A Case for Peering of Content Delivery Networks

Rajkumar Buyya[1], Al-Mukaddim Khan Pathan[1], James Broberg[2] and Zahir Tari[2]

[1]**Gri**d Computing and **D**istributed **S**ystems (GRIDS) Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne, Carlton, VIC 3053, Australia
{raj, apathan}@csse.unimelb.edu.au

[2]Distributed Systems and Networking Discipline
School of Computer Science and Information Technology
RMIT University, VIC 3083, Australia
{jbroberg, zahirt}@cs.rmit.edu.au

Abstract: The proliferation of Content Delivery Networks (CDN) reveals that existing content networks are owned and operated by individual companies. As a consequence, closed delivery networks are evolved which do not cooperate with other CDNs and in practice, *islands* of CDNs are formed. Moreover, the logical separation between contents and services in this context results in two content networking domains. But present trends in content networks and content networking capabilities give rise to the interest in interconnecting content networks. Finding ways for distinct content networks to coordinate and cooperate with other content networks is necessary for better overall service. In addition to that, meeting the QoS requirements of users according to the negotiated Service Level Agreements between the user and the content network is a burning issue in this perspective. In this paper, we present an open, scalable and Service-Oriented Architecture based system to assist the creation of open Content and Service Delivery Networks (CSDN). These open CSDNs scale and support sharing of resources through peering with other CSDNs.

1. Introduction

Content Delivery Networks (CDN), which evolved first in 1998 [8], replicate content over several mirrored Web servers (i.e., surrogate servers) strategically placed at various locations in order to deal with the *flash crowds* [1]. Geographically distributing the Web servers' facilities is a method commonly used by service providers to improve performance and scalability. A CDN has some combination of a content-delivery infrastructure, a request-routing infrastructure, a distribution infrastructure and an accounting infrastructure. CDNs improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication. Thus CDNs offer fast and reliable applications and services by distributing content to cache servers located close to end-users [9].

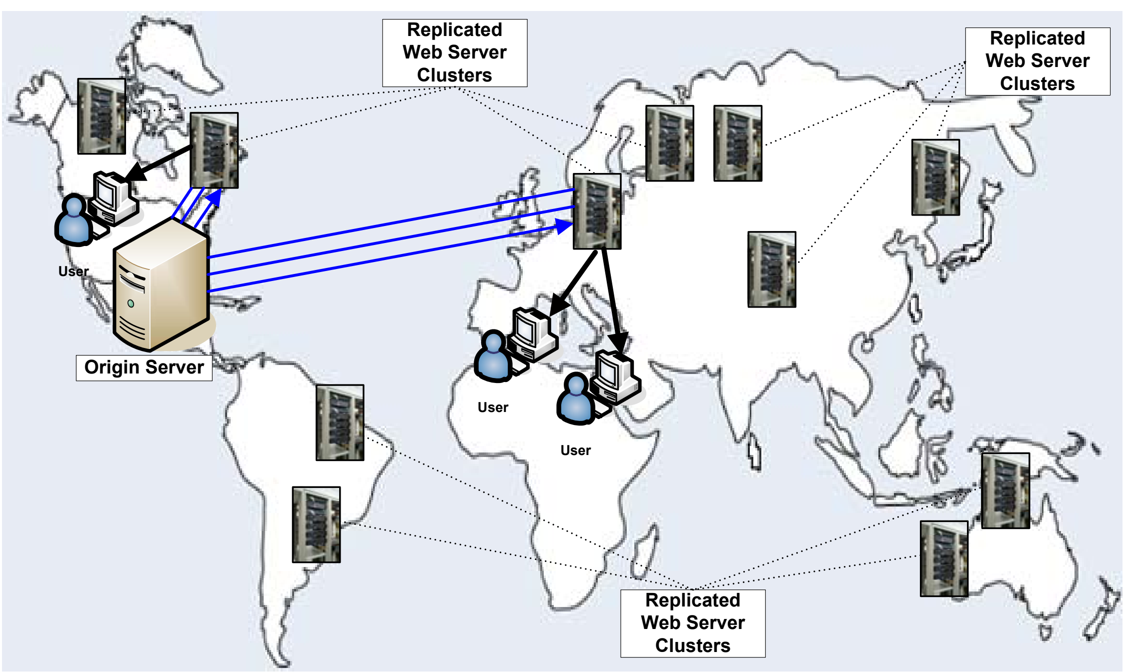

Figure 1: Abstract architecture of a Content Delivery Network (CDN)

Figure 1 shows a typical content delivery environment where the replicated Web server clusters are located at the edge of the network to which the end-users are connected. In such CDN environment, Web content based on user requests are fetched from the origin server and a user is served with the content from the nearby replicated Web server. Thus the users end up communicating with a replicated CDN server close to them and retrieves files from that server.

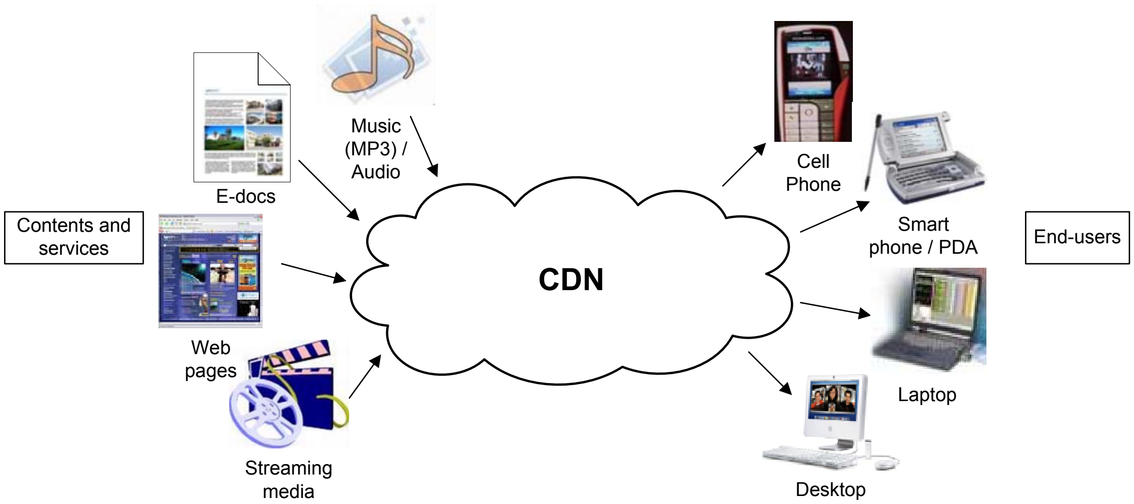

Figure 2: Content/services provided by a CDN

CDN providers ensure the fast delivery of any digital content. They host third-party content including static content (e.g. Static HTML pages, images, documents, software patches etc), streaming media (e.g. audio, real time video etc) and varying content services (e.g. directory service, e-commerce service, file transfer service etc.). The sources of content are large enterprises, Web service providers, media companies, news broadcasters etc. The end-users interact with the CDN specifying the content/service request through cell phone, smart phone/PDA, laptop, desktop etc. Figure 2 depicts the different content/services served by the CDN to end-users.

