Chapter 16
Internetworking of CDNs

Mukaddim Pathan, Rajkumar Buyya, and James Broberg

16.1 Introduction

The current deployment approach of the commercial Content Delivery Network (CDN) providers involves placing their Web server clusters in numerous geographical locations worldwide. However, the requirements for providing high quality service through global coverage might be an obstacle for new CDN providers, as well as affecting the commercial viability of existing ones. It is evident from the major consolidation of the CDN market, down to a handful of key players, which has occurred in recent years. Unfortunately, due to the proprietary nature, existing commercial CDN providers do not cooperate in delivering content to the end users in a scalable manner. In addition, content providers typically subscribe to one CDN provider and thus can not use multiple CDNs at the same time. Such a closed, non-cooperative model results in disparate CDNs. Enabling coordinated and cooperative content delivery via *internetworking* among distinct CDNs could allow providers to rapidly "scale-out" to meet both flash crowds [2] and anticipated increases in demand, and remove the need for a given CDN to provision resources.

CDN services are often priced out of reach for all but large enterprise customers. Further, commercial CDNs make specific commitments with their customers by signing Service Level Agreements (SLAs), which outline specific penalties if they fail to meet those commitments. Hence, if a particular CDN is unable to provide Quality of Service (QoS) to the end user requests, it may result in SLA violation and end up costing the CDN provider. Economies of scale, in terms of cost effectiveness and performance for both providers and end users, could be achieved by

Mukaddim Pathan
GRIDS Lab, Department of CSSE, The University of Melbourne, Australia, e-mail: apathan@csse.unimelb.edu.au

Rajkumar Buyya
GRIDS Lab, Department of CSSE, The University of Melbourne, Australia, e-mail: raj@csse.unimelb.edu.au

James Broberg
GRIDS Lab, Department of CSSE, The University of Melbourne, Australia, e-mail: brobergj@csse.unimelb.edu.au

R. Buyya et al. (eds.), *Content Delivery Networks*,

leveraging existing underutilized infrastructure provided by other CDNs. For the purposes of this chapter, we term the technology for interconnection and interoperation between CDNs as "peering arrangements" of CDNs or simply "CDN peering", which is defined as follows:

Definition of 'peering arrangement' – *A peering arrangement among CDNs is formed by a set of autonomous CDNs* $\{CDN_1, CDN_2, \ldots, CDN_n\}$*, which cooperate through a mechanism* ***M*** *that provides facilities and infrastructure for cooperation between multiple CDNs for sharing resources in order to ensure efficient service delivery. Each* CDN_i *is connected to other peers through a 'conduit'* C_i*, which assists in discovering useful resources that can be harnessed from other CDNs.*

While the peering of CDNs is appealing, the challenges in adopting it include designing a system that virtualizes multiple providers and offloads end user requests from the primary provider to peers based on cost, performance and load. In particular we identify the following key issues:

- *When to peer?* The circumstances under which a peering arrangement should be triggered. The initiating condition must consider expected and unexpected load increases.
- *How to peer?* The strategy taken to form a peering arrangement among multiple CDNs. Such a strategy must specify the interactions among entities and allow for divergent policies among peering CDNs.
- *Who to peer with?* The decision making mechanism used for choosing CDNs to peer with. It includes predicting performance of the peers, working around issues of separate administration and limited information sharing among peering CDNs.
- *How to manage and enforce policies?* How policies are managed according to the negotiated SLAs. It includes deploying necessary policies and administering them in an effective way.

Therefore, 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. Moreover, to ensure sustained resource sharing between CDN providers, peering arrangements must ensure that sufficient incentive exists for all participants [18]. These issues are deeply interrelated and co-dependent for a single CDN. However, they must now be considered 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.

In this chapter, we present an approach for CDN peering, which helps to create "open" CDNs that scale well and can share resources with other CDNs, and thus evolving past the current landscape where non-cooperative CDNs exist. In our architecture, a CDN serves end user requests as long as the load can be handled by itself. If the load exceeds its capacity, the excess end user requests are offloaded to the CDN network of the peers. We also present two new models to support peering of CDNs and identify the challenges associated with realizing these models.

The remainder of the chapter is organized as follows. In Sect. 16.2, we demonstrate the significance and relevance of CDN peering. Next we present the related work highlighting their shortcomings. In Sect. 16.4, we present our approach for CDN peering, followed by the new models to assist CDN peering. Then we discuss the challenges in implementing peering CDNs. In Sect. 16.7, we also identify related core technical issues to be addressed. Finally, we conclude the chapter in Sect. 16.8.

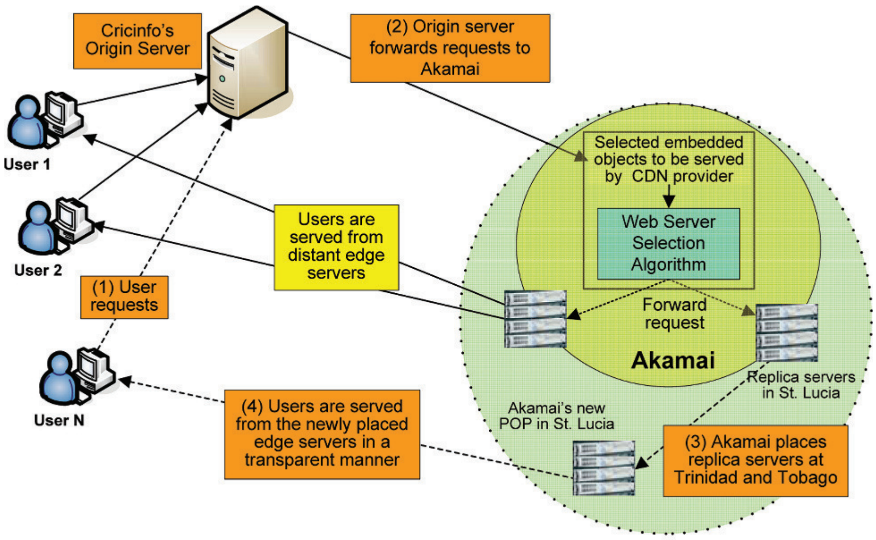
16.2 Significance of CDN Internetworking

As noted in earlier chapters, popular Web sites often suffer congestion, bottlenecks, and even lengthy downtime due to large demands made on the resources of the provider hosting them. As discussed in Chap. 11, this phenomenon can manifest itself as instances of unexpected flash crowds resulting from external events of extreme magnitude and interest or sudden increases in visibility after being linked from popular high traffic Websites like Slashdot[1] or Digg.[2] Increases in demand on Web servers can also be more predictable, such as the staging of a major events like the Olympic Games or the FIFA World Cup. The level of demand generated for many popular Web sites can often be impossible to satisfy using a single Web server, or even a cluster. In 1998, the official Soccer World Cup Website received 1.35 billion requests over 3 months, peaking at 73 million requests per day, and 12 million requests per hour [2]. Similarly high volumes were seen during the 1998 Winter Olympic Games, with the official Website servicing 56.8 million requests on a peak day (and a maximum of 110,414 requests per minute) [13]. During Sept. 11, 2001, server availability approached 0 % for many popular news Websites with pages taking over 45 sec. to load, if at all [15]. Given that end users will wait as little as 10 sec. before aborting their requests, this can lead to further bandwidth and resource wastage [12].

