNs every time a CDN interconnection relationship is established.

8.2 CDN interconnection service protection

The security of communication between CDN-I entities, as specified in clause 6, is achieved by using the SSL/TLS protocol to encrypt and authenticate all communication channels. In most scenarios this consists of using the HTTPS protocol instead of plain HTTP for CDN-I procedure messages. In some cases it may also mean using secure versions of content distribution protocols.

The authenticity of peer party can be always reliably verified by checking the security certificate at the time of connection set-up. If the certificate is neither known nor signed by a trusted certification authority then the connection should not be established. If it is established, then it shall be considered untrustworthy.

The method of establishing secure connection between two CDNs is described in clause 8.2.1.

8.2.1 Secure CDN-I connection establishment

The establishment of a secure CDN-I connection should begin by trying to connect to the Interconnection Control Interface of a peer CDN via the HTTPS protocol, using procedures described in clause 6.1.1. If the connection can be established and the certificates are valid and considered trustworthy, then the connection between the interconnection control entities is assumed to be trusted.

If it is not possible to connect to the interconnection control interface via HTTPS, then HTTP shall be used instead and the connection shall be considered untrusted. If the HTTPS connection can be achieved but the certificate cannot be verified then it can still be used but shall also be considered untrusted. The decision about whether interconnection establishment may continue even when the connection between interconnection control entities is untrusted shall be made according to a local security policy of a particular CDN (not defined by the present document). TLS/SSL contains the option of no encryption algorithm (NULL). This option shall not be used within the scope of the present document.

After the Interconnection Control entities are successfully interconnected, a capability exchange takes place, see clause 6.5.2. If the capabilities of a peer CDN indicate that it can use trusted connection for communication between RCFs or DCFs then these may be bootstrapped using secured connections. If all such connections between two CDNs are established via secured protocols and all the certificates are valid then the communication between two CDNs shall be considered trusted. Secure CDN interconnection may be a requirement of a security policy of a specific CDN. It may also be a requirement of the content provider. In such scenario the content provider shall indicate this requirement by the means of a metadata parameter indicating that a specific content may only be distributed between CDNs that are interconnected via secure and trusted connections.

8.3 CDN interconnection content and metadata authenticity

The authenticity of content and its metadata shall be achieved by protecting integrity by means of the security certificates. Content and or its metadata can be signed with a private key belonging to the content provider. The certificate and checksum generated for authenticity, generated with the use of the content provider's private key, shall be included as a part of the metadata file. Any CDN that requires verifying the authenticity of a piece of content or its metadata should have access to an authority where it can verify the certificate included in the metadata file. An example of such authority may be the DNSSEC system. It may also suffice for the certificate to be signed by the uCDN's private key (which should already be trusted at time of such check).

8.4 Security policy definition by content provider

It shall be possible for the content provider to define rules related to the security requirements of the content they want to provide. These rules should be indicated within the metadata file accompanying the content. The content provider should guarantee the authenticity and integrity of this metadata file by signing it with its private key and publishing its corresponding public key in a certificate so that it can be used for verification.

Annex A (informative): Interfaces and Functions

This informative annex provides insight of CDN interconnection interfaces under study by other SDOs and projects. It also provides an initial list of functions that should be supported by the CDN interconnection architecture.

A.1 CDN interconnect interfaces under study

CDN interconnection is being studied by several SDO and projects. This clause provides insight of CDN interconnection interfaces under study by those SDOs and projects.

A.1.1 ETSI CDN

Figure A.1.1.1 shows the reference points of the ETSI CDN architecture (TS 182 019 [i.6]).

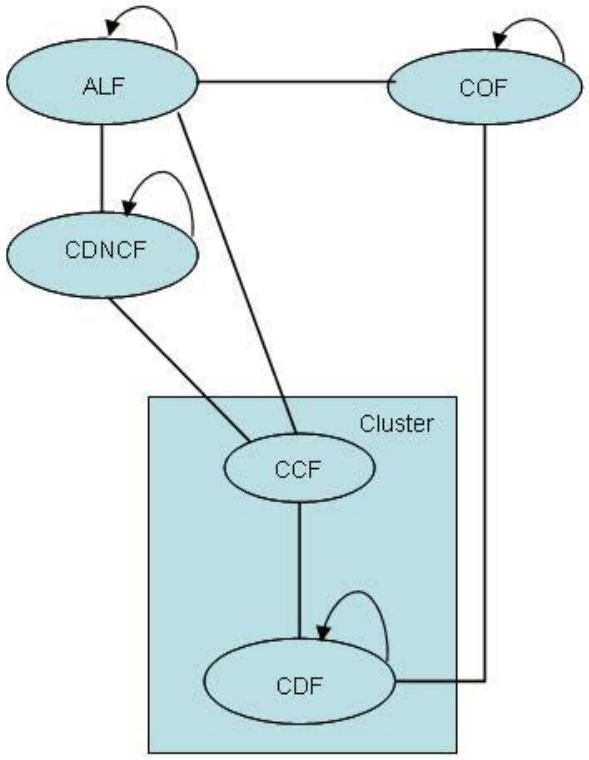

Figure A.1.1.1: Reference points in ETSI CDN

The following are the horizontal interfaces that may be relevant for CDN interconnection:

