
Software Requirements Specification

for

Internetworking of Content Delivery Networks through *Peering*

Version 0.1

Prepared by

Mukaddim Pathan

**Grid Computing and Distributed Systems (GRIDS) Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne, Parkville, VIC 3010, Australia**

Table of Contents

Table of Contents	ii
Change Log	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions and Terminology	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	4
2.3 User Classes and Characteristics	6
2.4 Operating Environment	7
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
3. System Features	8
3.1 Service Registration	8
3.2 Request Initiation to Trigger Peering	9
3.3 Mediator Invokes Creation of Negotiated Relationship	10
3.4 Resource Discovery through PA	11
3.5 Operational Management	12
3.6 Disband or Re-arrangement	14
4. External Interface Requirements	15
4.1 User Interfaces	15
4.2 Communications Interfaces	15

Change Log

Version	Author	Date	Reason For Changes
0.1	Mukaddim Pathan	10.03.08	Initial Draft

1. Introduction

1.1 Purpose

This document describes the requirements specification (SRS) for the software infrastructure (or product) that enables the internetworking of Content Delivery Networks (CDNs) through *peering*, henceforth termed as ‘CDN peering’, and provides an overall description of it. It presents a means for distinct CDNs to coordinate and cooperate with other CDNs, by investigating and developing (a) models for effective internetworking between CDNs through *peering*; (b) protocols for service delivery in a cooperative environment of CDNs; (c) some concrete examples (technology trends) that exhibits the notion of content networking; and (d) policies for autonomic management of service level through resource negotiation in an on-demand basis. Thus, this document provides a basis for evaluating the proposal for internetworking between CDNs. This is the version 0.1 of the software requirements specification.

1.2 Document Conventions and Terminology

When writing this SRS for CDN peering, the following terminologies are used:

- *Web server (WS)*: The container of content comprising of two layers—overlay, which is a collection of Web service host (e.g. Apache, Tomcat), Service Level Agreement (SLA)-allocator, and policy agent, and core, which refers to the underlying hardware infrastructure.
- *Mediator*: A policy-driven entity, authoritative for policy negotiation and management.
- *Service registry (SR)*: Discovers and stores resource and policy information in local domain.
- *Peering Agent (PA)*: A resource discovery module in the peering CDNs environment.
- *Policy repository (PR)*: A storage of Web server, mediator and peering policies.
- P_{WS} : A set of Web server-specific rules for content storage and management.
- P_M : A set of mediator-specific rules for interaction and negotiation.
- $P_{Peering}$: A set of rules for creation and growth of the peering arrangement.

1.3 Intended Audience and Reading Suggestions

This document is written for the researchers, software developers, advanced practitioners, documentation writers, and users involved in CDN domain to initiation an open discussion for exploring development opportunities regarding the internetworking between CDNs. Section 2 discusses the steps that are to be undertaken to “bring-up” or “cease” an internetworking arrangement between CDNs. In the next section, system features with their functional requirements are presented to highlight the major services provided by the intended product. Then the external interface requirements highlighting the logical characteristics of each interface between the software product and the users are discussed. Finally, this specification is concluded with the reference documents on which this document is based on.

1.4 Project Scope

The final product enabling CDN peering assists in coordinated and cooperative content delivery via internetworking among distinct CDNs to allow providers to rapidly “scale-out” to meet both flash crowds and anticipated increases in demand, and remove the need for a single CDN to provision resources. An ad-hoc or planned peering of CDNs requires fundamental research to be undertaken to address the core problems of measuring and disseminating load information, performing request

assignment and redirection, enabling content replication and determining appropriate compensation among participants on a geographically distributed “Internet” scale. In contrast to a single CDN, for which these issues are deeply interrelated and co-dependent, the main thrust for the final product enabling CDN peering is to consider them in a coordinated and cooperative manner among many peered CDNs, whilst satisfying the complex multi-dimensional constraints placed on each individual provider. Each provider must ensure that their individual SLAs are met when serving content for its own customers to end users, while meeting any obligations it has made when participating in a group of many providers.

1.5 References

This document builds on the following references:

- An introduction to CDN technologies [1].
- The research problem for internetworking of CDNs [2]
- The architecture for CDN peering [3].
- The performance models to demonstrate the effects of peering and to predict user perceived performance [4].
- CDN peering models along with the challenges for implementation [5].

2. Overall Description

2.1 Product Perspective

CDN peering allows different CDN providers to share resources in order to provide larger scale and/or reach to each participant than they could not achieve otherwise. It is formed by a set of autonomous CDNs, which cooperate through a mechanism that provides facilities and infrastructure for cooperation in order to virtualize multiple providers. It is expected that an effective peering arrangement between CDNs would require multiple steps to occur.