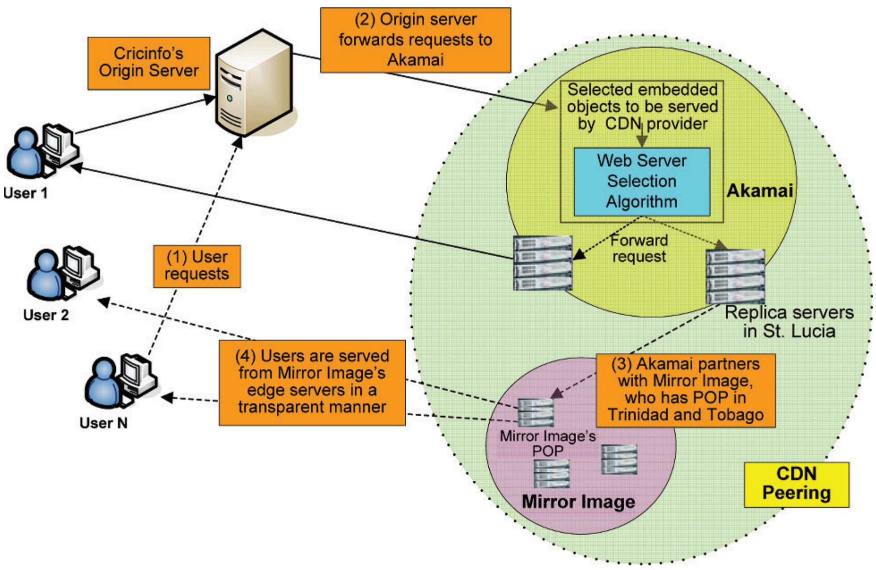
Peering CDNs could be a solution to handle flash crowds, Web resources overprovisioning, and adverse business impact. It is evident that significant gains in cost effectiveness, performance, scalability and coverage could be achieved if a framework existed that enabled peering between CDNs to allow coordinated and cooperative load sharing. To better understand the peering of CDNs, consider the following scenario in Fig. 16.1. Suppose that the ICC Cricket World Cup is being held in the Caribbean, and www.cricinfo.com is supposed to provide live media coverage. As a content provider, www.cricinfo.com has an exclusive SLA with the CDN provider, Akamai [10]. However, Akamai does not have a Point of Presence (POP) in Trinidad and Tobago (a Caribbean island), where most of the cricket matches will be held.

[1] http://www.slashdot.org

[2] http://www.digg.com



(a) Expansion of a CDN provider through placing new POP



(b) Peering between CDN providers

Fig. 16.1 A CDN peering scenario

As being the host of most of the cricket matches, people of this particular part of Caribbean are expected to have enormous interest in the live coverage provided by www.cricinfo.com. Since Akamai is expected to be aware of such event well in advance, its management can take necessary initiatives to deal with the evolving situation. In order to provide better service to the clients, Akamai management might decide to place its surrogates in Trinidad and Tobago, or they might use their other distant edge servers (as shown in Fig. 16.1(a)). Firstly, placing new surrogates just for one particular event would be costly and might not be useful after the event. On the other hand, Akamai risks its reputation if it can not provide agreed QoS for client requests, which could violate the SLA and still cause profit reduction. Hence, the solution for Akamai could involve cooperating with other CDN provider(s) to form a peering arrangement in order to deliver the service that it could not provide otherwise (depicted in Fig. 16.1(b)).

Peering arrangements between CDNs may vary in terms of the purpose, scope, size, and duration. We anticipate that in case of flash crowds, such a peering arrangement should 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. In case of long-duration events (as in Fig. 16.1), we would expect negotiation to include a human-directed agent to ensure that any resulting decisions comply with participating companies' strategic goals.

16.3 Related Work

Internetworking of resource providers is gaining popularity in the research community. An example of such a research initiative is InterGrid [3], which describes the architectures, mechanisms, and policies for internetworking grids so that grids can grow in a similar manner as Internet. Analyses of previous research efforts suggest that there has been only modest progress on the frameworks and policies needed to allow peering between providers. In CDNs context, the reasons for this lack of progress range from technological problems that need solving, to legal and commercial operational issues for the CDNs themselves. For CDNs to peer, they need a common protocol to define the technical details of their interaction as well as the duration and QoS expected during the peering period. Furthermore, there can often be complex legal issues involved (e.g. embargoed or copyrighted content) that could prevent CDNs from arbitrarily cooperating with each other. Finally, there may simply be no compelling commercial reason for a large CDN provider such as Akamai to participate in CDN peering, given the competitive advantage that its network has the most pervasive geographical coverage of any commercial CDN provider.

The internet draft by Internet Engineering Task Force (IETF) proposes a Content Distribution Internetworking (CDI) Model [9], which allows CDNs to have a means of affiliating their delivery and distribution infrastructure with other CDNs who have content to distribute. According to the CDI model, each content network treats neighboring content networks as *black boxes*, which uses commonly defined protocol for content internetworking, while internally uses its proprietary protocol. Thus, the internetworked content networks can hide the details from each other. The CDI Internet draft assume a federation of CDNs but it is not clear how this federation is built and by which relationships it is characterized.

A protocol architecture [21] for CDI attempts to support the interoperation and cooperation between separately administered CDNs. In this architecture, performance data is interchanged between CDNs before forwarding a request by an authoritative CDN (for a particular group), which adds an overhead on the response time perceived by the users. Moreover, being a point-to-point protocol, if one end-point is down the connection remains interrupted until that end-point is restored. Since no evaluation has been provided for performance data interchange, the effectiveness of the protocol is unclear.

CDN brokering [3] allows one CDN to intelligently redirect end users dynamically to other CDNs in that domain. This DNS-based system is called as Intelligent Domain Name Sever (IDNS). The drawback is that, the mechanism for IDNS is proprietary in nature and might not be suitable for a generic CDI architecture. Although it provides benefits of increased CDN capacity, reduced cost, and better fault tolerance, it does not explicitly consider the end user perceived performance to satisfy QoS while serving requests. Moreover, it demonstrates the usefulness of brokering rather than comprehensively evaluating a specific CDN's performance.

Amini et al. [1] present a peering system for content delivery workloads in a federated, multi-provider infrastructure. The core component of the system is a peering algorithm that directs user requests to partner providers to minimize cost and improve performance. However, the peering strategy, resource provisioning, and QoS guarantees between partnering providers are not explored in this work.

From a user-side perspective, Cooperative Networking (CoopNet) [15] provides cooperation of end-hosts to improve network performance perceived by all. This cooperation between users is invoked for the duration of the flash crowd. CoopNet is found to be effective for small Web sites with limited resources. But the main problem of the user-side mechanisms is that they are not transparent to end users, which are likely to restrict their widespread deployment. Hence, it can not be used as a replacement and/or alternative for cooperation among infrastructure-based CDNs.

CoDeeN [16, 23] provides content delivery services, driven entirely by end user demands. However, utilizing its services is not transparent to the end users, as they require them to "opt-in" by setting their browser proxy manually to interact with the CoDeeN network. This user-driven approach means that CoDeeN is essentially an elaborate caching mechanism rather than a true CDN. The authors also noted that the system could be easily abused by bandwidth hogs, password crackers, and licensed content theft, requiring CoDeeN to implement some rudimentary measures such as IP blacklisting and privilege separation for local and external users. Currently, CoDeeN only runs on PlanetLab nodes. Cooperation with external content providers is mentioned by the authors but has yet to be explored.