- ALF - ALF (Qq'). The Asset Location Function is a functional entity having the knowledge of the available content, the content location and others content parameters. The Qq' reference point between ALFs allows one ALF to query another about the addresses having the requested content. It can be considered a subset of the Qq reference point between CDNCF and ALF. This reference point may exist between two ALFs belonging to different CDNs.

- CDNCF - CDNCF (Yq). The Content Delivery Network Controller Function the function which manages one or more [CDN] clusters. The Yq reference point is used to allow a CDNCF to proxy a request to another CDNCF for handling. This reference point may exist between two CDNCFs belonging to different ETSICDNs.

- CDF-CDF (Cf). Tasks of Content Delivery Function are a.o. handling content delivery. The Cf reference point between CDF and CDF allows delivering content between the two CDFs for content distribution. The CDF is always instructed where to go to acquire content. This reference point may exist between two CDFs belonging to different ETSI CDNs.

A.1.2 IETF CDN-I

A.1.2.1 General

The published CDN Interconnection Problem Statement document RFC 6707 [i.2] provides a rationale for CDN interconnection, CDN-I terminology and a basic architecture outline. That outline provides the following four interfaces:

- CDN-I Control Interface
- CDN-I Request Routing Interface
- CDN-I Logging Interface
- CDN-I Metadata Interface

The **CDN-I Control Interface** is used primarily for bootstrapping of all the other interfaces and exchange of the static information related to the CDN. It also includes some of the basic content control functions of the interconnection. Some of these include content purge requests and also the initiation of a content pre-positioning process.

The **CDN-I Request Routing Interface** is responsible for ensuring that all user requests can be redirected to the most appropriate node of the CDN federation. Other than the primary role of handling client request redirects, it also handles the exchange of CDN-I information needed to properly make the request routing decisions.

The **CDN-I Metadata** Interface is responsible for proper distribution of content related metadata. This metadata may contain informative data related to the content but also technical rules that should be considered when the files are being delivered.

The **CDN-I Logging Interface** is responsible for correct distribution of any logs and reports related to the delivery of the client from the dCDN to the client.

A.1.2.2 IETF CDN-I compatibility with ETSI CDN-I

This clause discusses compatibility between RFC 6707 [i.2] and the present document. As both the IETF specifications on CDN interconnection, and the ETSI ones are work-in-progress at the time of writing the present document, all statements in the present clause about mutual compatibility have an ephemeral nature.

At the architectural level, the following mappings can be made.

- The CDN-Ic reference point for interconnection control from the present document seems to map on the union of the CDN-I Control Interface and the CDN-I Logging Interface from RFC 6707 [i.2].
- The CDNr reference point for request routing and content distribution control from the present document seems to map on the CDN-I Request Routing Interface and CDN-I Metadata from RFC 6707 [i.2].

There are some mismatches, as the present document distinguishes content-item specific status reporting from generic logging, whereas RFC 6707 [i.2] has a single logging interface.

There are differences in terminology. The term "metadata" in RFC 6707 seems to correspond to [data for] "content distribution control" in the present document. The term "acquisition" in RFC 6707 [i.2] seems to correspond to "content exchange" in the present document.

A.1.3 ATIS CSF

ATIS document 0200003 [i.3] provides CDN interconnection use case specification and high-level requirements. The document contains a picture showing the following interfaces between two carrier CDNs:

- Operations & Customer Care: SLA/outages/ticketing/Special customer requests.
- Back-Office: Provisioning, Logs, settlements.
- Routing: Traffic distribution, load management, AMT Relay address.
- Delivery: Features, Capacity reservation, Origin access, multicast sources/groups.
- Network Interconnection: Access, Security.

The ATIS document does not specify details of those interfaces.

A.1.4 FP7 OCEAN

The FP7 OCEAN project (http://www.ict-ocean.eu/) is studying among others CDN interconnection.

A.2 Functionality of the CDN interconnection

TS 102 990 [1] specifies use cases and requirements for CDN interconnection. This clause highlights some of the requirements to start architecture discussions.