1. *Initiation*: A CDN’s reach and scale is limited by its ability to handle peak load, cost of equipment, scalable infrastructure, and/or demand for the increased coverage of its infrastructure. Peering allows a particular CDN to achieve larger scale/reach through resource sharing with other CDN(s). It is triggered by an initialization request sent to the mediator under exceptional circumstances, e.g. flash crowds, when the (primary) CDN realizes that it cannot handle a part of the workload on its Web Servers (WSs). The triggering condition must consider the expected and unexpected load increases in the initiating CDN.
2. *Negotiated relationship*: The controlling interest of a CDN to interconnect with other CDNs leads to the creation of a negotiated relationship. In the business domain, this relationship (most likely) would take the form of a legal document which describes the expected level of services from the involving parties such as storage requirements, the required rate of transfer, cost of services; the expected duration of receiving service, penalties for service violations, and other preconditions such as the initiating CDN’s preference to gain resources at a particular region. Negotiated relationships can also be established through means non-technical terms (financial statement) or technical terms (SLAs). Thus, negotiated relationships must specify the interactions among entities including service administration, coordination and disband (or re-arrangement) of internetworked CDNs. In the CDN peering architecture, the mediator instance obtains the resource and access information from the Service Registry (SR), whilst SLAs and other policies from the Policy Repository (PR). The mediator instance on the primary CDN’s behalf generates its service requirements based on the current circumstance and SLA requirements of its customer(s). Therefore, divergent policies are

allowed that specify the information that can be shared during interaction through providing a certain level of visibility to preserve privacy.

3. *Resource discovery*: Once the initiating CDN identifies its roles and activities through the created negotiated relationships for coordination and cooperation between CDNs, the next step is to choose potential CDNs to peer with. The mediator instance passes the service requirements to the local PA to discover external resources from peers. The PA performs the resource discovery process through predicting performance of the peers, working around issues of separate administration and limited information sharing among enlisted CDNs. If there are any preexisting peering arrangements (for a long term scenario) then these will be returned at this point. Otherwise, it carries out short term negotiations with the Peering Agent (PA) identified peering targets.
4. *CDN peering protocols*: The next step is to configure the 'CDN peering' protocols at the conduit of the respective CDNs in order to technically support the terms and policies implicitly specified through the negotiated relationships. This step includes advertising the configurations (topology aspects, geographical proximity, capability, performance, etc.) of enlisted CDNs through inter-PA communication. On establishment of a peering arrangement, these protocols also allow participating CDNs to exchange information regarding the content availability and assists to redirect requests to an optimal peer. Request-redirectation in a peering arrangement depends on the content distribution and request-routing policies (specified in the CDN peering protocols) associated with the content as well as the specific algorithms and methods used for directing these requests.
5. *Operational management*: When the primary CDN acquires sufficient resources from the peers to meet its SLA with the customer, the new peering arrangement becomes operational. Hence, necessary functional policies are deployed and administered in an effective way. Once a peering arrangement is established, all participating parties cooperate in the execution of common goal(s). Peering also enables the CDNs to exchange accounting information to perform billing based on the negotiated relationships. If no CDN is interested in such peering, peering arrangement creation through re-negotiation is resumed by tuning the negotiated relationships with reconsidered service requirements.
6. *Disband or re-arrangement*: An existing peering arrangement may need to either disband or re-arrange itself (within the scope of the negotiated relationships) if any of the following conditions hold: (a) the circumstances under which the arrangement was formed no longer hold; (b) peering is no longer beneficial for the participating CDNs; (c) an existing peering arrangement needs to be expanded further in order to deal with additional load; or (d) participating CDNs are not meeting their agreed upon contributions.