1.1. Motivations for Peering CDNs

Existing Content Delivery Networks (CDN) are proprietary in nature. They are owned and operated by individual companies. They have created their own closed delivery network, which is expensive to setup and maintain. Although there are many commercial CDN providers, they do not cooperate in delivering content to end-users in a scalable manner. In addition, content providers are typically subscribed to one of the CDN providers and are unable to utilize services of multiple CDN providers at the same time. Such a closed, non-cooperative model results in creation of *islands* of CDNs. Running a global CDN requires enormous amount of capital and labor. To compromise expense, CDN providers should partner together so that each can supply and receive services in a cooperative and collaborative manner that one CDN cannot provide to content providers otherwise.

Commercial CDNs charge customers for their services, and in turn they are bound with strong commitment with their end-users to meet the negotiated Service Level Agreement (SLA). An SLA is a part of contract between the service providers and their consumers. It describes provider's commitment and specifies penalties if those commitments are not met. The objective of a CDN is to satisfy its customers with competitive services. If a particular CDN provider is unable to provide quality service to the end-user requests, it may result in SLA violation and adverse business impact. In such scenarios, one CDN provider should partner with other CDN provider(s) which has caching servers located near to the end-user and serve the user's request, meeting the Quality of Service (QoS) requirements.

Internetworking among CDNs can better be described by illustrating a scenario depicted in Figure 3. Consider that the ICC Cricket World Cup 2007 is to be held in the Caribbean and www.cricinfo.com is supposed to provide live media coverage of the cricket matches from there. As a content provider www.cricinfo.com has exclusive service level agreement with the CDN provider, Akamai. However, Akamai does not have any Point of Presence (POP) in Trinidad and Tobago (one of the Caribbean islands) where most of the cricket matches would be held. Akamai management may decide to place its surrogates in Trinidad and Tobago or they may use their edge servers which are in other Caribbean island (e.g. St. Lucia). In the first case, placement of new surrogates only due to a particular event would cost much for the CDN provider, which may be redundant after the event. On the other hand, Akamai might be at risk of losing reputation due to the inability to provide quality service according to the client requests; which may result in SLA violation and adverse business impact. If another CDN provider, Mirror-Image has its POP in Trinidad and Tobago, Akamai may partner together with Mirror-Image's edge servers in order to provide quality services based on negotiated SLA. Thus, through collaboration with another CDN provider, content networks can emphasize on customer satisfaction meeting end-user's QoS requirements, and hence it can minimize the exposed business impact of service level violations.

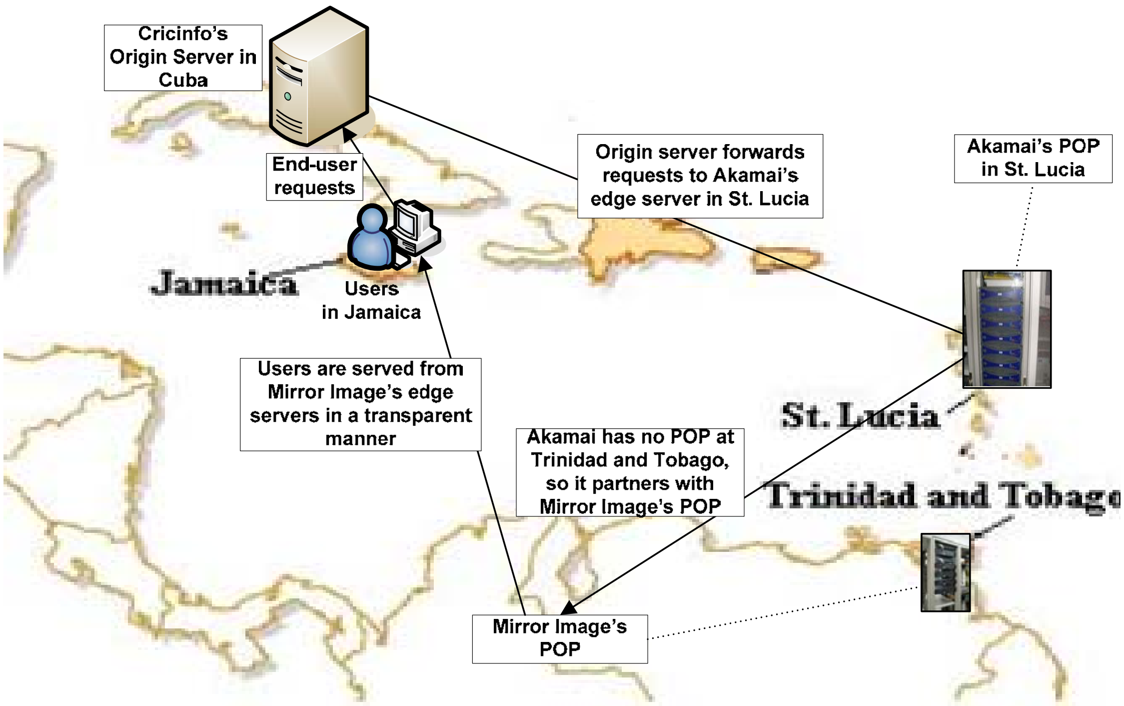

Figure 3: CDN internetworking scenario

To make content services an Internet infrastructure service, vendors have implemented content service networks (CSN) [22], which act as another network infrastructure layer built upon CDNs and provide next generation of CDN services. CSN appears to be another variation of the conventional CDN as it is just another layer of network infrastructure built around CDN. This logical separation between content and services under the 'Content Delivery/Distribution' and 'Content Services' domain, is undesirable considering the on-going trend in content networking. Hence, a unified content network which supports the coordinated composition and delivery of content and services, is highly desirable.

To overcome these problems, in this paper, we present an open, scalable and Service-Oriented Architecture (SOA) based system to assist the creation of open Content and Service Delivery Networks (CSDNs). These CSDNs scale and support sharing of resources through peering with other CSDNs. Web servers within each CSDN are capable of delivering services in order to meet QoS requirements of end-users. We propose Virtual Organization (VO) [3] model for forming CSDNs that not only support sharing of Web servers within their own networks, but also with other CSDNs. The realization of such a system using SOA and VO model makes it open, decentralized, cooperative and coordinated CSDN that scale and deliver services to end users in a timely and reliable manner. In accordance to the proposed architecture, we also define the key areas to be researched in relation to CSDNs by identifying the tickling issues to be addressed in the concerned areas.

1.2. Research contributions

The major research contributions of this paper are:

- Architecture of an open, decentralized, cooperative and coordinated Content and Service Delivery Network (CSDN) that scale and deliver services to end-users in a timely and reliable manner.
- A policy-based framework for enforcing SLA between CSDN and its customers that supports quality delivery of contents and services to the end-users.
- Research issues in accordance to the proposed Content and Service Delivery Networks, which are expected to help the researchers in the concerned field in the following ways:
    - To develop algorithms for large scale coordinated allocation of resources within and across different CSDNs.
    - To establish new theoretical models that analyze the complexity of network congestion problems and deal with requirements of load balancing for heavy-tailed service demands.

- To discover how load balancing approaches complement each other for highly variable task sizes, and
- To build a dynamic approach for estimating bandwidth and network latency (for load balancing purpose), which does not require excessive bandwidth to probe the network.