A.2.1 Content distribution, upstream or downstream initiated

Mechanism(s) will be needed to control the movement content from the Upstream CDN to the Downstream CDN. Content movement can be initiated by either party:

- Upstream CDN initiated: e.g. triggered by the Content Service Provider, the Upstream CDN may pre-provision (specific delivery servers of) the Downstream CDN with specific content.
- Downstream CDN initiated: e.g. triggered by the first Consumer request for a specific content or by a cache miss, the Downstream CDN pulls the retrieves the content from (a specified origin server of) the Upstream CDN.

Preferably, the mechanisms to do this are identical or very similar. Figure A.2.1.1 is an example of an inter-CDN content distribution flow. The dotted lines in the figure are CDN-internal flows, which are outside the scope of the present document. Most likely, a mechanism should be added for confirming that the distribution of content has taken place successfully.

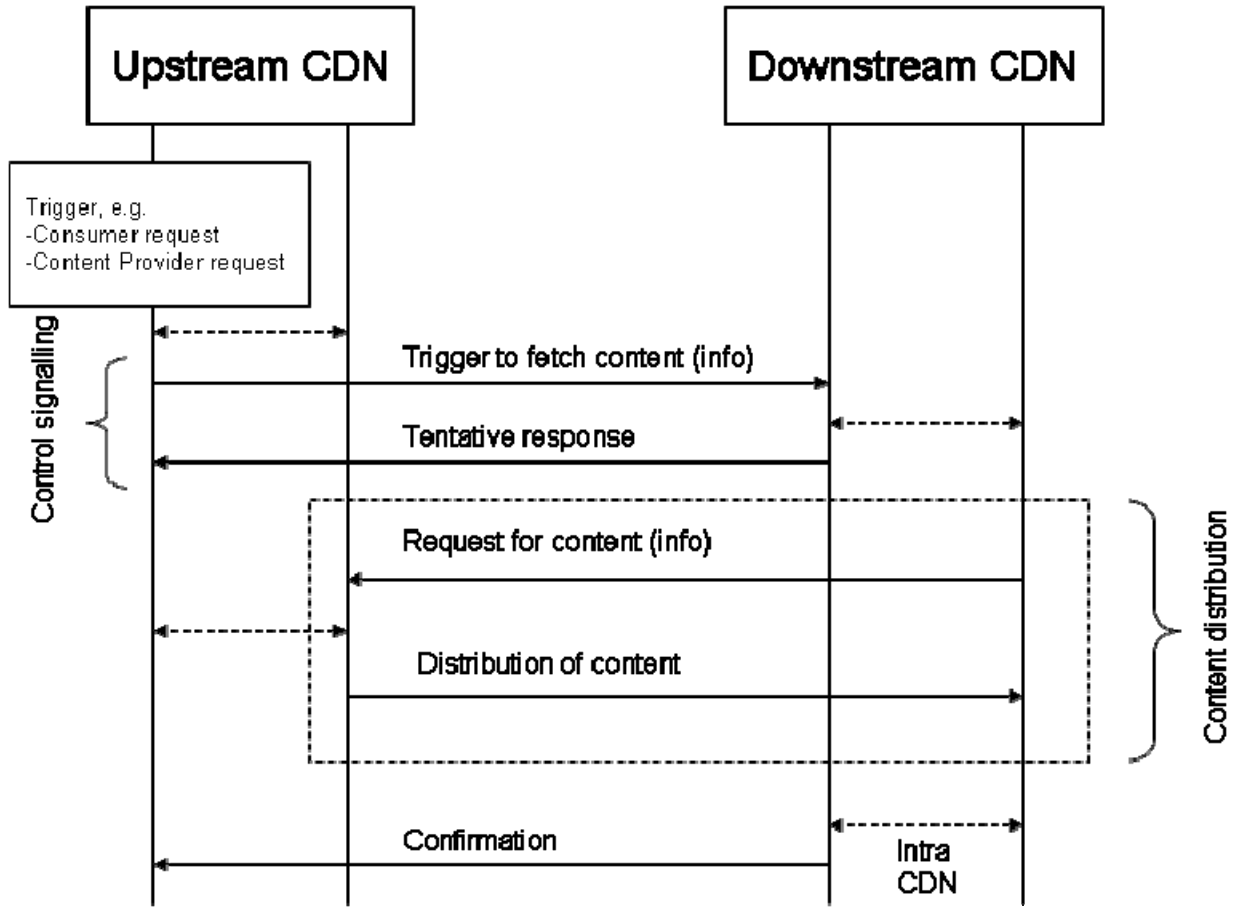

Figure A.2.1.1: Example of an inter-CDN content distribution flow

A.2.2 File-based and stream-based content

The CDN interconnection should support both file-based (e.g. content-on-demand) and stream-based (e.g. broadcast) content. This will lead into differences in content distribution and reporting.