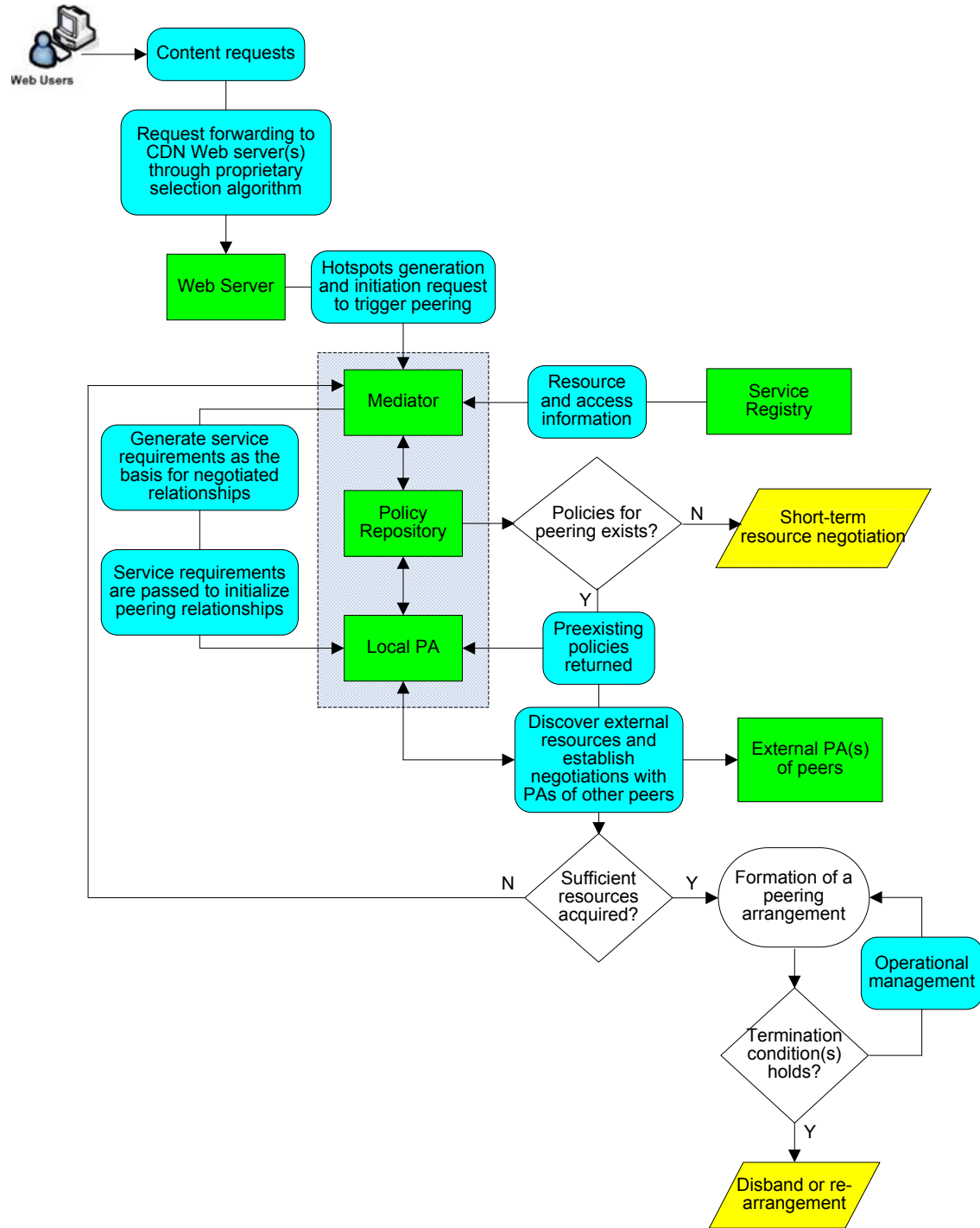


Figure 1: Interaction flows within the abstract view of peering CDNs

Figure 1 presents the interaction flows within the architecture of the CDN peering with an abstraction on its components.

2.2 Product Features

The software infrastructure enabling peering between CDNs can be featured with the following major goals:

- Development and validation of peering and manage the complexity of content delivery across Web servers of multiple CDNs that scale across the globe.
- Decrease cost of Web access, increase QoS through reduced latency, reduce server load, and bandwidth consumption (by a particular CDN server), thus improving the performance of content delivery.
- Assists an existing CDN to alleviate congestions by detecting and handling short-term load spikes (i.e. flash crowds) effectively.

The operations performed by the product components [3] assist to realize the above goals. Component-wise major functions are noted below.

The functions of the *Web server(s)* and its constituents are as follows,

- A Web server replicates content on-demand from the origin server and stores it for future use.
- In the event of Web hotspot, it initiates request to trigger peering.
- The Web services host ensures the delivery of content to end-users based on the negotiated policies with other CDNs.
- The policy agent is responsible (in conjunction with the mediator) for determining which resources can be delegated and under what conditions (policies) delegation is permitted.
- The SLA-allocator performs the provisioning and reservation of Web server's resources (e.g. CPU, bandwidth, storage etc.) to satisfy both local and delegated SLAs, and ensures that the terms of the SLAs are enforced.
- The Web servers' underlying algorithms perform on-demand caching, content selection, and routing between servers.

The *Mediator* performs the following major functions,

- It generates service requirements as the basis for negotiated relationships.
- It passes the service requirements to the PA.
- It works in conjunction with its local PA to discover external resources and to negotiate with other CDNs.
- Once a peering arrangement is established, it controls what portion of the Web traffic (i.e. end-user requests) is redirected to the Web servers of the peering CDNs, which content is replicated there, how the replication decision is taken, and which replication policies are being used.
- It ensures that the participating entities are able to adapt to changing circumstances (agility) and are able to achieve their objectives in a dynamic and uncertain environment (resilience)

The main functions of the *Service Registry* are as follows,

- It encapsulates the resource and service information for each CDN.
- It helps in discovering local resources through enabling the Web servers of CDN providers to register and publish their resource, service and policy details.
- In the face of traffic surges, it supplies any necessary local resource information to the mediator.
- When a new peering arrangement is established, an instance of the service registry is created that encapsulates all local and delegated external CDN resources.

The *Policy Repository* is used to perform the following functions,

- It virtualizes all of the policies within a peering arrangement including P_{WS} , P_M , and $P_{Peering}$ (i.e. Web server-specific policies, mediator policies, peering policies), along with any delegated policies for resources as a result of the peering arrangement.

- It provides a set of rules to the mediator to administer, manage, and control access to the resources in a peering arrangement.
- It returns existing peering policies to the PA during the establishment of long-term peering arrangements.