CoralCDN [11] utilizes a novel Peer-to-Peer (P2P) DNS approach to direct users to replica nodes in the CoralCDN overlay network, reducing the stress on origin servers and improving performance for users. CoralCDN is a cooperative network, but there is no means for nodes (or providers) to participate in peering or internetworking with nodes that are outside of PlanetLab. The nodes that can participate are only offered a coarse level control over their participation (such as allowing individual servers to specify their maximum peak and steady-state bandwidth usage) but there is no fine grained control over exactly what content a node has agreed to serve, nor are there service guarantees. Naturally, given that the service is free and research oriented, content is served on a best effort basis and no compensation is given for participating nodes.

Globule [19, 20] is an open-source collaborative CDN that allows almost any Web-hosting server to participate by installing a customized Globule Apache model, leveraging the ubiquitous nature of Apache as the leading Web server platform. Globule enables server-to-server peering, ad-hoc selection, creation, and destruction of replicas, consistency management and relatively transparent redirection (via HTTP or DNS) of clients to high-performing replicas. Participants in the Globule CDN can act as a *hosting server*, a *hosted server*, or both. This means they can serve content for other users sites as well as their own, in addition to leveraging other participants resources to replicate their own sites. Bandwidth and resource limits can be applied to hosted servers but depend on appropriate facilities being available on the hosting server to enforce this (such as bandwidth limiting Apache modules and "jail" environments to cap resource usage) rather than being handled by Globule itself. A brokerage service is offered where participants can register and access other participants' details in order to initiate negotiations for hosting requests. Such negotiations could include pricing and compensation agreements but this has not been explored deeply in Globule. Security and data integrity aspects (such as dealing with malicious users) are recognized but still remain an open problem for the Globule CDN.

DotSlash [25] is a community driven "mutual" aid service that offers support for small sites that would not have the resources to cope during instances of flash crowds. Provided the site in question has configured itself to access DotSlash, the service automatically intervenes during the flash crowd, allocating and releasing "rescue" servers depending on the load, and is phased out once the flash load passes. A service directory is utilized to allow participants to find each other easily. Participants in DotSlash can only exist in three fixed and mutually exclusive states—*SOS state* where a participant is overloaded and receiving help from other participants, *rescue state* where a participant is aiding another participant in SOS state, and *normal state*. Given the community-driven nature of DotSlash, there is no facility available for internetworked nodes to receive compensation (monetary or resources in-kind) for participating in the peering arrangement.

16.4 Architecture for CDN Internetworking/Peering

Internetworking between different CDNs remains an open problem. The notion of CDN internetworking through a peering mechanism is appealing as a means to address unexpected flash crowds, as well as anticipated short or long term increases in demand, when a single CDN has insufficient resources. They could also allow CDNs (that may not have resources in a particular location) to utilize the resources of other CDNs, by forming a peering arrangement. Thus, peering CDNs can address localized increases in demand for specific content. However, as discussed in Sect. 16.3, many *collaborative* CDNs exist, who function in isolation from each other and commercial CDNs operate with differing policies, methodologies, and QoS expectation. As such, in order for these disparate CDNs to peer, we need to formalize the manner in which they will peer, how they interact, and how QoS levels are set and managed.

In previous work [5, 17], we have presented a policy-driven peering CDNs framework (depicted in Fig. 16.2). The terminologies used to describe the system architecture are listed in Table 16.1. The initiator of a peering negotiation is called a *primary* CDN; while other CDNs who agree to provide their resources are called *peering* CDNs. The endpoint of a peering negotiation between two CDNs is a contract (SLA) that specifies the peer resources (Web servers, bandwidth etc.) that will be allocated to serve content on behalf of the primary CDN. The primary CDN manages the resources it has acquired insofar that it determines what proportion of the Web traffic (i.e. end user requests) is redirected to the Web servers of the peering CDNs.

Figure 16.3 illustrates the typical steps to create a peering arrangement. We summarize these steps in the following:

Step 1. *Creation of a peering arrangement starts when the (primary) CDN provider realizes that it cannot handle a part of the workload on its Web server(s). An initialization request is sent to the mediator.*

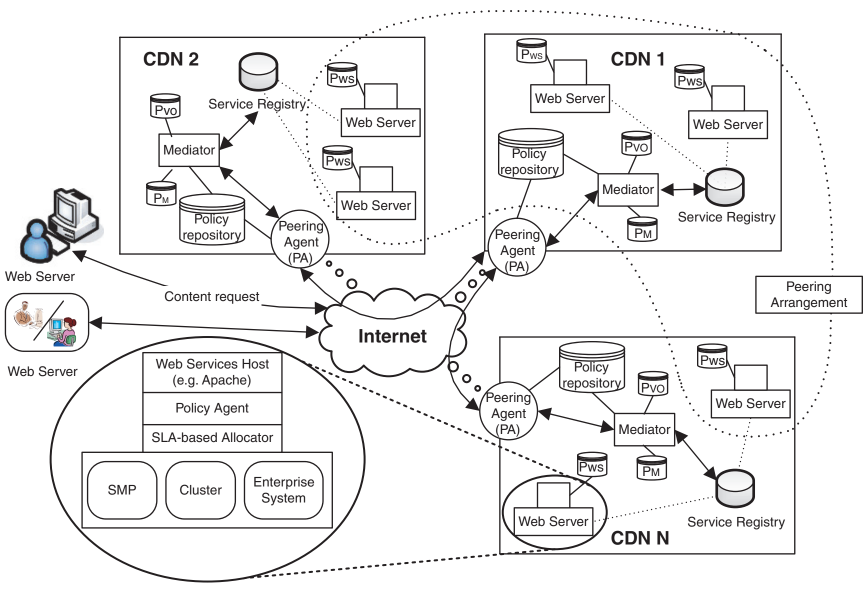


Fig. 16.2 Architecture of a system to assist the creation of peering CDNs

Table 16.1 List of commonly used terms

Terminology	Description
Web server (WS)	A container of content
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

Step 2. *The mediator instance obtains the resource and access information from the SR, whilst SLAs and other policies from the PR.*
Step 3. *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). Hence, it needs to be expanded to include additional resources from other CDNs.*
Step 4. *The mediator instance passes the service requirements to the local Peering Agent (PA). 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 PA identified peering targets.*
Step 5. *When the primary CDN acquires sufficient resources from its peers to meet its SLA with the customer, the new peering arrangement becomes operational. If no CDN is interested in such peering, peering arrangement creation through re-negotiation is resumed from Step 3 with reconsidered service requirements.*

An existing peering arrangement may need to either disband or re-arrange itself if any of the following conditions hold: (a) the circumstances under which the peering was formed no longer hold; (b) peering is no longer beneficial for the participating

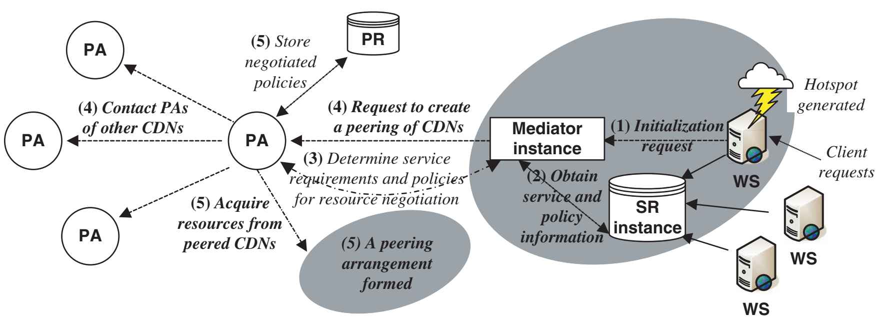


Fig. 16.3 Typical steps for creating a peering arrangement

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.