- File-based content has a specified end:
  - So the successful distribution of the full content can be signalled.
  - Reporting can be done on a per-file basis.
- Stream-based content does not have a pre-determined end:
  - So only the successful set-up of publication points for stream relay can be signalled.
  - Reporting may be different.

Adaptive streaming is a special case, as the content is file based, but the content does not need to have a pre-determined end, as the manifest file may be updated on a regular basis. It was also recognized by IETF WG CDN-I that adaptive streaming needs special attention.

Multicast in the Downstream CDN may be another special case.

A.2.3 Request routing, per request or not

Both IETF and ATIS present a request-routing interface to ensure that the Downstream CDN can handle specific content delivery requests. The interface could be used on a per-user request basis, but that would be potentially inefficient and delay prone. It could also be used to exchange information for a group of requests, which is equivalent to a capability exchange. Effectively, this could be some sort of resource reservation by the Upstream CDN on (servers of) the Downstream CDN. A point of discussion is the level of detail that the interface can or should provide. Can the Upstream CDN for example reserve a specified amount of storage and streaming capacity in a specific Downstream CDN server?

A.2.4 Reporting or logging

Clause 6.8 of TS 102 990 [1] contains several requirements on pushing and pulling content status reports and transaction reports, and monitoring of on-going content delivery. Two approaches are possible:

- Logging: specify only an interface and/or messages to exchange unprocessed log files.
- Reporting: also specify (extensible) datamodel(s) for the summarized status and transaction reports.

The former approach is the simplest and most detailed, whereas the latter approach is more rigorous and has a better separation of concerns. Especially in case of adaptive streaming, the former approach can lead to excessively sized log files.

A.2.5 Security mechanisms

TS 102 990 [1] requires security mechanism for among others:

- Verification of the integrity of the content (unchanged, unmutilated)
- Authentication of the content source
- Authorization of content requests by Consumers
  - In particular, an "anti-deep-linking" mechanism is required
- DRM-related requirements

A.2.6 Content adaptation

Content adaptation has been discussed in detail by IETF, resulting in a decision that IETF CDN-I rel.1 will not support content adaptation in any form. As TS 102 990 [1] has specified requirements on content adaptations, mechanisms are needed to support it.

Annex B (informative): Datamodel analysis

B.1 Metadata structure

B.1.1 General

Metadata represents any data that is related to a content and used by the CDNs in order to distribute it properly and effectively. This clause lists all the data that can be distributed using Metadata Exchange and groups it into several categories.

B.1.2 CDN Blacklists/Whitelists

This group of metadata includes the information about which CDNs can the content be distributed to. It indicates weather the content can be distributed to any CDN, any CDN but those listed in a blacklist or only those CDNs listed in a whitelist.

B.1.3 Capabilities required for content delivery

This group of metadata lists the capabilities that a CDN is required to have in order to be allowed to deliver the content. Such capabilities can be:

- Delivery protocol support (HTTP, RTSP, RTMP, MMS/RTSP and others).
- Delivery format support (HTML, WMV, H.264, AAC, FLV, MP3 and others).
- Data protection support (HTTPS, Tokens, content access lists and others).
- Delivery method support (multicast, unicast and others).

B.1.4 Content Access Lists

Content access lists are metadata that define which clients can the content be delivered to. It is a list structure very similar to standard network access lists starting with several rows of allow or deny statements and ending with a deny all or allow all statement. These statements have to be evaluated by the delivery node in order to decide whether a client request is to be honoured or not. There are multiple methods by which clients can be identified within the access lists. Some of them can be IP prefixes (whole networks or hosts), country identifies (matched to IPs using geo location), cookies, https credentials and others.

B.1.5 Content Manipulation Policy

This clause of metadata includes information about how can the content be manipulated by the CDN before it is delivered. Some of these parameters can be:

- Ability to use specified delivery protocol.
- Ability to use specified delivery format.
- Codecs that can be used to reformat the content.
- Codec parameters to be honoured:
  - Resolutions allowed.

  - Bandwidths allowed.
  - Key Frame frequencies allowed.
  - Others.
- Ability to use specific delivery method.
- Ability to re-encrypt content to different encryption.

B.1.6 Multi-Segment related metadata

This clause of metadata includes information related to content that is a segment of a larger group. This information can include:

- Unique content group identifier.
- A link to a manifest related in which the segment is linked.
- Identifier of previous segment.
- Actual segment number.
- Identifier of next segment.
- Total segment count.

B.1.7 Security and DRM related metadata

This clause contains security related data. Some of this data can include:

- Certificates proving content originality.
- Certificates and keys related to encryption functionality.
- Certificates used for https authorization.
- Checksums used to verify data integrity.
- Certification authority certificates.

