**Content Delivery Networks (CDN)**

**By: Kapil Phulwani , Aakash Jain , Jayant Jain, Roshan Jha and**

**Himanshu Khunger**

**CONTENTS**

1. **Introduction**
2. **Features**
3. **Benefits**
4. **Architecture**
5. **Contents and Services**
6. **Layered Architecture**
7. **Interaction and communication protocols**
8. **Response and request Routing**
9. **Best Practices or Recommendations**

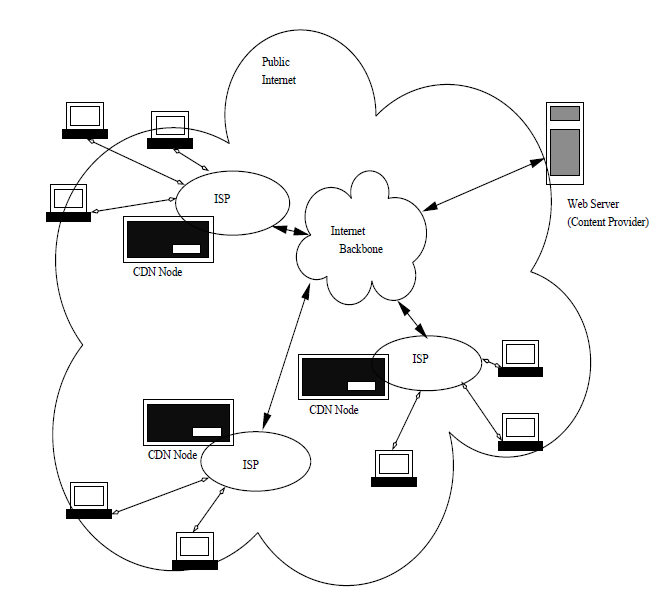
**Introduction:** The growth of the World Wide Web and new modes of Web services have triggered an escalating increase in Web content and Internet traﬃc . The Web content consists of static content (e.g. Static HTML pages, images, documents, software patches), streaming media (e.g. audio, real time video) and varying content services (e.g. directory service, e-commerce service, ﬁle transfer service). As the Web content and the Internet traﬃc increases, individual Web servers ﬁnd it diﬃcult to fulfill to the needs of end-users. In order to store and serve huge quantities of Web content, Web server farms :- a cluster of Web servers functioning as a single unit are introduced.

Even those Web server farms ﬁnd it diﬃcult to deal with ﬂash crowds - large number of simultaneous requests for a popular content - that are frequently experienced in Web traﬃc. Moreover, those server farms are geographically distant from the end-users in most of the cases. The non-proximity of the Web servers to the end-users badly aﬀect the response time of the Web requests, resulting in undesirable delays.

The basic idea is to improve the performance and scalability of content retrieval by geographically distributing a network of Web servers around the globe and allowing several content providers to host their content in those servers. . It allows a number of content providers to upload their Web content into the same network of Web servers (also called, CDN servers) and thereby to reduce the cost of content replication and distribution.

In a typical CDN environment(As shown in the Figure), the replicated Web server clusters are located at the edge of the network to which the end-users are connected. The end-users interact with the CDN specifying the content-service request through cell phone, PDA, laptop, desktop etc. The 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 retrieve ﬁles from that server.

The below figure shows the **Native CDN’s and Web Content Distribution.**



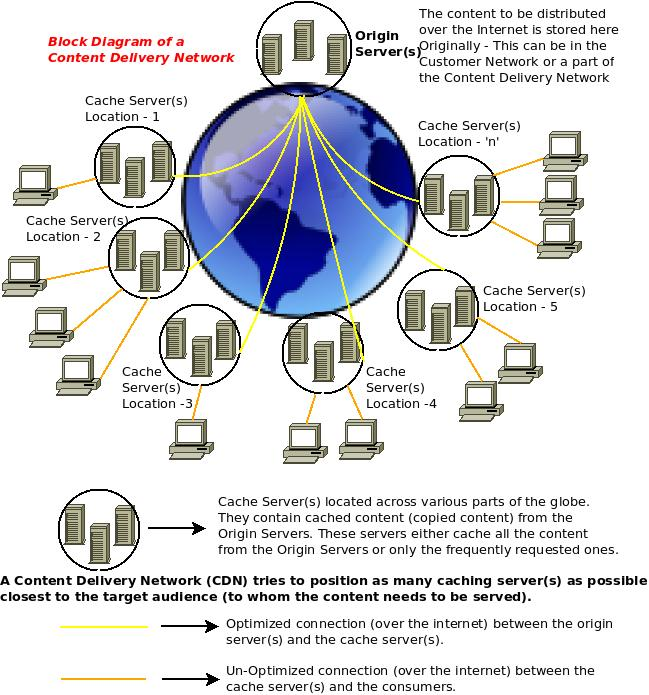
**Features:** Features of CDN’s are :-

* **Availability -** System Architecture without single points of failure , in service software update and/or infrastructure extensions .
* **Operation -** because the quality and speed of delivery depends on the distance between CDN server and the Consumer, delivery caches need to be distributed in close proximity to end-users .
* **Performance -** High def streaming video requires sustained high throughput and low latency .
* **Resilience -** Ability to mitigate congestion issues and resume interrupted downloads .
* **Security -** integrity and security measures to protect digital rights and prevent piracy .
* **Efficiency -** Intelligent content replication based on dynamic demand and content popularity .