A *Peering Agent* carries out the following major functions,

- It acts as a policy-driven resource discovery module for establishing negotiations.
- It exchanges policy, resource information, and service requirements with external PAs.
- It is used as a conduit by the mediator to establish negotiations with PAs of other peers and to acquire resources from them.

The operations performed by the components of the CDN peering is driven by semi-autonomous logic that ensures content is served reliably through content replication, request-routing and redirection whilst maintaining constant awareness of the health (e.g. load information) of participants. These major architectural features of the CDN peering are briefly described in the following:

- *Content replication* is performed using a cooperative pull-based approach where participating CDNs assist each others in serving data. Thus, content replication is featured through extending it to participating servers from the peers in a given peering arrangement, subject to the available resources they contribute.
- *Load distribution* is performed by measuring the load information and disseminating it within individual CDNs and amongst other CDNs using a hierarchical approach, where current bandwidth and resource usage of web servers in a CDN is reported to the CDN gateway (i.e. mediator, PA and policy repository as a single conceptual entity) in a periodic or threshold-based manner. The gateways of participating CDNs then communicate aggregated load information describing the load of their constituent servers.
- *Request assignment* and *redirection* is performed at multiple levels – at the DNS, at the gateways to local clusters, and also (redirection) between servers in a cluster. Therefore, end-users can be assigned via DNS (by the peering agents of participating CDNs updating their DNS records regularly) and also via redirection at the CDN gateway when appropriate.

2.3 User Classes and Characteristics

The users of the software infrastructure enabling CDN peering can be differentiated by using their membership and contributions to the system. A given peering arrangement consists of *explicit* and *implicit* members. Explicit members include the *primary* CDN, which is the initiator of a peering relationship, and any *peering* CDNs who cooperate for resource sharing. Implicit members are *content providers* and *end-users*. Implicit members are transparent to a peering arrangement but they share the benefit from it. In addition, users can vary based on the purpose, size, scope and duration of peering. For instance, a short-term peering arrangement is to be automated to react within a tight time frame—as it is unlikely that a human directed negotiation would occur quickly enough to satisfy the evolved niche. On the other hand, establishment of a long-term peering arrangement calls for a human-directed agent to ensure that any resulting decisions comply with participating CDNs' strategic goals. Users can also be classified because of the preferential treatment that they may receive due to the policy that pertains to a particular provider's business logic. Moreover, individual users (or a group of users) can have dynamic QoS requirements depending on the situations that will result in "customized" content delivery. Thus, users (or class of users) can be differentiated based on user-defined QoS specifications while accessing the service.

2.4 Operating Environment

The product (i.e. prototype system) enabling CDN peering is expected to be deployed in a real-world test bed such as PlanetLab for global testing, observations, and for performance evaluation. In this regard, existing Web services technologies will be studied in detail to examine the feasibility of leveraging them. A modular implementation stack could be developed on top of the existing standard application layer (e.g. Apache, Tomcat) and protocols (e.g. HTTP, CDI, HTPC). A modular implementation approach would be useful to perform testing on modules at different stages to ensure correct implementation. It is anticipated that a cryptographically secure auction-based framework will be used to assist content replication among peered CDNs to allow incentives for all participants. Load information could be measured and disseminated within individual CDNs and among other CDNs using distributed load indices such as Distributed Hash Table (DHT) or variations of it. Request assignment and redirection could predominantly rely on DNS-level end-user assignment combined with a rudimentary request assignment policy such as weighted round-robin or least-loaded-first, which updates the DNS records to point to the most appropriate replica server of the peers.

2.5 Design and Implementation Constraints

The challenges in developing the product include virtualization of multiple providers and offloading end-user requests from the primary CDN provider to peers based on cost, performance and load. Due to the proprietary nature of existing CDNs, limited information about response time or service cost is typically available from individual CDNs, and load balancing control is retained by an individual provider within its own Web servers. Therefore, request-redirections must occur over distributed sets of Web servers belonging to multiple CDN providers, without the benefit of the full information available, as in the single provider case. Moreover, an implementation model for the product enabling CDN peering could be based on a complex combination of attributes such as Web server responsiveness or load, expected network delay, or geographic location. Several of these potential attributes vary over time and there is no single repository for listing the value of attributes such as geographic location or expected delay for all Internet-connected systems. It is anticipated that the values used in a CDN peering implementation model are likely to be based on heuristics.

2.6 User Documentation

Along with the software product, a user manual would be written to help people understand the working methodology and usage of the developed prototype system. It would be written for non-technical individuals and the level of content or terminology would differ considerably from, for example, a System Administration Guide, which is more detailed and complex. The user manual would follow common user documentation styles capturing purpose and scope of the product along with key system features and operations; step-by-step instructions for using the system including conventions, messaging structures, quick references, tips for errors and malfunctions; pointers to reference documents; and glossary of terms.

2.7 Assumptions and Dependencies

The product would build on leveraging existing systems. In this regard, necessary inspirations could be obtained by analyzing related systems such as CoDeeN, Coral, Globule, and MotusNet. In particular, the design and implementation approach of MotusNet could be helpful to draw a clear guideline for developing the intended prototype.