B.1.8 Reporting related metadata

This clause lists information about which logs and other information should be gathered when delivering the content. This metadata can indicate:

- The need to format reports in a defined way.
- The need to include specific information in the reports.
- The need to send (or upload) reports directly to specified destinations.
- The need to transfer reports in a specific way.
- The need to encrypt reports using a specific method/key.

Annex C (informative): Scenarios for using CDN-I procedures

C.1 General

Clause 6 of the present document specifies a set of independent procedures for CDN interconnection. This annex provides some examples how procedures can be used in combination to achieve different types of CDN interconnection. Clause C.2 describes the basic life-cycle of an interconnection between two CDNs. Clauses C.3, C.4 and C.5 describe multiple alternatives representing actual content delivery. They differ by the level of control and reporting that the CDNs are willing to exchange.

C.2 Basic CDN Interconnection life-cycle

In this scenario an interconnection is established between a uCDN and a dCDN. This interconnection is then used to deliver dCDN footprint and capability information to the uCDN and to bootstrap the other required interfaces. These are then used to distribute content to the dCDN. From that point the client requests can be redirected to the dCDN. As the dCDN completes client requests it gathers logs and reports describing the delivery process of specific pieces of content. It uses the request and content control interface to deliver this information back to the uCDN. In the end the uCDN initiates the termination of the CDN-I interconnection. The dCDN completes all outstanding client requests, delivers all related reporting information and terminates the interconnection.

Figure C.2.1 describes this scenario through procedures. It lists the procedures in chronologic order and assigns them to the CDN that initiated their execution.