We have chosen to adapt the IETF policy-based framework to administer, manage, and control access to network resources [24]. Whilst the usage of such a framework has received preliminary investigation for individual CDNs [22], it had not been considered under a framework with multiple peering CDNs. The policy framework consists of four basic elements: *policy management*, *policy repository*, *policy enforcement point (PEP)*, and the *policy decision point (PDP)*.

In the standard IETF policy framework, the admin domain refers to an entity which administers, manages, and controls access resources within the system boundary. An administrator uses the policy management tools to define the policies to be enforced in the system. The PEPs are logical entities within the system boundary, which are responsible for taking action to enforce the defined policies. The policies that the PEPs need to act on are stored in the policy repository. The results of actions performed by the PEPs have direct impact on the system itself. The policy repository stores polices generated by the administrators using the policy management tools. The PDP is responsible for retrieving policies from the policy repository, for interpreting them (based on *policy condition*), and for deciding on which set of policies are to be enforced (i.e. *policy rules*) by the PEPs. Choosing where these logical elements reside in a CDN system will obviously have a significant effect on the utility and performance experienced by participating CDNs and end users, and must be considered carefully and specifically depending on the particular CDN platform that is implementing them.

A policy in the context of peering CDNs would be statements that are agreed upon by the participants within the group of peered CDNs. These statements define what type of contents and services can be moved out to a CDN node, what resources can be shared between the participants, what measures are to be taken to ensure QoS based on negotiated SLAs, and what type of programs/data must be executed at the origin servers.

The proposed model for peering CDNs in Fig. 16.2 has been mapped to the IETF policy framework, as shown in Table 16.2. The policy repository is responsible for storing policies generated by the policy management tool used by the administrator of a particular peering group of CDNs – typically the initiator of the grouping. The policy repository virtualizes the Web server, mediator, and peering policies. These policies are generated by the policy management tool used by the administrator of a particular peering group. The distribution network and the Web server components (i.e. Web Services host, Policy Agent, SLA-based Allocator) are the instances of PEPs, which enforce the peering CDN policies stored in the repository. The peering agent and mediator are instances of the PDPs, which specify the set of policies to be negotiated at the time of collaborating with other CDNs, and pass them to the peering agent at the time of negotiation. The policy management tool is administrator dependent, and will vary depending on the CDN platform. A direct benefit of using such policy-based architecture is to reduce the cost of operating of CDNs by promoting interoperability through a common peering framework, and thus allowing CDNs to meet end user QoS requirements under conditions of heavy load.

Table 16.2 Policy mapping

Policy Framework Component	Peering CDNs Component	Specified Policies	Description
System	*Peering CDNs*	All policies in the system	The distributed computing and network infrastructure for peering CDNs
Admin domain	*Peering arrangement*	Negotiated peering policies	An administrative entity for resource management and access control
Policy management tool	*Administrator dependent*	–	An administrator dependent tool to generate policies
Policy repository	*Policy repository*	Web server, peering and mediator policies	Storage of policies in the system
Policy Enforcement Points (PEPs)	*Web Services host, Policy Agent, SLA-based allocator*	Web server policies	A logical entity which ensures proper enforcement of policies
PDPs	*Mediator*	Mediator policies, peering policies	An authoritative entity for retrieving policies from the repository

16.4.1 Performance Gain Through Peering

We develop the performance models based on the fundamentals of queuing theory to demonstrate the effects of peering between CDNs and to characterize the QoS performance of a CDN.

It is abstracted that N independent streams of end user requests arrive at a conceptual entity, called *dispatcher*, following a Poisson process with the mean arrival rate $\lambda_i, i \in \{1,2,\ldots,N\}$. The dispatcher acts as a centralized scheduler in a particular peering relationship with independent mechanism to distribute content requests among partnering CDNs in a user transparent manner. If, on arrival, a user request can not be serviced by CDN i, it may redirect excess requests to the peers. Since this dispatching acts on individual requests of Web content, it endeavors to achieve a fine grain control level. The dispatcher follows a certain policy that assists to assign a fraction of requests of CDN i to CDN j.

For our experiments, we consider an established peering arrangement consisting of three CDNs. It is assumed that the total processing of the Web servers of a CDN is accumulated and each peer contains same replicated content. The service time of each CDN's processing capability follows a general distribution. The term 'task' is used as a generalization of a request arrival for service. We denote the processing requirements of an arrival as 'task size'. Each CDN is modeled as an M/G/1 queue with highly variable Hyper-exponential distribution which approximates a heavy-tailed Bounded Pareto service distribution (α, k, p) with variable task sizes. Thus, the workload model incorporates the high variability and self-similar nature of Web access.

In our performance models, participating providers are arranged according to a non-preemptive Head-Of-the-Line (HOL) priority queuing system. It is an M/G/1 queuing system in which we assume that user priority is known upon their arrival to a CDN and therefore they may be ordered in the queue immediately upon entry. Thus, various priority classes receive different grades of service and requests are discriminated on the basis of known priority. In our model, an incoming request (with priority *p*) joins the queue behind all other user requests with priorities less than or equal to *p* and in front of all the user requests with priority greater than *p*. Due to this nature of the peering CDNs model, the effect of peering can be captured irrespective of any particular request-redirection policy.

For our experiments, we consider the expected waiting time as an important parameter to evaluate the performance of a CDN. The expected waiting time corresponds to the time elapsed by a user request before being served by the CDN. In our peering scenario, we also assume an SLA of serving all user requests by the primary CDN in less than 20000 time units.

16.4.1.1 QoS Performance of the Primary CDN

First, we provide the evidence that a peering arrangement between CDNs is able to assist a primary CDN to provide better QoS to its users. The Cumulative Distribution Function (C.D.F) of the waiting time of the primary CDN can be used as the QoS performance metric. In a highly variable system such as peering CDNs, the C.D.F is more significant than average values.

Figure 16.4(a) shows the C.D.F of waiting time of the primary CDN without peering at different loads. From the figure, we see that for a fair load $\rho = 0.6$, there is about *55 %* probability that users will have a waiting time less than the threshold

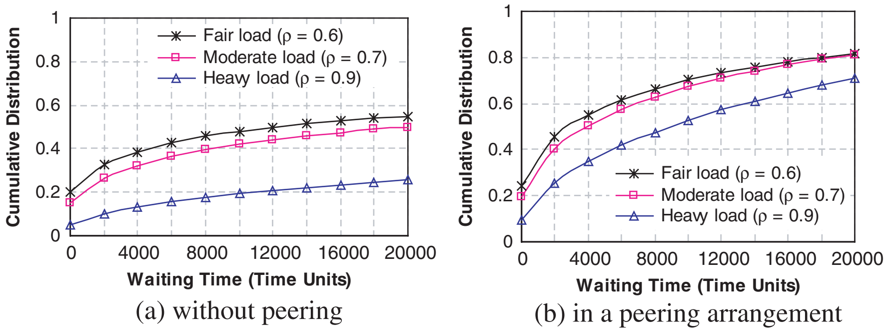


Fig. 16.4 Cumulative distribution of waiting time of the primary CDN

of *20000* time units. For a moderate load $\rho = 0.7$, there is about *50 %* probability to have a waiting time below the threshold, while for a heavy load $\rho = 0.9$, the probability reduces to $> 24\,\%$.