3. System Features

The major services and functional requirements for the product can be illustrated by system features. This section is organized by use cases for major system features. In the following, necessary description is provided for each use cases in the system. Each use case description provides information of the associated actors, triggering condition, preconditions, postconditions, response sequences, exceptions and functional requirements (assumptions). Being a major important section of the SRS, this section is expected to go through iterative improvement to make the most logical sense for the intended product.

3.1 Service Registration

This feature is associated with the registration of resource and service information of each CDN to its SR. The use case for this feature is shown in Figure 2.

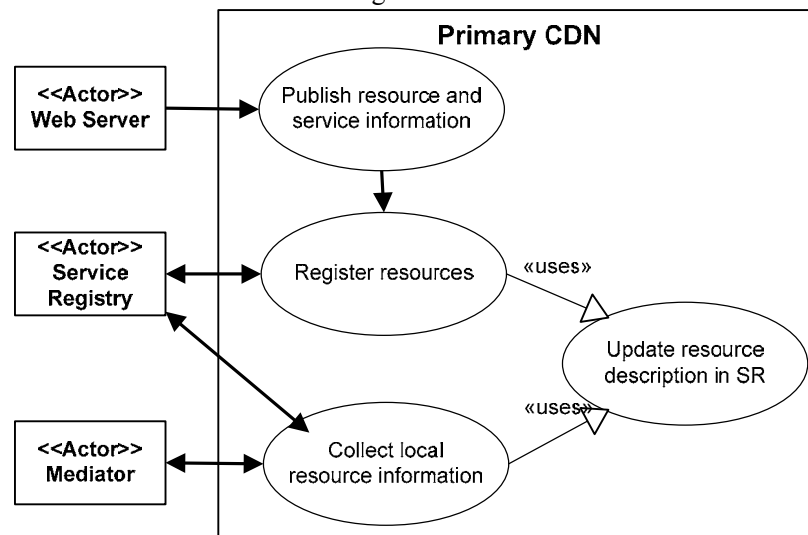


Figure 2: Resource and service information are registered in SR

3.1.1 Actors

WS: Publishes its resource and service information to the SR.

SR: Registers available local resources at the CDN provider and updates them.

Mediator: Collects up-to-date resource information from SR.

3.1.2 Trigger

Service registration is triggered when either one of the following occurs: (a) resources are available as a provider starts operating; (b) previously registered resource information needs to be updated; (c) available local resource information is required in the event of traffic surges; and (d) local and delegated external resource information is to be encapsulated in an SR instance included in an established peering arrangement.

3.1.3 Preconditions

Available local resources of a CDN provider are detected along with their service information such as CPU, storage, upload and download rate, etc. These resources could be provisioned and reserved to satisfy SLAs.

3.1.4 Postconditions

Resources are registered in the service registry and being updated in a regular basis.

3.1.5 Stimulus/Response Sequences

1. Each CDN Web server publishes resource and service information.
2. If it is a new resource, its service information is registered in the SR with a new resource ID. Resource ID counter is incremented.
3. Else, resource information in SR is updated in a regular basis.
4. In the face of traffic surges, information on available local resources along with their IDs is supplied to the Mediator.
5. Local and delegated external resource information is encapsulated in the SR instance in the established peering arrangement.

3.1.6 Exceptions

If a resource fails, its service information is removed from SR and the resource ID counter is decremented.

3.1.7 Functional Requirements

- REQ-1: The format for service information description is defined
- REQ-2: The interaction protocol between Web Server-SR, SR-mediator is defined.
- REQ-3: Resource provisioning, delegation and reservation policies are in place.
- REQ-4: There is an established norm that any resource failure will be reported
- REQ-5: The interaction protocol between Web Server-SR, SR-mediator is defined.

3.2 Request Initiation to Trigger Peering

This feature relates to the initialization of peering between CDNs. The use case for this feature is shown in Figure 3.

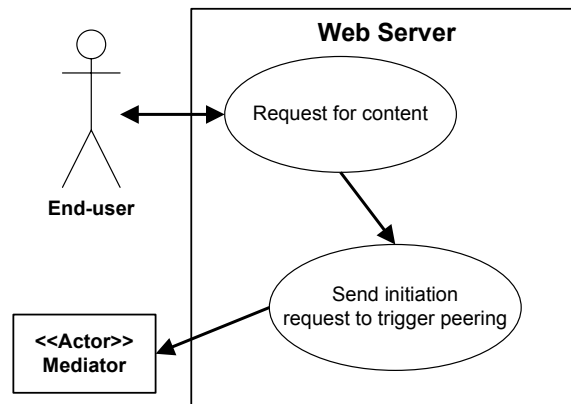


Figure 3: WS sends initialization request to trigger peering

3.2.1 Actors

End-user: Requests for content.

Mediator: Receives initiation request from WS to trigger peering.

3.2.2 Trigger

It is invoked when a (primary) CDN realizes that it cannot handle a part of the workload on its WS(s). The triggering condition considers the expected and unexpected load increases in the initiating (primary) CDN.