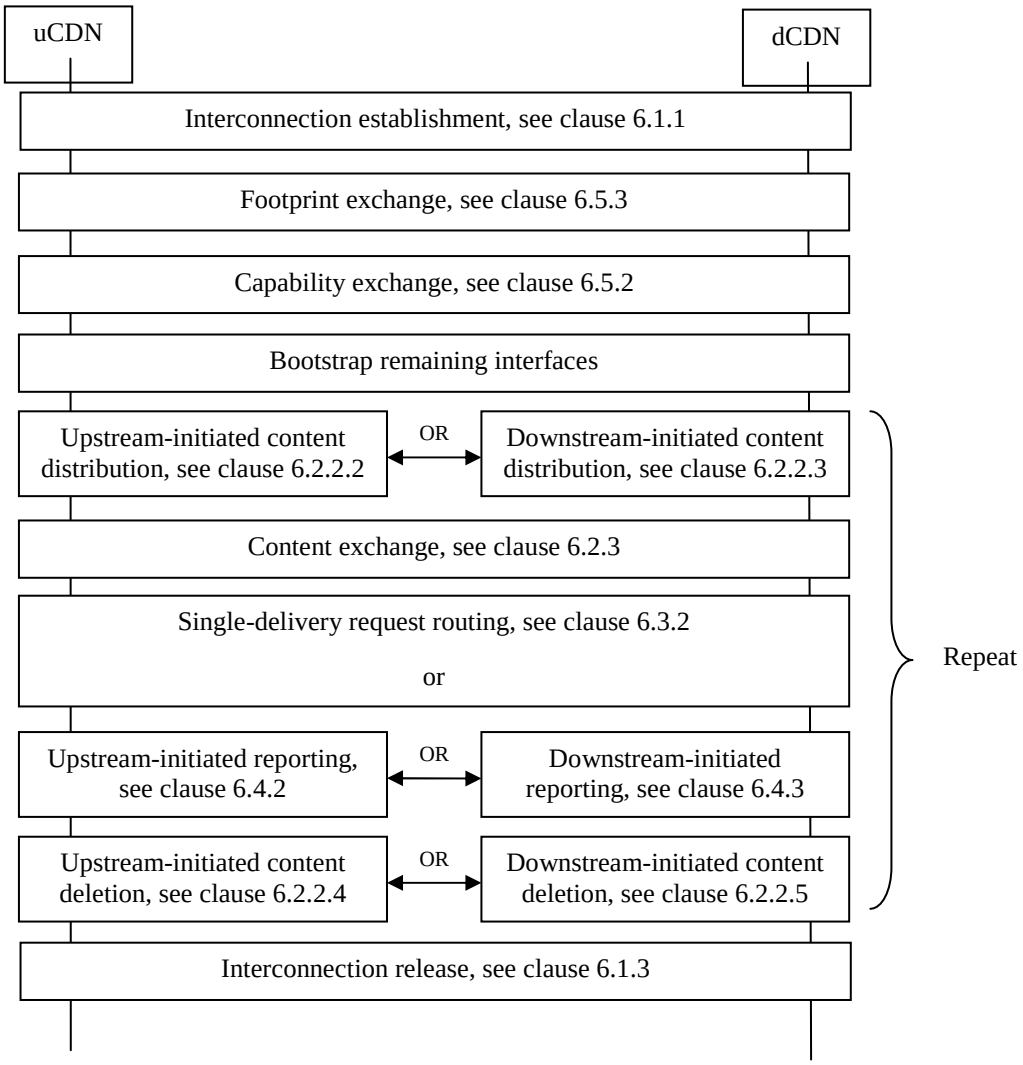

Figure C.2.1: Scenario basic CDN interconnection life-cycle

C.3 Premium delivery of content

Figure C.3.1 shows a procedure scenario aimed at premium delivery of content. The uCDN (on behalf of the Content Provider) is in full control of each individual delivery. The uCDN uses the upstream-initiated content distribution procedure (clause 6.2.2.2) to distribute content to the dCDN, and it waits until the dCDN confirms that it has fully received the content item. From that point, for each user request, the uCDN checks with the dCDN whether it can deliver the identified content item at the agreed quality, and it subsequently directs the user request to dCDN. The uCDN also requests the dCDN for reporting on the delivery of the content item to that user.

This scenario is appropriate when the Content Provider requires for control of the premium content delivery.

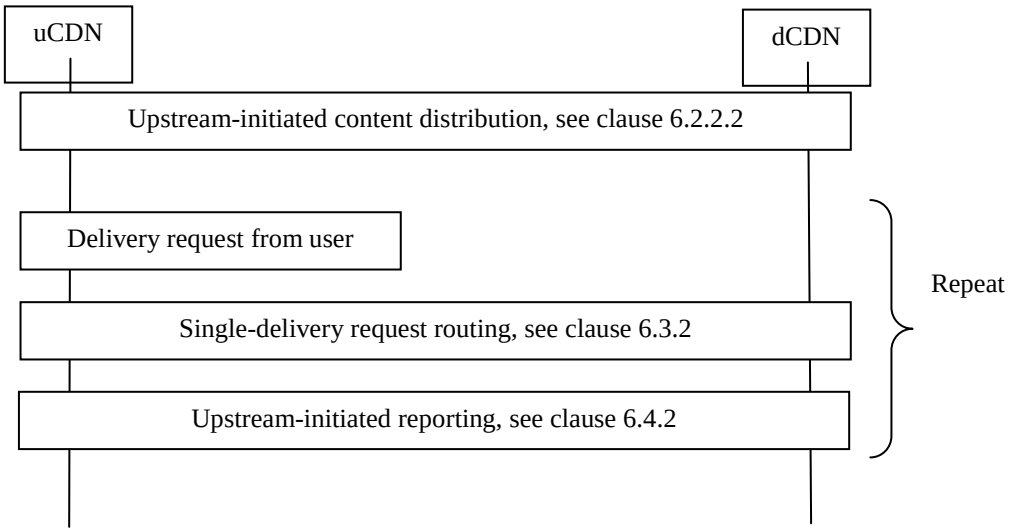

Figure C.3.1: Scenario for premium delivery of content.

C.4 Managed delivery of content

Figure C.4.1 shows a procedure scenario aimed at managed delivery of content. The uCDN (on behalf of the Content Provider) uses the CDN-I interfaces to manage the dCDN selection, but it relies on the dCDN for the quality of experience for actual delivery. On a regular basis, the uCDN receives footprint information from the dCDN. The uCDN directs delivery requests from a user to a dCDN that is selected on basis of footprint information (and possibly also other information, like capabilities). Upon a cache miss, the dCDN performs a downstream-initiated content distribution to get the to-be-delivered content in real time. The dCDN decides what and when to report to the uCDN.

This scenario is practical when the uCDN and the dCDN have a high-speed link to quickly handle cache misses.