The rest of the paper is organized as follows: Section 2 provides an insight into the content delivery networks highlighting the existing representative CDNs, basic interactions flow in a typical CDN, and related technical issues; Section 3 presents the model for peering of CDNs, and proposes the architecture of an open, scalable and Service-Oriented Architecture (SOA) based system to assist the creation of cooperative and coordinated CSDNs; Section 4 enlightens the research issues that are to be addressed for the proposed CSDN architecture; Section 5 explores the related work in CDN peering and Section 6 concludes the paper

2. Insight into Content Delivery Networks

2.1. Content Delivery Networks: State of the Art

Within a Content Delivery Network (CDN), end-user's requests are served from the Web servers distributed around the Internet that cache the content originally stored in the origin server. A CDN is different from P2P networks and data grids in the sense that, CDNs are dedicated to caching Web content so that users are able to access it faster; while P2P content sharing networks are vertically integrated to achieve a single goal (for example, file-sharing), and data grids provide a platform through which users can access aggregated computational, storage and networking resources to execute their data-intensive applications on remote data. There are many commercial and academic CDNs at present. Here, only the representative ones have been described. A detailed listing of existing CDNs with brief description can be found in [35].

Akamai (www.akamai.com) technologies [2][5] evolved out of an MIT research effort aimed at solving the *flash crowd* problem, is the market leader in providing content delivery services. It owns more than 18,000 servers over 1000 networks in 70 countries. Akamai's approach is based on the observation that serving Web content from a single location can present serious problems for site scalability, reliability and performance. Hence, a system is devised to serve requests from a variable number of surrogate origin servers at the network edge [6]. To effectively manage dynamic content Akamai supports a new markup language Edge Side Includes [7]. Thousands of organizations have formed trusted relationships with Akamai, improving revenue and reducing costs by maximizing their online business performance.

Mirror Image (www.mirror-image.com) is a global network for online content, application and transaction delivery, provides Content Delivery, Streaming Media, Web Computing and Reporting solutions that offer customers a smarter way to create more engaging Web experiences for users worldwide. It has surrogate servers located in 22 countries around the world. Customers of Mirror Image include Creative, Open Systems, and SiteRock.

Limelight Networks (www.limelightnetworks.com) is a content delivery network for Internet distribution of video, music, games and downloads. Limelight's advanced content delivery network provides high performance delivery of digital media and software via the Internet. Surrogate servers of LimeLight Network are located in 72 locations around the world. It supports distributed on-demand and live delivery of video, music, games and downloads.

Coral (www.coralcdn.org) is a free, peer-to-peer content distribution network designed to mirror Web content. Coral is designed to use the bandwidth of volunteers to avoid slashdotting and to reduce the load on Web sites and other Web content providers in general. During beta testing, the Coral node network is hosted on PlanetLab, a large scale distributed research network of 400 servers, instead of third party volunteer systems. Of those 400 servers, about 275 are currently running Coral. The source code is freely available under the terms of the GNU GPL.

Globule (www.globule.org) is an open-source collaborative content delivery network developed at the Vrije Universiteit in Amsterdam. It is implemented as a third-party module for the Apache HTTP Server that allows any given server to replicate its documents to other Globule servers. This can improve the site's performance, maintain the site available to its clients even if some servers are down, and to a certain extent help to resist the flash crowds and the Slashdot effect.

2.2. Basic interactions in a CDN

Figure 4 provides the high level view of the basic interaction flows among the components in a Content Delivery Network (CDN) environment. Here, discovery.com is the content provider and Akamai is the CDN that hosts the content of discovery.com. The interaction flows are: 1) user requests content from www.discovery.com by specifying its URL in the Web browser. User's request is directed to the origin server of discovery.com; 2) when discovery.com receives a request, its Web server makes a decision to provide only the basic contents (e.g. index page of the site) that can be served from its origin server; 3) to serve the high bandwidth demanding and frequently asked contents (e.g. embedded objects – fresh content, navigation bar, banner ads etc. Figure 5 shows such a Web page which contains the embedded objects served by Akamai CDN), discovery's origin server redirects user's request to the CDN provider (Akamai, in this case); 4) using the proprietary selection algorithm, the CDN provider selects the replica server which is 'closest' to the end-user, in order to serve the requested embedded objects; 5) selected replica server gets the embedded objects from the origin server, serves the end-user requests and caches it for subsequent request servicing.

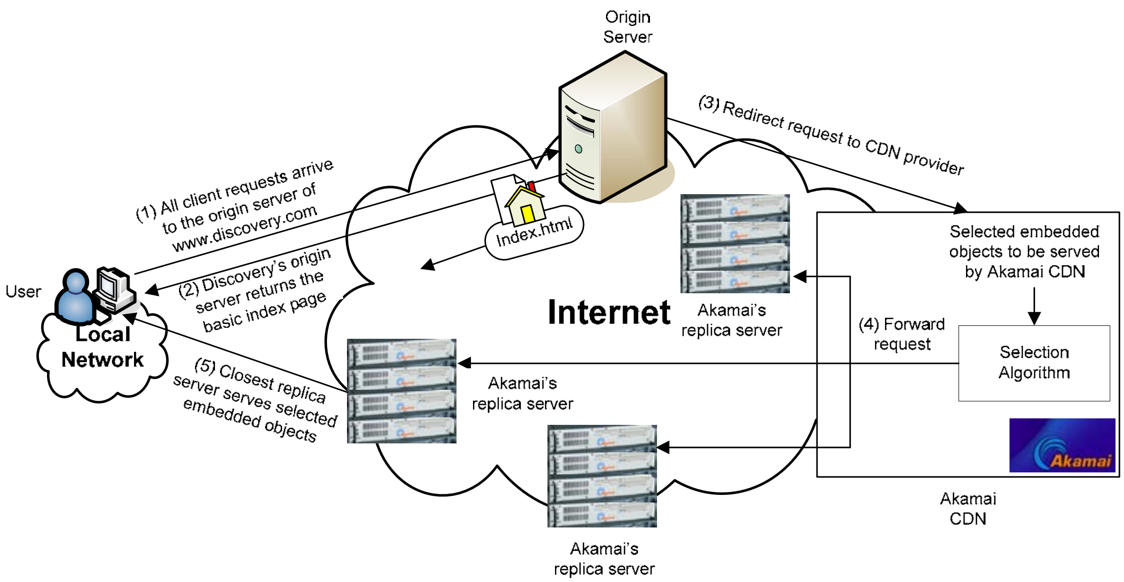

Figure 4: Basic interaction flows in a CDN environment

Figure 5: Typical embedded Web page contents served by Akamai CDN.

2.3. Technical Issues Involved in Content Delivery Networks

Since CDNs spread over multiple geographic location and involves multiple distributed components, there are several technical issues related to CDN content delivery. Those are discussed here:

Placement of surrogates in CDN: Since location of surrogate servers is closely related to the content delivery process, it puts extra emphasis on the issue of choosing the best location for each surrogate. Some theoretical approaches (e.g. minimum K-center problem [10], k-hierarchically well-separated trees (k-HST) [10][11]), some heuristics (e.g. Greedy [12] – which incrementally places replicas, topology-informed placement strategy [14]), server placement algorithms (e.g. Hot Spot [13] and Tree-based [15]), and scalable replica management framework (e.g. Scan [16]) have been developed to model the surrogate server placement problem. For surrogate server placement, the CDN administrators also determine the optimal number of surrogate servers using single-ISP and multi-ISP approach [17].