Figure 16.4(b) shows the C.D.F of the primary CDN with peering at different loads. By comparing Fig. 16.3(a) and Fig. 16.3(b), it can be found that for a fair load $\rho = 0.6$, there is about *80 %* probability that users will have a waiting time less than the threshold of *20000* time units. Therefore, in our scenario, peering assists the primary CDN to achieve a QoS performance improvement of about *31 %*. For a moderate load $\rho = 0.7$, there is $> 81\,\%$ probability for users to have waiting time below the threshold, an improvement of about *38 %*. For a heavily loaded primary CDN with $\rho = 0.9$, the probability is about *70 %*, which leads to an improvement of $> 65\,\%$. Moreover, for loads $\rho > 0.9$, still higher improvement can be predicted by the performance models. Based on these observations, it can be stated that peering between CDNs, irrespective of any particular request-redirection policy, achieves substantial QoS performance improvement when comparing to the non-peering case.

16.4.1.2 Impact of Request-Redirection

Now, we study the impact of request-redirection on the expected waiting time of users on the primary CDN. A request-redirection policy determines which requests have to be redirected to the peers. We have evaluated different request-redirection policies within the peering CDNs model. Here, we only demonstrate the performance result using *Uniform Load Balanced (ULB)* request-redirection policy that distributes the redirected content requests uniformly among all the peering CDNs. Our aim is to show that even with a simple request-redirection policy, our performance model exhibits substantial performance improvement on the expected waiting time when compared to the non-peering case.

In our experiments, no redirection is assumed until primary CDN's load reaches a threshold load ($\rho = 0.5$). This load value is also used as the *baseline load* for comparing waiting times at different primary CDN loads. Any load above that will be 'shed' to peers. Each peer is ready to accept only a certain fraction (acceptance threshold) of the redirected requests. Any redirected request to a given peer exceeding this acceptance threshold is simply dropped to maintain the system equilibrium. We consider lightly loaded peers (load of peer 1 and peer 2 are set to $\rho = 0.5$ and $\rho = 0.4$ respectively), while tuning the primary CDN's load ($0.1 \leq \rho \leq 0.9$). It can be noted that a weighted average value of waiting time is presented in order to capture the effect of request-redirection.

From Fig. 16.5, we find that, without request-redirection when the primary CDN's load approaches to *1.0*, the user perceived performance (in terms of waiting time) for service by the primary CDN tends to infinity. On the other hand, with request-redirection the waiting time of the primary CDN decreases as the requests are redirected to the peers. It is observed that for a fair load $\rho = 0.6$, there is about *43 %* reduction in waiting time, while for a moderate load $\rho = 0.7$, it becomes about

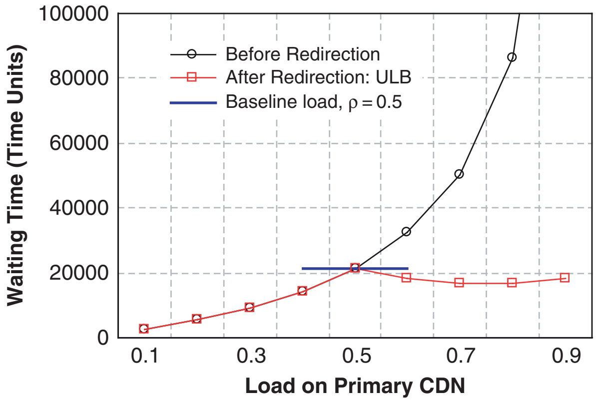


Fig. 16.5 Impact of request-redirection on waiting time of the primary CDN for uniform request-redirection policy

66 %, and for a heavy load $\rho = 0.9$, it reaches to > 90 %. From the results, it is clear that even a naive request-redirection policy like ULB can guarantee that the maximum waiting time is below *20000* time units (as per the SLA). Therefore, better performance results can be anticipated with a scalable and efficient request-redirection policy. Our results also confirms that redirecting only a certain fraction of requests reduces instability and overload in the peering system because the peers are not overwhelmed by bursts of additional requests.

16.5 New Models for CDN Peering

In this section, we propose two new models to assist CDN peering. They are *brokering-based* and *QoS-driven (customized) brokering-based* models. They can be used to complement our peering CDNs model presented in Sect. 16.4. To better understand the uniqueness of these endorsing models and to compare them with existing ones, we first revisit *conventional*, *P2P-based*, and *Internetworked/peered* CDNs. Then we present our newfangled ideas for forming peering CDNs. In Table 16.3, we compare the existing and proposed CDN models and summarize their unique features.

16.5.1 Existing CDN Models

In a *conventional* CDN, end users request content from a particular content provider's Web site. The actual content itself is served by the CDN employed by the content provider from the edge server nearest the end user. There is typically an

Table 16.3 Comparison of CDN models

Features	Typical CDN Models		Advanced Models for CDN Peering		
	Conventional CDNs	P2P-Based CDNs	Peering CDNs	Brokering-Based	QoS-Driven (Customized) Brokering-Based
Nature of Content Delivery	Based on Web server Collaboration	Based on peering and content availability	Based on CDN internet-working/peering	Based on CDN performance	Based on user defined QoS (Customized)
Responsibility for effective content delivery	CDN Provider	Peers/Users	Primary CDN Provider	Content Provider	Content Provider
Entities in agreement	CDN-Content Provider	No real agreement (Self-interested users)	CDN-Content Provider, CDN-CDN	CDN-Content Provider	CDN-Content Provider
Agreement nature	Static	N/A	Short-term or long-term	Policy-based	Dynamic
Scalability	Limited	High	High	High	High
Cooperation with external CDNs	No	No	Yes	Yes	Yes
Cooperation between CDNs	No	No	Yes	No, CDNs work in parallel	No, CDNs work in parallel
Cooperation between users	No	Yes	No	No	No

agreement between the content provider and the CDN provider specifying the level of service that the content provider expects its end users to receive, which may include guaranteed uptime, average delay, and other parameters. Examples of conventional CDNs include Akamai, Limelight Networks, and Mirror Image. They are typically singular entities that do not collaborate with each other to deliver content and meet their service obligations. This approach is most suited to providers that already have pervasive, globally deployed infrastructure and can deploy edge servers close to the majority of their customers, and have enough capacity to deal with peak loads (caused by flash crowds) when their occur. Whilst cooperation between CDNs does not occur, the Web servers of a CDN cooperate among themselves (collaborative content delivery) to ensure content is replicated as needed and all SLAs are met. Responsibility for effective content delivery rests solely on the CDN provider that has agreed to deliver content on behalf of a content provider.

In a *P2P-based* CDN, content providers utilize end users nodes (either fully or as a supplement to a traditional CDN) in order to deliver its content in a timely and efficient manner. Examples of P2P-based CDNs include CoDeeN, Coral, and Globule. The first two are deployed on the volunteer nodes in PlanetLab, while the third runs on end user nodes. CoopNet and DotSlash are other examples where the first allows end users to cooperate during the period of flash crowds to improve user perceived network performance; and the latter is a community-driven "mutual" aid service to alleviate flash crowds. In this type of CDNs, end users can cooperate to improve the performance perceived by all, especially in the same geographical area as many users around the same edge can assist each other in receiving content. This cooperation can be invoked dynamically in the time of need (flash crowds). No real agreement exists that defines a minimal level of participation from contributing end users, making specific QoS targets hard to enforce for content providers. Given that the users themselves are self-interested entities that receive no compensation for participating in such a peering arrangement, they will only perform content delivery when it suits them.