3.2.3 Preconditions

Exceptional circumstances such as flash crowds have occurred to place unanticipated load on CDN WSs.

3.2.4 Postconditions

An initialization request is sent to the mediator to trigger peering.

3.2.5 Stimulus/Response Sequences

1. End users request for content from CDN WSs.
2. Flash crowds occurred due to a sudden burst in traffic.
3. Hotspot is generated and provider is unable to handle excess load on its WSs.
4. WS sends initialization request to the mediator to trigger peering.

3.2.6 Exceptions

If the user requests experience service timeout (threshold), initiation request is cancelled.

3.2.7 Functional Requirements

- REQ-1: Malicious requests are detected and rejected.
 REQ-2: Web servers have already replicated necessary content.
 REQ-3: Both anticipated and unanticipated user requests (traffic) are considered.
 REQ-4: The format of the initiation request is defined.

3.3 Mediator Invokes Creation of Negotiated Relationship

On receipt of the initialization request to trigger peering, the mediator of the primary CDN invokes negotiation. Figure 4 shows use case for this feature.

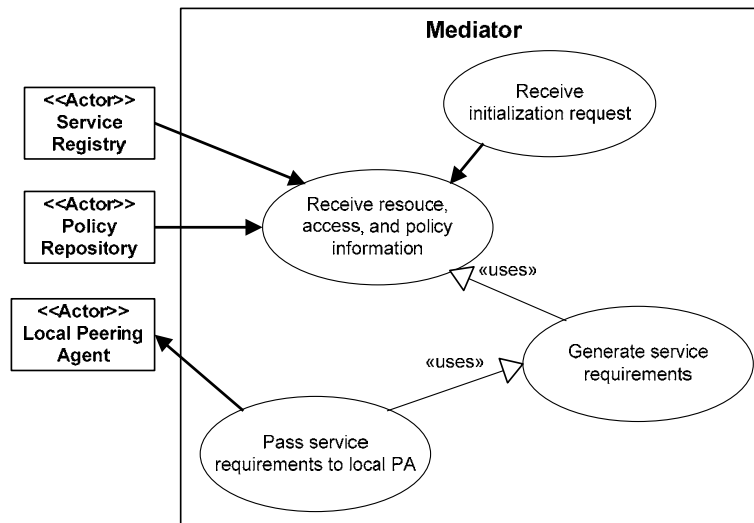


Figure 4: Mediator invokes negotiation through passing service requirements

3.3.1 Actors

SR: Sends resource and access information.

PR: Sends policy information for establishing negotiated relationships.

PA: Receives service requirements from the mediator.

3.3.2 Trigger

Upon receiving the initialization request to trigger peering, the mediator generates service requirements and passes it to local PA.

3.3.3 Preconditions

Initialization request is received.

3.3.4 Postconditions

Service requirements are generated and they are sent to the local PA.

3.3.5 Stimulus/Response Sequences

1. An initialization request to activate peering is received by the mediator.
2. The mediator instance obtains resource and access information from the SR and policy information from PR to establish negotiation.
3. The mediator generates service requirements.
4. If the service requirements are acceptable according to the provider's policy, mediator passes it to the local PA.
5. Else, reject user requests.

3.3.6 Exceptions

If the user requests can not be accepted according the provider policies, service requirements are not passed to the local PA.

3.3.7 Functional Requirements

REQ-1: The format of service requirements is defined.

REQ-2: Mediator-SR, mediator-PR, mediator-PA interaction protocols are defined.

REQ-3: Mediator works in conjunction with the PA to establish negotiation.

3.4 Resource Discovery through PA

The PA of a given primary CDN negotiates with PAs of the peers to perform external resource discovery. Figure 5 shows the use case related to this feature.

3.4.1 Actors

External PA: Negotiates with the PA of the primary CDN.

PR: Receives negotiated policies from the local PA.

3.4.2 Trigger

Local PA communicates with the PAs of peers to discover external resources.

3.4.3 Preconditions

Service requirements are received from the mediator.

3.4.4 Postconditions

Negotiation is performed and a peering arrangement is established through inter-PA communications.

3.4.5 Stimulus/Response Sequences

1. Local PA receives service requirements from the mediator.

2. If any peering policies exist, they are returned to establish long-term peering arrangement.
3. Else, short-term negotiation is performed.
4. Local PA in conjunction with the mediator interacts with external PAs through inter-PA communication protocol.
5. External resources are discovered and negotiation is performed with selected CDN peers.
6. If sufficient resources are acquired, a peering arrangement is established
7. Else, mediator re-evaluates service requirements and sends to the local PA to perform re-negotiation.