Selection of content: The choice of content to be delivered to the end-users is important for content selection. Content can be delivered to the customers in full or in partial. In full-site content delivery the surrogate servers perform *entire replication* in order to deliver the total content site to the end-users. In contrast, partial content delivery provides only embedded objects – such as Web page images – from the corresponding CDN.

Request Routing: To select the most appropriate surrogate server for content routing several routing schemes can be used. They are: Global Server Load Balancing (GSLB) [23], client multiplexing [19], URL rewriting, Anycasting and CDN peering. These routing schemes are stated in the following: In *Global Server Load Balancing (GSLB)*, service nodes (which serve contents to end-users) consisting of a GSLB-enabled Web switch and a number of real Web servers are distributed in several locations around the world. The GSLB-enabled switches are responsible for routing the client requests. In *Client multiplexing*, the client obtains the physical addresses of a set of physical replica servers and chooses one to send its request to. In *URL rewriting*, the origin server redirects the clients to different surrogate servers by rewriting the dynamically generated pages' URL links. The *Anycasting* approach, the client's request is sent to one server that serves the anycast address for a group of replicated Web servers. In *CDN peering* approach, Peer-to-peer content networks are formed by symmetrical connections between host computers. Peered CDNs deliver content on each other's behalf.

Outsourcing of content: Given a set of properly placed surrogate servers in a CDN infrastructure and a chosen content for delivery, it is crucial to decide which content outsourcing practice is to follow. Content outsourcing is of three types: cooperative push-based, non-cooperative pull-based and cooperative pull-based. In *Cooperative push-based* approach, content is pushed to the surrogate servers from the origin and each request is directed to the closest surrogate server or otherwise the request is directed to the origin server. In *Non-cooperative pull-based* approach, client requests are directed (DNS redirection, HTTP redirection or URL rewriting [18]) to their closest surrogate servers. If there is a cache miss, surrogate servers pull content from the origin server. The *cooperative pull-based* approach differs from the non-cooperative approach in the sense that surrogate servers cooperates each other to get the requested content in case of cache miss. Using a distributed index, the surrogate servers find nearby copies of requested content and store in the cache.

Content replication and caching: Replicating content is common in large scale distributed environment like CDNs. Commercial CDNs (e.g. Akamai [2][5]) replicate content across the globe for high profile customers such as Symantec and Apple, that need to deliver large volumes of data in a timely manner. Content caching in CDNs can be intra-cluster and inter-cluster basis. For *Intra-cluster* [32] caching of contents either of query-based scheme [27], digest-based scheme [28], directory-based scheme [29], hashing-based scheme [30][31] can be used. *Inter-cluster* [33] content routing is necessary when intra-cluster content routing fails.

Accounting/billing mechanism of CDN: CDN providers charge their customers according to the content delivered (i.e., traffic) by their surrogate servers to the clients. There are technical and business challenges in pricing CDN services. The average cost of charging of CDN services is quite high [20]. The most influencing factors affecting the price of CDN services include: bandwidth cost; variation of traffic distribution; size of content replicated over surrogate servers; number of surrogate servers; reliability and stability of the whole system and security issues of outsourcing content delivery [9]. CDNs support an accounting mechanism that collects and tracks information related to request routing, distribution and delivery [21]. This mechanism gathers information in real time and collects it in for each CDN component. This information can be used in CDNs for accounting, billing and maintaining purposes.

3. A Model for Peering CDNs

In this section, we present the model of an open, scalable, and Service Oriented Architecture (SOA) based system. This system assists the creation of open CSDNs (Content and Service Delivery Networks) that scale and support sharing of resources with other CSDNs through cooperation and coordination. Thus, it helps to overcome the problem of creation of *islands* of CDNs, to ensure the quality of services based on SLA negotiation, and to find a solution to the problem of the logical separation between Content Delivery Network (CDN) and Content

Services Network (CSN). We propose a Virtual Organization (VO) [3] model for forming CSDNs that not only support sharing of Web servers within their own networks, but also with other CSDNs. To enforce quality of service according to the negotiated SLA among the participants of VO-model for CSDN, we also apply the policy framework defined by IETF/DMTF. The architecture of such a system is shown in Figure 6.

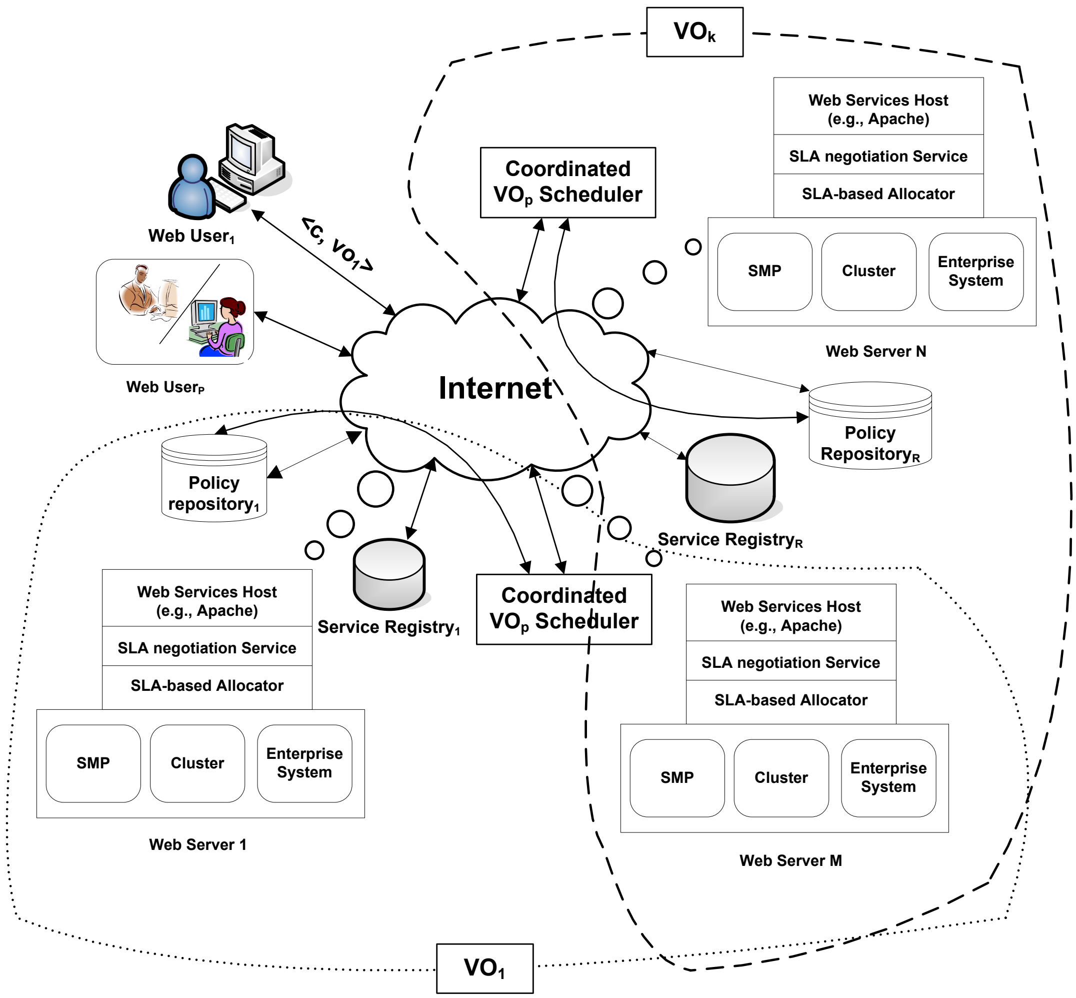

Figure 6: Architecture of open, scalable, Service-Oriented Architecture (SOA) based system to assist the creation of cooperative and coordinated CSDNs (Content and Service Delivery Networks)