In *Internetworked/peered* CDNs, like the conventional CDNs, a content provider employs a particular CDN provider to serve its content to end users. The chosen CDN could peer with other CDN(s) to assist it to deliver content and meet any SLA it may have established with the content provider. Examples of peering CDNs include IETF CDI model [9], CDN brokering [3], peering of multi-provider content delivery services [1] and our peering CDNs [5, 17]. However, we note that it is ultimately the primary CDN provider's responsibility to ensure that the target QoS level is met. In this case, end users request for content from a particular content provider's Web site. Content can be served by any CDN in the peering relationship. A centralized dispatcher (or an authoritative CDN) within a particular peering relationship, typically run and managed by the initiator of the peering, is responsible for redirecting requests to multiple peers. The agreement between multiple CDNs is separate from that made between a content provider (customer) and the primary CDN. As such, the originating CDN is responsible for the performance of any peering CDN it employs to meet its obligation to the content provider.

16.5.2 Brokering-Based Peering CDNs

Figure 16.6 shows the first of the two models that we propose to assist the creation of peering CDNs. In this case, "cooperative" content delivery is achieved by the content provider, who leverages the services of multiple CDNs to ensure appropriate geographical coverage and performance targets are met. Content provider has the responsibility for efficient content delivery. The interaction flows are: (1) users request content from the content provider by specifying its URL in the Web browser. Client's request is directed to content provider's origin server; (2) the content provider utilizes a brokering system of its own in order to select CDN(s) for delivering content to the end users. A given content provider can select multiple CDNs (based on a CDN's QoS performance, capabilities, current load, and geographical location) for delivering content to its users. The selected CDNs do not need to be aware that they are working in parallel with each other, as the content provider handles the management and separation of responsibilities; (3) a *policy-based* agreement between the content provider and CDN(s) is established; (4) once peering is established, the proprietary algorithm of the selected CDN(s) chooses optimal Web server to deliver desired content to the user.

In order to join in a peering arrangement according to this model, CDN providers can compete each other to provide improved performance. Content provider will keep track of CDNs' performance. Hence, selection of CDN(s) can be based on history information on performance for similar content. It can also give preferential treatment to its users based on certain policy (can be as simple as "receive service according to payment" or any other complex policy).

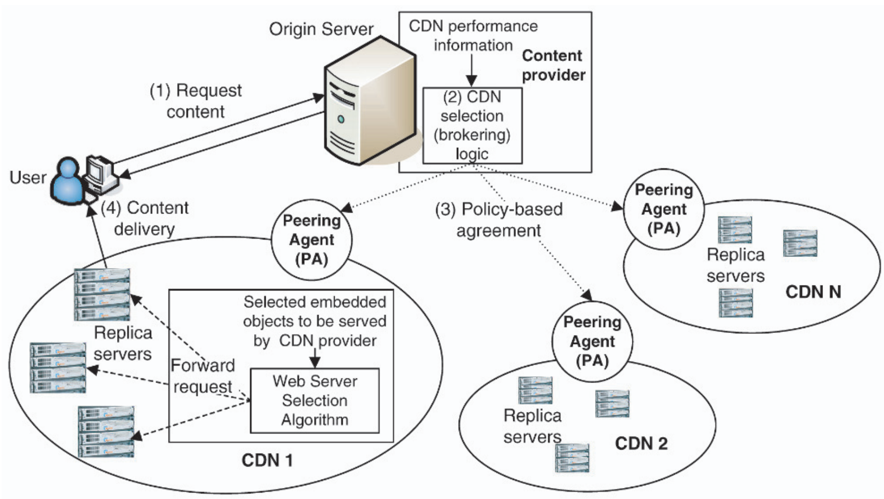


Fig. 16.6 Brokering-based approach to form peering CDNs

16.5.3 QoS-Driven (Customized) Brokering-Based Peering CDNs

While the model in the previous section considers the performance of each potential participant for creating peering CDNs, it does not specifically consider the QoS required by the end users. Users can have dynamic requirements depending on situations (e.g. flash crowds) that will "customize" content delivery. Therefore, sophistication on user-defined QoS is required to be adopted in the model, which may depend on the class of users accessing the service. Hence, in Fig. 16.7 we show an improvement on the previous model to assist peering CDNs formation. In this model, content provider performs the participant selection dynamically based on the individual user (or a group of users) QoS specifications. The interaction flows are: (1) users requests content from the content provider with specific QoS requirements and it reaches the content provider's origin server; (2) content provider uses a dynamic algorithm (based on user-defined QoS) to select CDN(s); (3) content provider establishes *dynamic* agreement with the CDNs it utilizes to ensure user QoS targets are met; (4) once peering is established with the selected CDN(s), desired content is delivered from the optimal Web server of the selected peer(s).

Such peering arrangements are user-specific and they vary in terms of QoS target, scope, size, and capability. It is evident that content provider has the responsibility for effective content delivery through dynamic peering arrangements. Thus, if a particular peering arrangement fails to meet the target QoS to effectively deliver content to the users, content provider re-negotiate with the CDN providers to establish new peering arrangement(s). In Fig. 16.7, we show that in the initial peering arrangement, CDN 1 is responsible for delivering content to the users. As the user QoS requirements change (shown in dotted line), content provider revokes the (customized) CDN selection logic to re-establish a new peering arrangement. In new

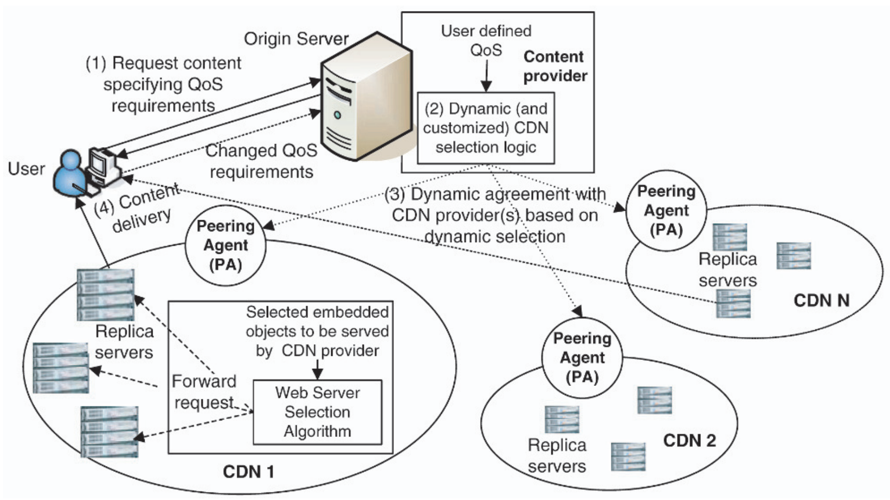


Fig. 16.7 QoS-driven (customized) brokering-based approach to form peering CDNs

peering arrangement, CDN N is the new participant, which delivers content to the end users from its Web server.

16.6 Challenges in Implementing the CDN Peering

There are a number of challenges, both technical and non-technical (i.e. commercial and legal), that have blocked rapid growth of peering between CDNs. They must be overcome to promote peering CDNs. In this section, we outline some of the more common stoppers for uptake of CDN peering.