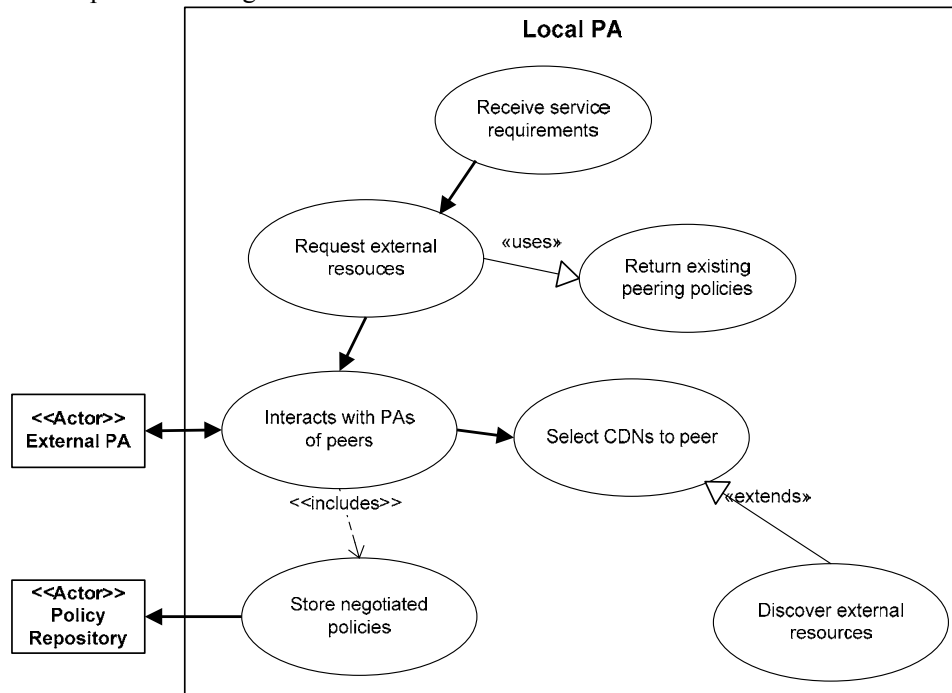


Figure 5: PAs interact to discover resources of peers

3.4.6 Exceptions

If the user requests can not be accepted according the provider policies, service requirements are not passed to the local PA.

3.4.7 Functional Requirements

- REQ-1: Interaction protocols between PAs are identified.
- REQ-2: Malicious requests are identified and acted upon.
- REQ-3: There are existing policies for long-term peering arrangement.
- REQ-4: There is procedure to perform short-term negotiation.

3.5 Operational Management

This feature is responsible for the inter-CDN protocol configuration, resource initialization and for ensuring effective operations of the established peering arrangement. The use case for this feature is shown in Figure 6.

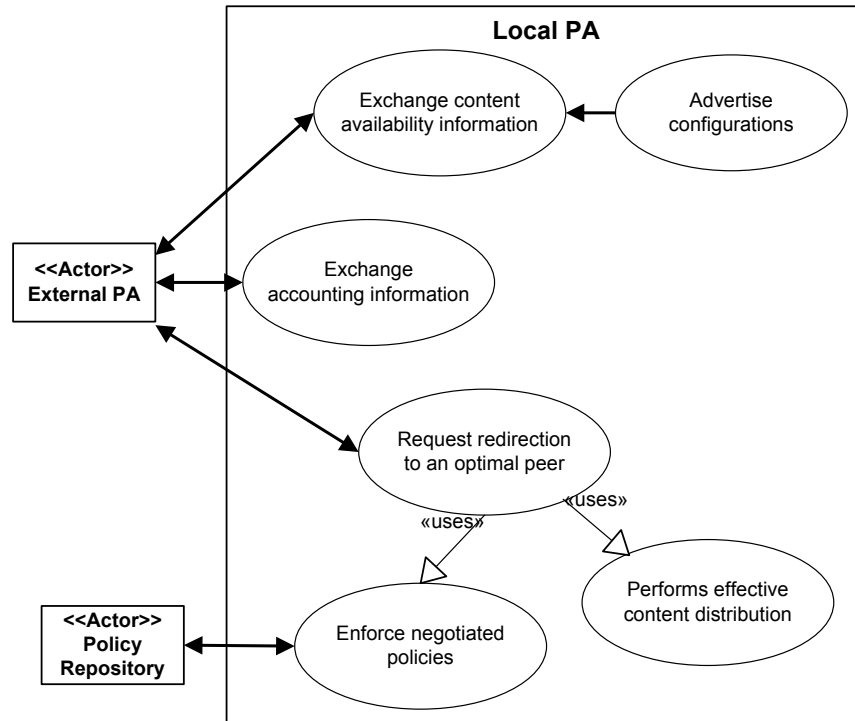


Figure 6: Peers cooperate to ensure effective content delivery

3.5.1 Actors

External PA: Exchanges configurations and content availability information, and accepts requests to an optimal peer.

PR: Assists in enforcing negotiated policies.

3.5.2 Trigger

Once policies are negotiated and a peering arrangement is established, PAs interact each other in the execution of common goal(s).

3.5.3 Preconditions

A peering arrangement is established.

3.5.4 Postconditions

Necessary functions policies are deployed and administration for effective operations.