3.1. Components of CSDN

Some of the elements of the CSDN architecture presented above are described below:

Web Server: This entity is the most important element of a Content and Service Delivery Network (CSDN). A CSDN is formed consisting of the Web servers using the VO [3] model. Web servers are responsible for storing contents and value-added services as infrastructure services, and delivering them in a reliable and cooperative manner. Web servers within each CSDN are capable of delivering contents and services in order to meet QoS requirements of end-users (i.e. Web users). The structure of a Web server can be divided into two layers: an *overlay layer* and the *core*. In the *overlay layer*, a Web server consists of a Web service host (e.g., Apache and Tomcat), an SLA-negotiation service module and an SLA-based Allocator. The negotiation module with the help of coordinated VO scheduler is responsible for cooperation and coordination with other Web servers (located in local or global CSDN) through SLA-based negotiation. The SLA-based allocator is put in place in order to deliver contents and services based on the negotiated SLAs with other Web servers of local or global CSDNs. The *core* of the Web server consists of high performance computing systems such as SMPs (Symmetric Multiprocessors), Cluster and/or other enterprise systems (e.g. desktop grids). The underlying devices and tools of the Web server are responsible for storing content and services, and assist in responding end-user requests in a timely and reliable manner in order to meet the negotiated QoS requirements. For content and service location and routing, the underlying technologies of the Web servers perform on-demand cooperative caching through coordination with other Web servers. Efficiently balancing the load across different Web servers is critical to produce the required QoS. Hence, Web servers are adopted with appropriate load and resource distribution strategies.

Web user: Web users are the clients who request for content and/or services from the Web servers. The requested content is served to the clients either from the Web server receiving client requests or from any other Web server within the VO, which is *closest* to the Web user.

Coordinated VO scheduler: A coordinated VO scheduler is put in each VO which is responsible for ensuring collaboration and coordination with other CSDNs though policy exchange and scheduling of contents and services.

Service registry: A service registry enables VOs to register their cluster resources. SLA negotiator Service and Allocator module use this service registry to negotiate QoS parameters and resource allocation to maximize the potential of cooperative CSDNs.

Policy repository: A policy repository is used to store the policies generated by the administrators. These policies are a set of rules to administer, manage, and control access to VO resources. They provide a way to consistently mange the components deploying complex technologies.

3.2. Layered Architecture of CSDN

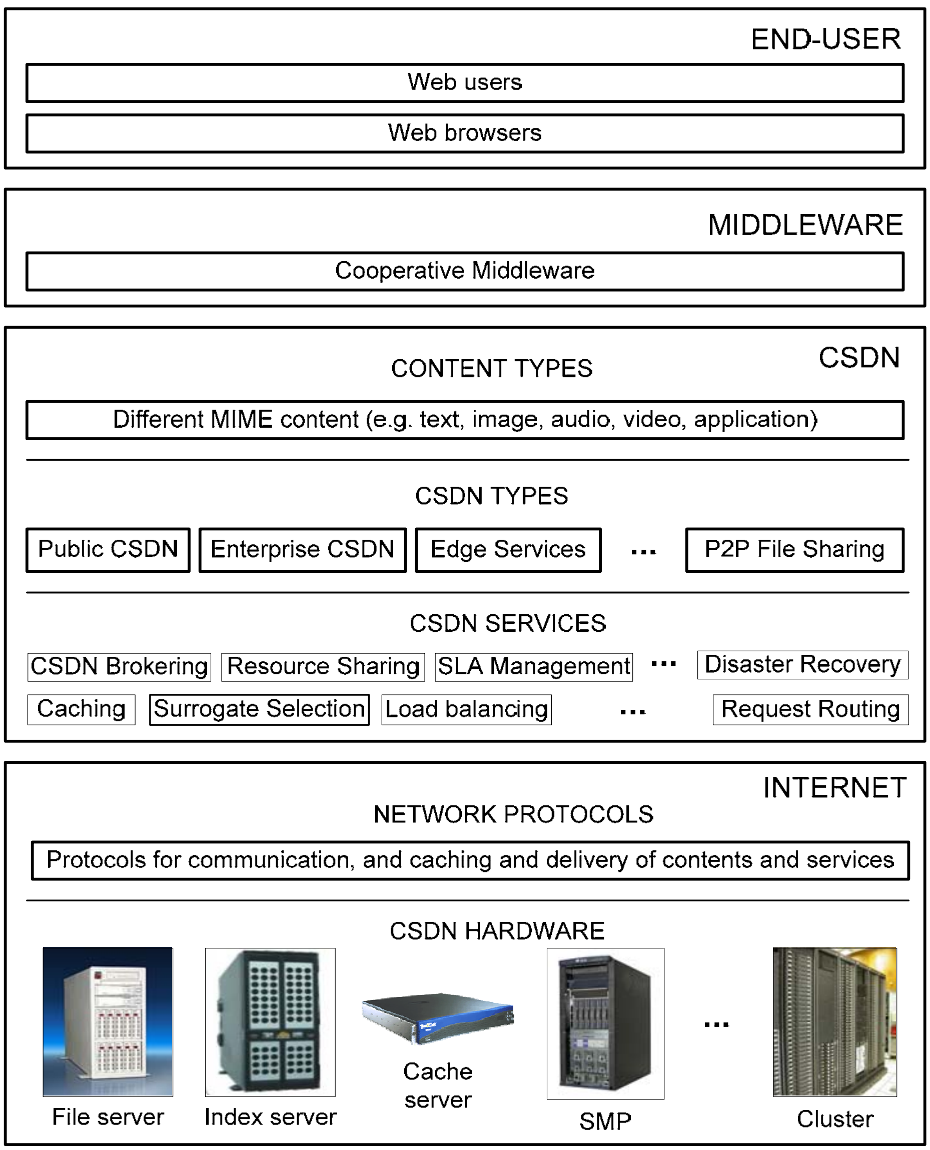

Figure 7: Layered architecture of CSDN

The proposed Content and Service Delivery Network (CSDN) can be described as a layered architecture as shown in figure 7. Here below we describe the layers using a bottom-up approach.

Internet: The base of the Content and Service Delivery Networks (CSDN) is the Internet layer. This layer consists of the hardware for CSDN (e.g. SMP, clusters, file server, index server etc), and core internet protocols (e.g. TCP/UDP, FTP) as well as CSDN specific internet protocols (e.g. Internet cache protocol (ICP), Hypertext Caching Protocol (HTCP), Cache Array Routing Protocols etc.) for communication, caching and delivery of contents and services.