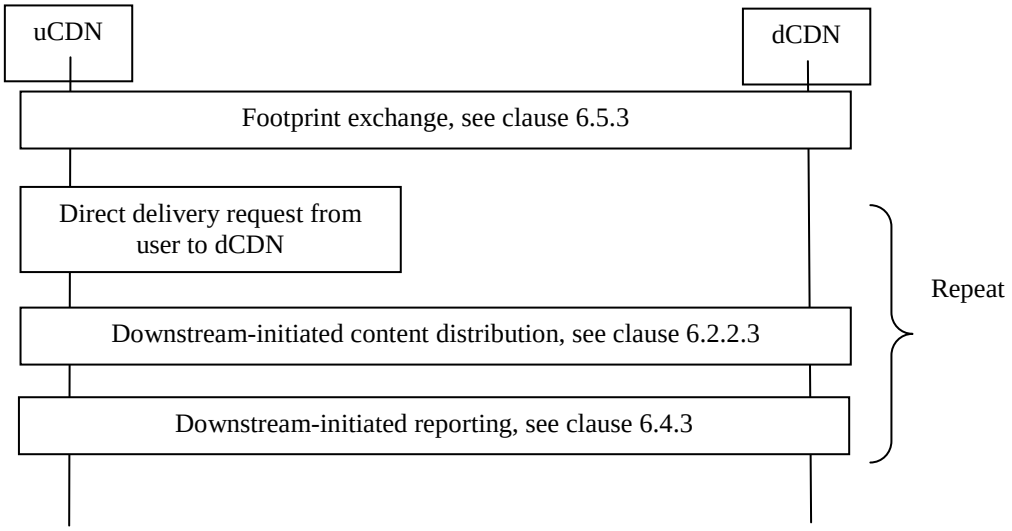

Figure C.4.1: Scenario for managed delivery of content

C.5 Best-effort delivery of content

Figure C.5.1 shows a procedure scenario aimed at best-effort delivery of content. In this scenario, the uCDN just directs delivery requests from its users to the dCDN without any previous checks or content distribution. Upon a cache miss, the dCDN performs a downstream-initiated content distribution to get the to-be-delivered content in real time.

This scenario is practical when the uCDN wishes to fully rely on the dCDN for delivery.

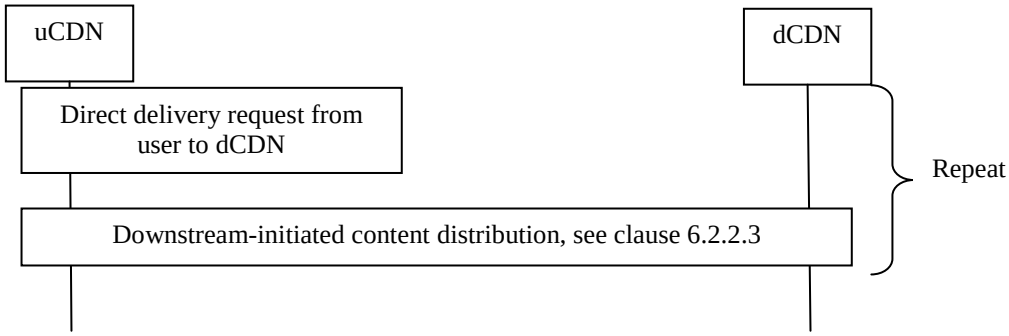

Figure C.5.1: Scenario for best-effort delivery of content

Annex D (informative): Datamodels

This annex presents examples of data models that may be used for CDN interconnection. It represents a set of data that is sufficient to properly keep all the states required for execution of all the basic procedures described in clause 6 of the present document.

D.1 CDN related data models

This category of data represents the information about a specific CDN network from the interconnection perspective.

- **CDN information** contains basic data that is relevant to specific CDN.
- **CDN footprint** defines a single footprint segment that is related to a specific CDN.
- **CDN capabilities** describe the capabilities of a specific CDN within a specific footprint segment.

D.1.1 CDN information data model

Key	Group	Value type	Comment
id	Basic	string	Id of CDN
name	Basic	string	Name of CDN
icfInterface	Basic	url	Address of Interface for Interconnection exchange. (SOAP, REST, etc.)
rcfInterface	Basic	url	Address of Interface for Interconnection exchange. (SOAP, REST, etc.)

D.1.2 CDN footprint data model

Key	Group	Value type	Comment
footprintID	Basic	integer	Footprint identifier
footprintTYPE	Basic	string	Defines the footprint type (ip prefix, geolocation, BGP AS number)
CDN-ID	Basic	integer	CDN identifier
footprintDATA	Basic	string	The variable containing footprint data (as defined n footprintTYPE)

D.1.3 CDN Capabilities data model

Key	Group	Value type	Comment
footprintID	Basic	integer	Footprint identifier
httpFileTransfer	Basic	boolean	Describes the network's capability of delivering files via http
streaming	Basic	boolean	Describes the network's capability of delivering streams
isAuthorization	Basic		
authentication	Basic	string	Coma separated list of mechanisms
dynamicStreaming	Extented	Set of (HLS, HDS, MSS, DASH)	Defines supported dynamic streaming methods

D.2 Content related data models

This clause describes the data models relevant to specific content items and their delivery:

- **Content related metadata data model** includes all the basic information describing the content.
- **Content distribution reporting data model** includes all the data related to the distribution status of a specific content.
- **Content request source data model** includes all the data related to a source of content delivery request.
- **Content delivery reporting data model** includes all the data related to delivery of a specific content.