- *Legal/copyright issues.* There can often be complex legal issues associated with the content to be delivered (e.g. embargoed or copyrighted content) that could prevent CDNs from arbitrarily cooperating with each other. Interactions between peering partners must consider any legal issues associated with the content to be served when delegating it to participating mirror servers from different CDN providers. For instance, if a content provider needs some software or documents that contained logic or information that was embargoed by certain governments (i.e. its access is restricted), all participating CDN providers would have to ensure this was enforced to comply with the appropriate laws. Currently, academic CDNs such as CoDeeN and Coral offer little to no control on the actual content a participating node delivers, and as such participants in these systems could be inadvertently breaking these laws. Content that is copyrighted (e.g. publications, digital media) needs to be carefully managed to ensure that the copyright holder's rights are respected and enforced. The operation (e.g. caching and replication) of some CDNs are user-driven rather than initiated by the content provider, who would prefer to distribute their content on their own terms rather than have it populated in caches worldwide without their consent.
- *Global reach.* As discussed in the previous section, the most common scenario for CDN providers is a centrally managed, globally distributed infrastructure. Companies such as Akamai and Mirror Image have their own far-reaching global networks that cover the vast majority of their customers needs. Indeed, their pervasive coverage is essentially their competitive advantage, and allows them to target the higher end of the customer market for these services. However, few providers can match their global reach, and as such they have little commercial or operational incentive to peer with other smaller providers.
- *Consolidation in CDN market.* Direct peering might be advantageous for small CDN providers, if they wish to compete with larger providers based on coverage and performance. In recent years there has been an enormous consolidation of the CDN marketplace from 20-30 providers down to 5-10 providers of note. It is clear that smaller providers found it difficult to compete on coverage and performance with Akamai and Mirror Image, and subsequently ceased operation or were acquired by the larger providers.
- *Challenges in brokering-based CDN peering.* An approach where a content provider itself manages the selection and contribution of many CDNs to distribute

its content seems appealing, especially, if they have the resources and know-how to manage such an effort. CDN providers could be chosen on their respective merits (e.g. locality, performance, price) and their efforts combined together to provide a good experience for their customers. However, enforcing QoS to ensure a good end user experience (essentially trying to create a robust and predictable overlay network) could be challenging when dealing with multiple providers, especially when they are not actually collaborating, rather simply operating in parallel.

- *Challenges in P2P-based CDN peering.* There has been a growing trend in the last decade toward exploiting user-side bandwidth to cooperatively deliver content in a P2P manner. Whilst initially this started against the wishes of content providers (e.g. Napster, Gnutella), eventually content providers embraced P2P technology, in particular BitTorrent, in order to distribute large volumes of content with scalability and performance that vastly exceeded what was possible with a traditional globally distributed CDN. Content providers have utilized this effectively to distribute digital media (movies, music), operating systems (e.g. Linux) and operating systems patches, games and game patches. With end user bandwidth increases as a result of the proliferation of high-speed broadband, content providers leverage the masses, which upload data segments to peers as they download the file themselves. However, this approach is only effective for popular files, and can lead to poor end user experience for a content that is not being 'seeded' by enough users. As such, it is difficult for content providers to guarantee any particular QoS bounds when the nodes distributing the content are simply end users themselves that may have little motivation to cooperate once they have received their data.
- *Lack of incentives for cooperation.* Further complicating the widespread dependence of this approach is a backlash by Internet Service Providers (ISPs) who are unhappy with the content providers pushing the burden and cost of content delivery onto end users (and subsequently the ISPs themselves). Many ISPs are now actively blocking or throttling BitTorrent and other P2P traffic in response to this trend, to minimize increased utilization and reduction in revenue per user and the resulting cost it places on the ISP in provisioning additional capacity. Many ISPs in more geographically isolated countries (on the so-called 'edges') such as Australia and New Zealand are in particularly unique situations, depending on a small number of expensive data pipes to North America and Europe. As a result, the broadband access offered by ISPs in these regions have fixed data quotas (rather than 'unlimited') that end users are restricted to, in order to ensure they remain profitable. These conditions further discourage widespread adoption and participation by end users in cooperative content delivery.

16.7 Technical Issues for Peering CDNs

Proper deployment of peering CDNs exhibits unique research challenges. In this section, we present some of those unique issues that are to be addressed for peering

CDNs. While there are some solutions existing for related problems in the CDN domain, the notion of internetworking/peering of CDNs poses extra challenges. Therefore, we provide a research pathway by highlighting the key research questions for the realization of peering CDNs.

16.7.1 Load Distribution for Peering CDNs

The load distribution strategy for peering CDNs includes *request assignment* and *redirection*, *load dissemination*, and *content replication*. Coordination among these core issues is another important consideration for successful exploitation of load distribution strategy.

Request redirection and *assignment* to geographically distributed Web servers of peers requires considering end user's location, server loads, and link utilization between the end user and server in addition to task size (i.e. processing requirements of a content request). It should also address the need to handle dynamically changing conditions, such as flash crowds and other unpredictable events. Request assignment and redirection can be performed in a CDN at multiple levels – at the DNS, at the gateways to local clusters and also (redirection) between servers in a cluster [7, 8]. Commercial CDNs 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 [10]. In the peering CDNs, 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 (i.e. mediator, PA and policy repository as a single conceptual entity) when appropriate.

To deal with *Load dissemination issue*, the behavior of traffic can be modeled under expected peak load since in this case the server load is most severely tested. *Load information* can be measured and disseminated within individual CDNs and among other CDNs. A load index can provide a measure of utilization of a single resource on a computer system. Alternatively, it can be a combined measure of multiple resources like CPU load, memory utilization, disk paging, and active processes. Such load information needs to be disseminated among all participating CDNs in a timely and efficient manner to maximize its utility. Such indices will also be crucial to identify situations where forming a peering arrangement is appropriate (e.g. when servers or entire CDNs are overloaded) or when CDNs resources are underutilized and could be offered to other CDN providers. In this context, a hierarchical approach can be anticipated, where current bandwidth and resource usage of web servers in a CDN is reported to the CDN gateway in a periodic or threshold-based manner. The gateways of participating CDNs then communicate aggregated load information describing the load of their constituent servers.

Content replication occurs from origin servers to other servers within a CDN. Existing CDN providers (e.g. Akamai, Mirror Image) use a non-cooperative pull-based approach, where requests are directed (via DNS) to their closest replica server [10].

If the file requested is not held there, the replica server pulls the content from the origin server. Co-operative push-based techniques have been proposed that pushes content onto participating mirror servers using a greedy-global heuristic algorithm [6]. In this approach, requests are directed to the closest mirror server, or if there is no suitable mirror nearby, it is directed to the origin server. In the context of peering CDNs, this replication extends to participating servers from other CDNs in a given peering arrangement, subject to the available resources it contributes to the collaboration.

In summary, the following questions are to be addressed for distributing loads among peering CDNs:

- How to deduce a dynamic request assignment and redirection strategy that calculates ideal parameters for request-routing during runtime?
- How to ensure reduced server load, less bandwidth consumption (by particular CDN server) and improve the performance of content delivery?
- How do participating CDNs cooperate in replicating content in order to provide a satisfactory solution to all parties?
- What measures can be taken to ensure that the cached objects are not out-of-date? How to deal with uncacheable objects?