CSDN: This layer consists of the core functionalities of CSDN. It can be divided into three sub-layers: CSDN services, CSDN types and content types. A CSDN provides *core services* such as surrogate selection, request routing, caching, geographic load balancing etc.; and *user specific services* for SLA management, resource sharing, CSDN brokering etc. A CSDN can operate within enterprise domain, it can be for academic and/or public purpose or it can simply be used as edge servers of contents and services. A CSDN provides all types of MIME contents (e.g. text, audio, video etc) to its users.

Middleware: The layer contains the cooperative middleware for providing all technologies to support CSDN coordination.

User: At the top of the CSDN layered architecture, we have the Web users who connect to the CSDN by specifying the URL for content providers (i.e. Web sites) using their Web browsers.

3.3. VO model based CSDN organization

We propose Service-Oriented Architecture (SOA) based Virtual Organization (VO) [3] model for forming CSDNs that not only support sharing of Web servers within their own networks, but also with other CSDNs. A VO can be composed of a group of individuals and/or institutions that come together to share resources with a common purpose. In our proposed VO model, a VO is formed through coordination of Web servers of different CSDNs who have come together to share resources and collaborate on a single goal. Each VO consists of a service registry and a coordinated VO scheduler. The VO model defines the resources available to the participants through the use of the service registry and it enforces the rules (defined by the policies stored in the policy repository) for accessing and using the resources within the VO. The formation of VO may be stand alone or may be composed of a hierarchy of regional, national and international VOs. The realization of a CSDN using SOA and VO model makes it an open, decentralized, cooperative and coordinated CSDN. Thus, it scales and delivers services to end-users in a timely and reliable manner though peering with other CSDNs.

Standardized economy concepts can be deployed within the structure of the proposed VO model for CSDN. In such economy-based VO model, each CSDN provider is both a buyer and seller of its resources and/or services. CSDN providers partner together so that each can supply and receive services that one cannot provide to content providers otherwise. This participation is due to profit motive. CSDN cooperation/internetworking through such participation can be modeled as distributed market economy for buying and selling Web server resources (cache spaces), contents and/or services using similar approach as stated by Waldsburger et al. [36].

3.4. Policy based CSDN model for enforcing SLA

Policy is a set of rules defined by the administrator of a system that specifies the conditions for accessing, sharing and managing the resources within the system. Within our proposed VO-model based CSDN architecture we apply the standard policy framework defined by the IETF/DMTF. The basic policy architecture is shown in figure 8. In the standard policy framework, the *admin domain* refers to an entity which administers, manage and control access resources within the *system* boundary. An administrator uses the *policy management tools* to define the policies to be enforced in the system. The *policy enforcement points (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 *policy decision point* 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.

A policy in the context of proposed Content and Service Delivery Networks (CSDNs) would be statements that are agreed upon by the participants within the VO model based CSDNs. These statements define what type of contents and services can be moved out to a CSDN node, what resources can be shared between the VO participants, what measures are to be taken to ensure service quality based on negotiated SLA, and what type of programs/data must be executed at the origin servers.

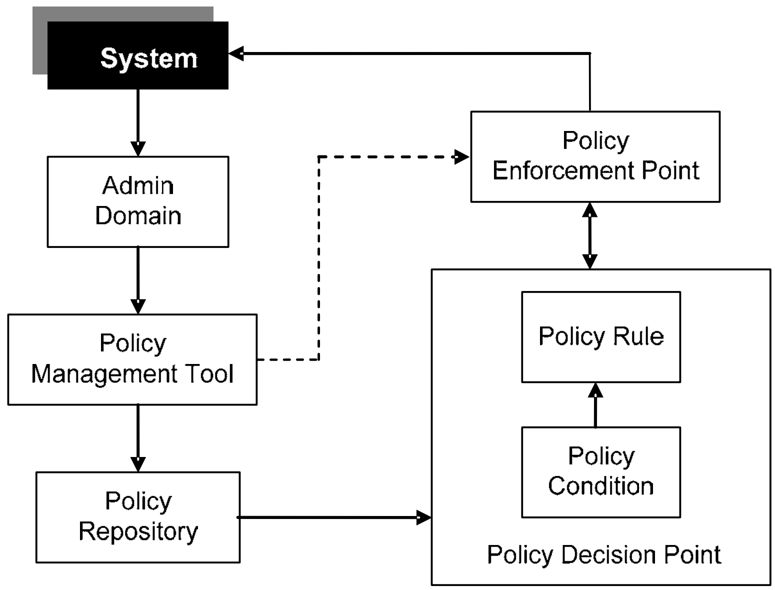

Figure 8: Basic policy framework

The proposed model for CSDN in figure 6 can be mapped to the basic policy framework in figure 8. The policy repository in figure 6 is responsible for storing policies generated by the policy management tool used by the VO administrator. The distribution network and the Web server components (i.e. SLA negotiation service, SLA based Allocator) are the instances of policy enforcement points (PEPs), which enforce the CSDN policies stored in the repository. Each coordinated VO scheduler is a policy agent, which is responsible for retrieving CSDN policies from policy repository. VO schedulers are the instances of the policy decision points (PDPs), which determine the set of policies to be enforced at the time of coordination and cooperation among VOs. The policy management tool is administrator dependent and it is not shown in figure 6. A direct benefit of using such policy-based architecture is to reduce the cost of operating of CSDNs and to meet end-user QoS requirements according to negotiated SLA.

4. Research Issues for CSDNs

We have already discussed some of the technical issues related to conventional Content Delivery Networks (CDN). Since Content and Service Delivery Networks (CSDN) spans across heterogeneous and distributed environment involving multiple CDNs, proper deployment of it exhibits unique research challenges. In this section, we present the unique issues that are to be addressed for CSDNs.

Load Distribution for cooperative CSDNs: The load distribution strategy for cooperative CSDNs to be developed includes four important components: *task assignment*, *traffic congestion*, *load dissemination*, and *cooperative caching*. Coordination among these core issues is another important consideration for successful exploitation of load distribution strategy.

Task assignment issue: The issue of task assignment for geographically distributed Web servers is significantly more challenging. It requires consideration of variable issues like end-user's location, the server loads and the link utilization between the end-user and server in addition to the size of the task. If this issue is to be solved, investigation of a geographical load distribution technique is needed which takes the best features of size-based approaches while addressing the need to handle dynamically changing conditions, such as *flash crowds* and other unpredictable events.

Traffic congestion issue: Significant research [4][24] make it clear that Web requests follow a more bursty pattern rather than a standard Poisson process. So, new approach is required to model CSDNs. CSDNs can be modeled using approaches like Markov Arrival Process (MAP) [25] model. Thus an algorithmic approach can be used based on a combination of recursive and matrix-geometric methods.

Load dissemination issue: To deal with the 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. A new approach can be developed to model a unified load index, which will incorporate heterogeneous information from a variety of sources and under conditions that change continuously.

Cooperative caching issue: In many distributed environments, it is a common issue to cache content at strategic locations for redundancy and performance reasons. The issue of content and service replication and caching is critical to the success of peering of CSDNs.