D.2.1 Content related metadata data model

Key	Group	Value type	Comment
id	Basic	string	Content identification
name	Basic	string	Content name
contentProvider	Basic	string	Content provider identification
description	Basic	string	Content description
contentType	Basic	string	Content MIME type
transferType	Basic	file/stream	Content transfer type
isInfinite	Basic	boolean	Identifies live streams
length	Basic	long	Number of bytes, content contains of
metadataVersion	Basic	timestamp	Time when metadata was updated from content provider
fileVersion	Basic	timestamp	Time when content was updated from content provider
adaptation	Extended	boolean	Defines if content adaptation is allowed
qosDelivery	Extended	structure	Defines the qos parameters for content

D.2.2 Content distribution reporting data model

Key	Group	Value type	Comment
content ID	Basic	string	ID of content
distributionPoint	Basic	string	Name of distribution point
status	Basic	no, download, ready, waiting for undeploy	Content status
hit ratio	Basic	double	Content-specific hit ratio
deploymentTime	Extended	long	Number of seconds transfer has taken

D.2.3 Content request source data model

Key	Group	Value type	Comment
identity	Basic	string	ID of user is optional
isMobile	Basic	boolean	Detect if device is moving (from GPS position, IP change,
ip	Basic	string	Client IPv4 address
ipV6	Basic	string	Client IPv6 address
location	Basic	GPS	Client location
downloaded	Extended	long	Number of bytes transferred towards requester

D.2.4 Content delivery reporting data model

Is list of one or more entities of following class.

Key	Group	Value type	Comment
contentID	Basic	string	ID of content
requestorID	Basic	string	Name of distribution point
startTime	Basic	timestamp	Timestamp of operation start
endTime	Basic	timestamp	Timestamp of operation stop
status	Basic	Progress, success, fail, interrupted, paused	Status of operation at the stop time

D.3 Data model entity relations

This clause visualizes the relations between various data model entities of CDN interconnection.

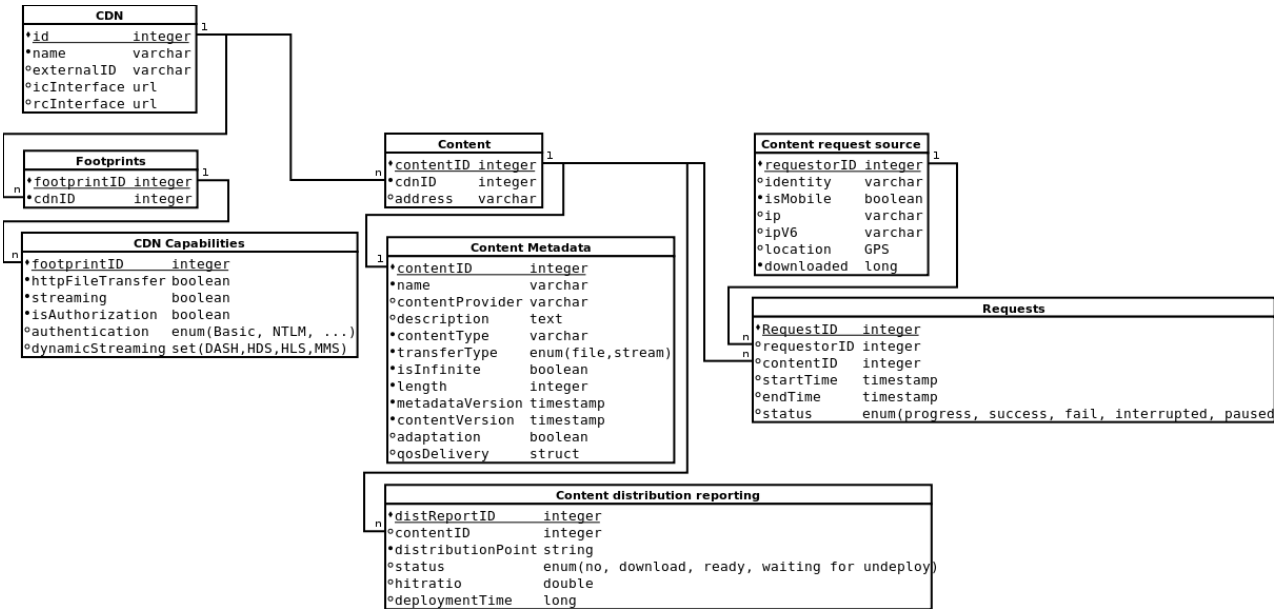

Figure D.3.1: CDN-I data model entity relations

History

Document history		
V1.1.1	April 2013	Publication