16.7.2 Coordination of CDNs

Any solution to the above core technical issues of load distribution must be coordinated among all participants in a peering arrangement in order to provide high performance and QoS. A cooperative middleware must be developed to enable the correct execution of solutions developed to address each core issue. Related to this issue, the key question to be addressed is:

- What kind of coordination mechanisms need to be in place which ensure effectiveness, allow scalability and growth of peering CDNs?

16.7.3 Service and Policy Management

Content management in peering CDNs should be highly motivated by the user preferences. Hence, a comprehensive model for managing the distributed content is crucial to avail end user preferences. To address this issue, content can be personalized to meet specific user's (or a group of users) preferences. Like Web personalization [14], user preferences can be automatically learned from content request and usage data by using data mining techniques. Data mining over CDN can exploit significant performance improvement through dealing with proper management of traffic, pricing and accounting/billing in CDNs. In this context, the following questions need to be addressed:

- How to make a value-added service into an infrastructure service that is accessible to the customers?
- What types of SLAs are to be negotiated among the participants? What policies can be generated to support SLA negotiation?
- How can autonomous policy negotiation happen in time to form a time-critical peering arrangement?

16.7.4 Pricing of Content and Services in CDNs

A sustained resource sharing between participants in peering CDNs must ensure sufficient incentives exist for all parties. It requires the deployment of proper pricing, billing, and management systems. The key questions to be addressed in this context are:

- What mechanisms are to be used in this context for value expression (expression of content and service requirements and their valuation), value translation (translating requirements to content and service distribution) and value enforcement (mechanisms to enforce selection and distribution of different contents and services)?
- How do CDN providers achieve maximum profit in a competitive environment, yet maintain the equilibrium of supply and demand?

16.8 Conclusion

Present trends in content networks and content networking capabilities give rise to the interest for interconnecting CDNs. Finding ways for distinct CDNs to coordinate and cooperate with other content networks is necessary for better overall service. In this chapter, we present an approach for internetworking CDNs, which endeavors to balance a CDN's service requirements against the high cost of deploying customer dedicated and therefore over-provisioned resources. In our approach, scalability and resource sharing between CDNs is improved through peering, thus evolving past the current landscape where disparate CDNs exist. In this chapter, we also present two new models to promote CDN peering and identify the associated research challenges. Realizing the concept of CDN peering should be a timely contribution to the ongoing content networking trend.

Acknowledgements Some of the materials presented in this chapter appeared in a preliminary form at IEEE DSOnline [5], UPGRADE-CN'07 [17], and TCSC Doctoral Symposium—CCGrid'07 [18]. This work is supported in part by the Australian Research Council (ARC), through the discovery project grant and Department of Education, Science, and Training (DEST), through the International Science Linkage (ISL) grant. The material in this chapter greatly benefited from discussions with K. H. Kim and Kris Bubendorfer.

References

1. Amini, L., Shaikh, A., and Schulzrinne, H. Effective peering for multi-provider content delivery services. In *Proc. of 23rd Annual IEEE Conference on Computer Communications (INFOCOM'04)*, pp. 850–861, 2004.
2. Arlitt, M. and Jin, T. Workload characterization of the 1998 world Cup Web site. *IEEE Network*, 14:30–37, 2000.
3. Assuncao, M., Buyya, R., and Venugopal, S. Intergrid: A case for internetworking islands of grids, *Concurrency and Computation: Practice and Experience (CCPE)*, Willey press, New York, USA, 2007.
4. Biliris, A., Cranor, C., Douglis, F., Rabinovich, M., Sibal, S., Spatscheck, O., and Sturm, W. CDN brokering. *Computer Communications*, 25(4), pp. 393–402, 2002.
5. Buyya, R., Pathan, M., Broberg, J., and Tari, Z. A case for peering of content delivery networks, *IEEE Distributed Systems Online*, 7(10), 2006.
6. Cardellini, V., Colajanni, M., and Yu, P. S. Efficient state estimators for load control policies in scalable Web server clusters. In *Proc. of the 22nd Annual International Computer Software and Applications Conference*, 1998.
7. Cardellini, V., Colajanni, M., and Yu, P. S. Request redirection algorithms for distributed Web systems. *IEEE Trans. on Parallel and Distributed Systems*, 14(4), 2003.
8. Colajanni, M., Yu, P. S., and Dias, D. M. Analysis of task assignment policies in scalable distributed Web-server systems. *IEEE Trans. on Parallel and Distributed Systems*, 9(6), 1998.
9. Day, M., Cain, B., Tomlinson, G., and Rzewski, P. A Model for Content Internetworking. IETF RFC 3466, 2003.
10. Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman R., and Weihl, B. Globally distributed content delivery. *IEEE Internet Computing*, pp. 50–58, 2002.
11. Freedman, M. J., Freudenthal, E., and Mazières, D. Democratizing content publication with coral. In *Proc. of 1st Symposium on Networked Systems Design and Implementation*, San Francisco, CA, pp. 239–252, 2004.
12. Guo, L., Chen, S., Xiao, Z., and Zhang, X. Analysis of multimedia workloads with implications for internet streaming. In *Proc. 14th international Conference on World Wide Web (WWW)*, pp. 519–528, 2005.
13. Iyengar, A. K., Squillante, M. S., and Zhang, L. Analysis and characterization of large-scale Web server access patterns and performance. *World Wide Web*, 2(1–2), 1999.
14. Mobasher, B., Cooley, R., and Srivastava, J. Automatic personalization based on Web usage mining, *Communications of the ACM*, 43(8), pp. 142–151, 2000.
15. Padmanabhan, V. N. and Sripanidkulchai, K. The Case for Cooperative Networking. In *Proc. of International Peer-To-Peer Workshop (IPTPS02)*, 2002.
16. Pai, V. S., Wang, L., Park, K. S., Pang, R., and Peterson, L. The dark side of the Web: an open proxy's view. In *Proc. of the Second Workshop on Hot Topics in Networking (HotNets-II)*, Cambridge, MA, USA, 2003.
17. Pathan, M., Broberg, J., Bubendorfer, K., Kim, K. H., and Buyya, R. 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 2007)*, Monterey, California, USA, 2007.
18. Pathan, M. and Buyya, R. Economy-based content replication for peering CDNs. TCSC Doctoral Symposium, In *Proc. of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007)*, Brazil, 2007.
19. Pierre, G. and van Steen, M. Globule: A platform for self-replicating Web documents. In *Proc. of the 6th International Conference on Protocols for Multimedia Systems (PROMS'01)*, The Netherlands, pp. 1–11, 2001.
20. Pierre, G. and van Steen, M. Globule: a collaborative content delivery network. *IEEE Communications*, 44(8), 2006.
21. Turrini, E. An architecture for content distribution internetworking. Technical Report UBLCS-2004-2, University of Bologna, Italy, 2004.

22. Verma, D.C., Calo, S., and Amiri, K. Policy-based management of content distribution networks, *IEEE Network*, 16(2), pp. 34–39, 2002.
23. Wang, L., Park, K. S., Pang, R., Pai, V. S., and Peterson, L. Reliability and security in the CoDeeN content distribution network. In *Proc. of Usenix Annual Technical Conference*, Boston, MA, 2004.
24. Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S, Huynh, A., Carlson, M., Perry, J., and Waldbusser, S. Terminology for policy-based management, IETF RFC 3198, 2001.
25. Zhao, W. and Schulzrinne, H. DotSlash: A self-configuring and scalable rescue system for handling Web hotspots effectively. In *Proc. of the International Workshop on Web Caching and Content Distribution (WCW)*, Beijing, China, 2004.