The following questions are to be addressed for distributing loads among peering CSDNs:

- How to deduce a dynamic task assignment strategy that calculates ideal parameters for task assignment during runtime?
- What algorithms and techniques are to be used for efficient processing of client requests submitted across the Internet?
- How do Web servers of CSDN(s) participate in cooperative caching to provide a satisfactory solution to all parties?
- How to ensure that the resulted cooperative caching ensures reduced bandwidth consumption (fewer requests and responses that need to go over the network) and reduced server load (fewer requests for a server to handle)?
- How to predict cache behavior based on the several parameters such as client population size, rate of change of contents, Zipf popularity [34] parameters etc.?
- What measures can be taken to ensure that the cached objects are not out-of-date? How to deal with uncacheable objects?

Coordination of CSDNs: Any solution to the above four core technical issues of load distribution must be coordinated amongst all participants of a CSDN in order to provide high performance and QoS. A cooperative middleware must be developed to enable the correct execution of the algorithms developed to address each core issue. A service registry needs to be developed to enable VOs to register their cluster resources. SLA negotiator Service and Allocator module are also needed to negotiate QoS parameters and resource allocation to maximize the potential of cooperative CSDNs.

Related to this issue, the following questions are to be addressed:

- What kind of coordination mechanisms need to be in place to allow scalability and growth of cooperative CSDNs?
- How to ensure the effectiveness of the coordination among the CSDNs while it is well established that cooperation among a small group of peers is effective?

Service and policy management: Content management in CSDNs is expected to be highly motivated by the user preferences. Hence, a comprehensive model for managing the distributed contents and services in CSDNs is crucial to avail end-user's preferences. To address this issue, contents can be personalized to meet specific user's (or a group of users) preferences. Like Web personalization [26], user preferences can be automatically learned from content request and usage data by using data mining techniques. Data mining over CSDN can exploit significant performance improvement through dealing with proper management of traffic, pricing and accounting/billing in CSDNs.

In this field, the following questions need to be addressed:

- How to manage large scale computational resources, contents and network services throughout the entire lifecycle of creation, acquisition, migration/replication to publication and termination?
- How to make value-added services an infrastructure service that is accessible to the customers?
- How to deliver appropriate contents and services to the interested clients in a timely and cost-effective manner?
- What types of Service Level Agreements (SLA) are to be negotiated and enforced among CSDN participants?
- What policies can be generated to enforce SLA negotiation? How these policies can be managed?
- How to meet the security requirements such as authentication, signing, encryption, access control, auditing? How to perform resource control to ensure user's privacy?

Pricing of contents and services in CSDNs: A potential problem for collaborative CSDNs can be free-riding; users of one CSDN are expected to take advantage of the resources (i.e., contents and services) offered to the network by other CSDNs without contributing their own resources. Free riding can significantly degrade the performance of content and service distribution and many CSDN may introduce mechanisms to discourage free-riding. Successful CSDN coordination requires that service and/or content providers deploy proper pricing, billing and management systems that let their networks exchange traffic and enforce SLA.

The following questions are to be addressed in this context:

- How to decide service pricing models in commercial CSDN environment, when considering local pricing, a competitive market and coordination among CSDNs? 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 to specify them?

- How do resource (i.e., content and service) providers achieve maximum profit in a competitive environment using such pricing mechanisms, yet maintaining the equilibrium of supply and demand?

5. Related Work

Peering of Content Delivery Networks is gaining popularity among researchers of the scientific community. Several projects/works are being conducted for finding ways to peer the CDNs for better overall performance. Some of those efforts are presented here.

The internet draft by IETF [21] proposes a Content Distribution Internetworking (CDI) Model, which allows the CDNs to have a means of affiliating their delivery and distribution infrastructure with other CDNs who have content to distribute. Thus, content internetworking allows different content networks to share resources in large scale and/or reach than they could independently achieve. 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 peering 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. The CDI proposal does not address the issue of policy management for enforcing SLAs among the peering CDNs.

An architecture for Content Distribution Internetworking (CDI) is presented in [37]. It discusses the design, implementation and evaluation of a protocols architecture that can effectively support the interoperation and cooperation of separately administered Content Delivery Networks (CDNs). A CDI allows every CDN in that CDI both to augment the number of potential content consumers and to get the content access faster (i.e. reduce user response time). This work shows that P2P models are not suitable for CDI construction since it does not provide significant benefits. Hence, some semi-centralized approach based on a star topology is used where an authoritative CDN is responsible for a particular group of content requests and the request is forwarded by this CDN to other CDN which will serve the requests. Thus, performance data has to pass only one hop since all CDNs forward their performance data to those CDNs to which they are federated. The protocol for CDI is termed as RIEPS (Routing IEP for Star topology). The main drawback of this protocol is – being a point-to-point protocol, if one end-point is down the connection remains interrupted until that end-point is restored.

The Content Internetworking Router (CiRouter) [38], implemented by FastTide allows clients to choose from where it wants to retrieve content. When CiRouter receives a request for an HTML page, it delivers the performance measurement and selection code to the client originating the request in order to enable the client to test several delivery alternatives. The client runs the code performing the response time measurements, analyzes output to choose a CDN and reports the result to the CiRouter. Relying on the client results, CiRouter modifies the URLs of the embedded objects in the requested HTML page at runtime and delivers the modified content to the client from the selected (by client) CDN. The disadvantage is this approach is – client has to be modified to run the performance code provided by the CiRouter.

CDN brokering introduced in [39] provides CDNs a way to interoperate by allowing one CDN to intelligently redirect clients dynamically to other CDNs in that domain. It uses techniques offered by the DNS to redirect client requests to the best CDN. The client DNS request is forwarded to the brokering CDN server (BDS) that is authoritative for a particular domain. This DNS-based system is called as Intelligent Domain Name Sever (IDNS). IDNS responds to DNS requests intelligently based on a dynamic, load-sensitive configuration rather than using static information. The drawback is that, mechanism for IDNS is proprietary in nature and might not be suitable for a generic CDI architecture.

Content Network Advertisement Protocol (CNAP) [40] is an advertisement protocol which is designed to facilitate the interconnection of separately administered CDNs. It is intended to communicate information for the purpose of performing request routing decisions between interconnected CDNs. CNAP is not a routing protocol but can be used to exchange information that may be used for inter-CN request-routing decisions. The Internet draft illustrating the CNAP does not specify the topology of the overlay network that the protocol requires.

6. Summary and Future Directions

In this paper, we present an open, scalable and Service-Oriented Architecture (SOA) based system to assist the creation of open Content and Service Delivery Networks (CSDNs). Thus we address the issue of content networks scaling and support sharing of resources with other CSDNs. We propose a VO model for forming CSDNs and a policy framework within the VO model. It will support management and sharing of content and services not only within their own networks, but also with other CSDNs. Delivery of resources (i.e., contents and services) in such an environment will meet QoS requirements of end-users according to the negotiated SLA. The realization of such a system is expected to serve the end users in a timely and reliable manner. In our work, we also define the key areas to be researched in relation to CSDNs in order to enable the vision of cooperative CSDNs by striking a balance between satisfying transient and fluctuating user demand, while being fair to all clients who have content hosted. It is expected that the proposed structure of Content and Service Delivery Network (CSDN) will be a timely contribution to the ongoing trend of Content Networking. Work is in progress on cooperating CDNs with joint collaboration between the GRIDS laboratory, University of Melbourne and the DSN laboratory, RMIT University, Australia.