3.5.5 Stimulus/Response Sequences

1. Once a peering arrangement is established, the initiating primary CDN through its local PA advertise configuration information to technically support the negotiated relationships between enlisted CDNs.
2. PAs exchange content availability and load information to identify an optimal peer to handle user requests.
3. Request is redirected to an optimal peer's Web server.
4. PAs exchange accounting information to perform billing based on negotiated relationships.
5. The policy repository instance in the established peering arrangement assists in the deployment, administration and enforcement of the functional policies.

3.5.6 Exceptions

If a peered CDN provider refuses to accept user requests, a given peering arrangement ceases.

3.5.7 Functional Requirements

REQ-1: Negotiation is established between selected CDN peers.

REQ-2: Primary CDN has already acquired sufficient external resources.

REQ-3: Malicious requests are identified and acted upon.

REQ-3: Functional policies are identified and deployed.

REQ-4: Effective content delivery is ensured through SLA satisfaction.

3.6 Disband or Re-arrangement

There could be circumstances under which a given peering arrangement may need to disband or re-arranges itself. This feature is associated with the use case shown in Figure 7.

3.6.1 Actors

External PA: Interactions with local PA of the primary CDN.

3.6.2 Trigger

Any of the termination condition holds.

3.6.3 Preconditions

There is an established peering arrangement.

3.6.4 Postconditions

A given peering arrangement is disband or it is re-arranged.

3.6.5 Stimulus/Response Sequences

1. A given peering arrangement is currently in operation and peers are cooperating for effective content delivery.
2. If any of the termination condition holds, the local PA disbands or re-arranges the peering arrangement.

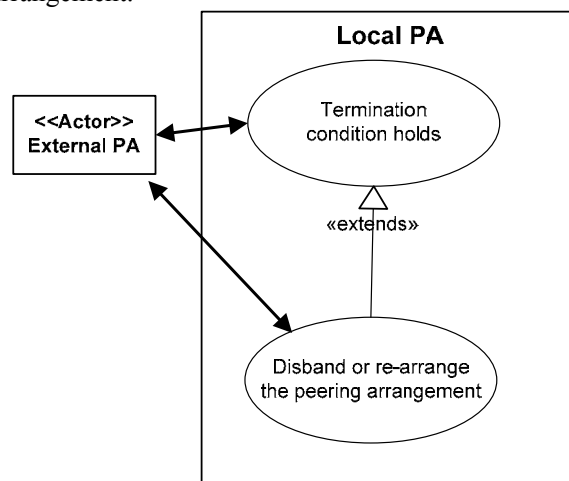


Figure 7: A peering arrangement is disband or re-arranged

3.6.6 Exceptions

If there are exceptional circumstances such as natural disaster, theft, etc. for which any peer is unable to honor the negotiated relationships, a given peering arrangement is not disband or re-arranged and SLA conditions are bypassed.

3.6.7 Functional Requirements

REQ-1: Policies identifying the consequences of SLA violation are defined.

REQ-2: Policies are in place to perform renegotiation for problem resolution.

REQ-3: Administrative policies are deployed for operations in a peering arrangement.

4. External Interface Requirements

4.1 User Interfaces

This section describes the logical characteristics of each interface between the intended software product and the users. For user interface design, common GUI standards will be followed along with the presence of keyboard shortcuts, error message display standards etc., and standard buttons and functions (i.e. help) will appear on every screen. Details of the user interface design are intended to be documented in a separate user interface specification.

4.2 Communications Interfaces

As mentioned earlier, the intended product to exploit existing Web service technologies to leverage existing infrastructures through building an overlay. The communication among software component would be performed through message passing over the IP network. From a technical point of view, TCP/IP will be used as the transport protocol, where each CDN server establishes a TCP connection to the network elements using a well-known port number. Messages can then be sent bi-directionally between the server and network elements. All messages consist of a fixed length-header containing the total data length and a request followed by a reply or an acknowledgement. Interaction among surrogates will be performed using HTTP or FTP.

References

- [1] Pathan, M., Buyya, R., Vakali, A. CDNs: State of the Art, Insights, and Imperatives. *Content Delivery Networks*. Buyya, R., Pathan, M., and Vakali, A. (eds.), Springer-Verlag, Germany, Mar. 2008.
- [2] Buyya, R., Pathan, M., Broberg, J., and Tari, Z. A Case for Peering of Content Delivery Networks, *IEEE Distributed Systems Online*, 7(10), USA, Oct. 2006.
- [3] Pathan, M., Bubendorfer, K., Kim, K. H., and An Architecture for Virtual Organization (VO)-Based Effective Peering of Content Delivery Networks, UPGRADE-CN'07, In *Proc. of the 16th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, USA, Jun. 2007.
- [4] Pathan, M., Broberg, J., and Buyya, R. An Approach for QoS-Aware Performance Modeling of Peering Content Delivery Networks. Technical Report, GRIDS-TR-2007-19, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, Oct. 2007.
- [5] Pathan, M., Buyya, R., Broberg, J. Internetworking of CDNs. *Content Delivery Networks*. Buyya, R., Pathan, M., and Vakali, A. (eds.), Springer-Verlag, Germany, Mar. 2008.