References

[1] M. Arlitt and T. Jin, *A Workload Characterization Study of 1998 World Cup Web Site*, IEEE Network, pp. 30-37, May/June 2000.

[2] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Weihl, *Globally Distributed Content Delivery*, IEEE Internet Computing, pp. 50-58, September/October 2002.

[3] L. Camarinha-Matos, H. Afsarmanesh, C. Garita and C. Lima, *Towards an Atchitecture for Virtual Enterprises*, Journal of Intelligent Manufacturing, 9(2), 1998.

[4] M. Zukerman, T. D. Neame and R. G. Addie, *Internet Traffic Modeling and Future Technology Implications*, IEEE INFOCOM, 2003.

[5] Akamai Technologies, Inc. www.akamai.com, 2006

[6] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine and D. Lewin, *Consistent Hasing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*, In Proceedings of 29th Annual ACM Symposium on Theory of Computing, ACM Press, NY, 1997, pp. 654-663.

[7] Oracle Corporation, Akamai Technologies, Inc., *Edge Side Includes*, www.esi.org

[8] F. Douglis, M. F. Kaashoek, *Scalable Internet Services*, IEEE Internet Computing, vol. 5, no. 4, 2001, pp. 36-37.

[9] G. Pallis and A. Vakali, *Insight and Perspectives for Content Delivery Networks*, Communications of the ACM, vol. 49, no. 1, ACM Press, NY, USA, January 2006. pp. 101-106.

[10] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, *On the placement of Internet Instrumentation*, In Proceedings of IEEE INFOCOM conference, pp. 295-304, Tel-Aviv, Israel, March 2000.

[11] Y. Bartal, *Probabilistic Approximation of Metric Space and its Algorithmic Applications*, In Proceedings of 37th Annual IEEE Symposium on Foundations of Computer Science, October 1996.

[12] P. Krishnan, D. Raz, Y. Shavitt, *The Cache Location Problem*, IEEE/ACM Transaction on Networking, 8(5), 2000.

[13] L. Qiu, V. N. Padmanabhan, G. M. Voelker, *On the Placement of Web Server Replicas*, In Proceedings of IEEE INFOCOM conference, pp. 1587-1596, Anchorage, Alaska, USA, April 2001.

[14] S. Jamin, C. Jin, A. R. Kure, D. Raz and Y. Shavitt, *Constrained Mirror Placement on the Internet*, In Proceedings of IEEE INFOCOM Conference, Anchorage, Alaska, USA, April 2001.

[15] B. Li, M. J. Golin, G. F. Italiano, D. Xin, K. Sohraby, *On the Optimal Placement of Web Proxies in the Internet*, In Proceedings of IEEE INFOCOM Conference, pp. 1282-1290, NY, USA, March 1999.

[16] Y. Chen, R. H. Katz and J. D. Kubiatowicz, *Dynamic Replica Placement for Scalable Content Delivery*, In Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 02), LNCS 2429, Springer-Verlag, pp. 306-318, 2002.

[17] A. Vakali and G. Pallis, *Content Delivery Networks: Status and Trends*, IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003.

[18] N. Fujita, Y. Ishikawa, A. Iwata and R. Izmailov, *Coarse-grain Replica Management Strategies for Dynamic Replication of Web Contents*, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 45, issue 1, pp. 19-34, May 2004.

[19] G. Peng, *CDN: Content Distribution Network*, Technical Report TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, NY 2003. http://citeseer.ist.psu.edu/peng03cdn.html

[20] J. Kangasharju, J Roberts and K. W. Ross, *Object Replication Strategies in Content Distribution Networks*, Computer Communications 25(4), pp. 367-383, March 2002.

[21] M. Day, B. Cain, G. Tomlinson and P. Rzewski, *A Model for Content Internetworking (CDI)*, Internet Engineering Task Force RFC 3466, February 2003. www.ietf.org/rfc/rfc3466.txt

[22] W. Y. Ma, B. Shen and J. T. Brassil, *Content Services Network: Architecture and Protocols*, In Proceedings of 6th International Workshop on Web Caching and Content Distribution(IWCW6), 2001.

[23] M. Hofmann and L. R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 129-134, 2005.

[24] V. Cardellini, M. Colajanni and P. S. Yu, *Request Redirection Algorithms for Distributed Web Systems*, IEEE Transaction on Parallel and Distributed Systems, 14(4), pp. 355-368, 2003.

[25] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker, July 1989.

[26] B. Mobasher, R. Cooley and J. Srivastava, *Automatic Personalization Based on Web Usage Mining*, Communications of the ACM, 43(8), pp. 142-151, August 2000.

[27] D. Wessels and K. Claffy, *Internet Cache Protocol (ICP), version 2*, Internet Engineering Task Force RFC 2186, September 1997. www.ietf.org/rfc/rfc2186.txt

[28] A. Rousskov and D. Wessels, *Cache Digests*, Computer Networks and ISDN Systems, vol. 30, no. 22-3, pp. 2155-2168, November 1998.

[29] S. Gadde, M. Rabinovich and J. Chase, *Reduce, Reuse, Recycle: An Approach to Building Large Internet Caches*, In Proceedings of 6th Workshop on Hot Topics in Operating Systems, pp.93-98, April 1997.

[30] V. Valloppillil and K. W. Ross, *Cache Array Routing Protocol v1.0*, Internet Draft, February 1998.

[31] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins and Y. Yerushalmi, *Web Caching with Consistent Hashing*, Computer Networks, 31(11-16), pp. 1203-1213, 1999.

[32] J. Ni, D. H. K. Tsang, *Large Scale Cooperative Caching and Application-level Multicast in Multimedia Content Delivery Networks*, IEEE Communications Magazine, vol. 43, issue. 5, pp. 98-105, May 2005.

[33] J. Ni, D. H. K. Tsang, I. S. H. Yeung and X. Hei, *Hierarchical Content Routing in Large-Scale Multimedia Content Delivery Network*, In Proceedings of IEEE International Conference on Communications, 2003 (ICC '03), vol. 2, pp. 854-859, May 2003.

[34] W. Li, *Zipf's Law*, Rockefeller University, New York, www.nslij-genetics.org/wli/zipf/

[35] Al-Mukaddim Khan Pathan, *Content Delivery Networks (CDN) Research Directory*, http://www.cs.mu.oz.au/~apathan/CDNs.html

[36] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart and W. S. Stornetta *Spawn: A Distributed Computational Economy*, IEEE Transactions on Software Engineering 18, 2, pp. 103-117, February 1992.

[37] E. Turrini, *An Architecture for Content Distribution Internetworking*, Technical Report UBLCS-2004-2, University of Bologna, Italy, March 2004.

[38] I. Chaudhri, CiRouter Whitepaper, July 2001.

[39] A. Biliris, C. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatscheck and W. Sturm, *CDN brokering*, Computer Communications, Volume 25, Issue 4, pp. 393-402, March 2002.

[40] B. Cain, O. Spatscheck, K. van der Merwe, L. Amini, A. Barbir, M. May and D. Kaplan, *Content Network Advertisement Protocol (CNAP)*, IETF draft, July 2002.