**Benefits:** Several benefits CDN’s provide are :-

* **Different Domains -** Browsers limit the number of concurrent connections (file downloads) to a single domain. Most permit four active connections so the fifth download is blocked until one of the previous files has been fully retrieved. You can often see this limit in action when downloading many large files from the same site **.** CDN files are hosted on a different domain. In effect, a single CDN permits the browser to download a further four files at the same time.
* **Files may be Pre-Cached -** Query is ubiquitous on the web. There’s a high probability that someone visiting your pages has already visited a site using the Google CDN. Therefore, the file has already been cached by your browser and won’t need to be downloaded again.
* **High Capacity Infrastructures -** You may have great hosting but I bet it doesn’t have the capacity or scalability offered by Google, Microsoft or Yahoo. The better CDNs offer higher availability, lower network latency and lower packet loss.
* **Distributed Data Centres -** If your main web server is based in Dallas, users from Europe or Asia must make a number of trans-continental electronic hops when they access your files. Many CDNs provide localized data centers which are closer to the user and result in faster downloads.
* **Built-in Version Control -** It’s usually possible to link to a specific version of a CSS file or JavaScript library. You can often request the “latest” version if required.
* **Usage Analytics -** Many commercial CDNs provide file usage reports since they generally charge per byte. Those reports can supplement your own website analytics and, in some cases, may offer a better impression of video views and downloads.
* **Boosts Performance and saves Money -** A CDN can distribute the load, save bandwidth, boost performance and reduce your existing hosting costs — often for free.

**Architecture for Content Delivery Networks (CDN):**

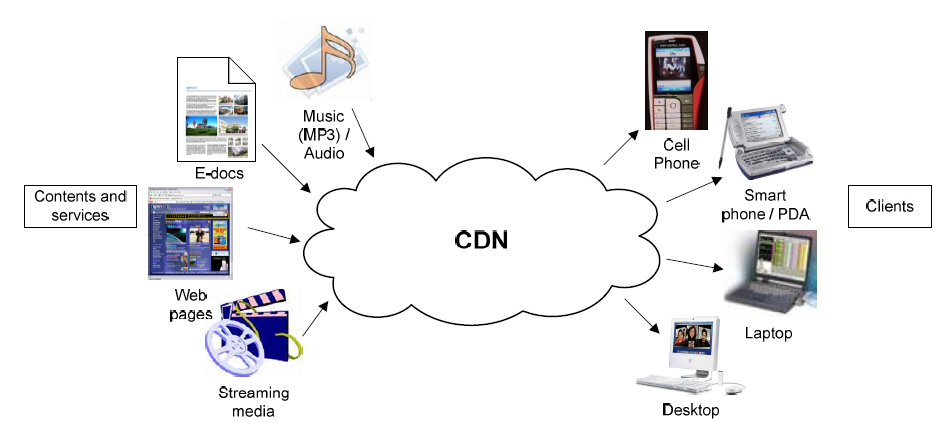


Various Web 2.0 technologies have transformed the Internet and the Internet is now enabling content to be delivered in various forms – Text, Images, Flash, Audio, Video, etc. When companies need to serve such diverse forms of content, most of the time, they distribute them by hosting them in servers in their data center(s). But the consumer pattern is not always uniform. There may be sudden bursts of traffic requesting web based content from consumers located around the globe. To accommodate for the demand, companies need to estimate and provision additional servers in quantities enough for handling peak loads. Moreover, on the Internet, there is always a factor of RTT (Round Trip Time) and Packet Loss that needs to be considered, and these parameters have a higher effect on the consumers who are located at greater distances from the servers.

It is to accommodate for these limitations, that Content Delivery Networks were created. A Content Delivery Network is a network of servers hosted by a service provider in multiple locations of the world (usually shared with multiple customers) so that the content could always be served from a server that is nearest to the consumer requesting for it. And besides, since multiple servers are used, the load is distributed and consumers get better quality content, faster.

A Content Delivery Network (CDN) consists of two components: The **Origin Server(s)** – where the content to be distributed over Internet is originally stored & **Cache Server(s)** – where the content is duplicated. There is generally one Origin Server (either in the customer’s data center or in the cloud, with the content delivery network service provider – for example) and many cache servers (in multiple locations across the globe) so that, when a consumer is requesting a particular content on the Internet, it can be served by a cache server nearest to the consumer’s geographical location if the content is available there. Otherwise, cache server fetches the content quickly from the origin server after protocol/route optimizations

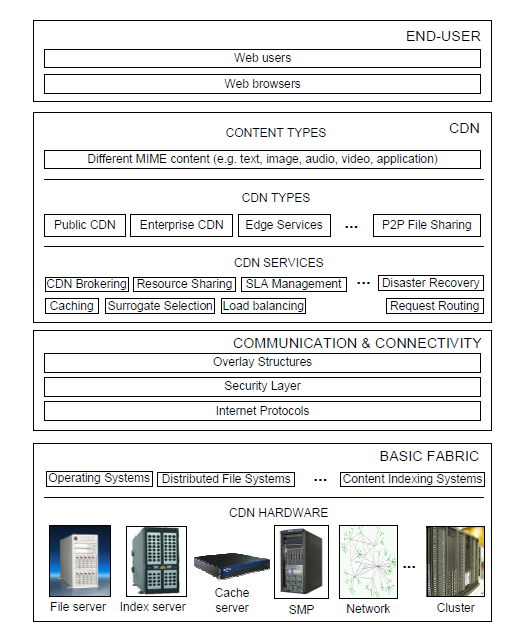
**Content and Services Provides by CDN:**

.

CDN providers charge their customers according to the content delivered (i.e. traffic) to the end-users by their surrogate servers. CDNs support an accounting mechanism that collects and tracks client usage information related to request-routing, distribution and delivery . This mechanism gathers information in real time and collects it for each CDN component. This information can be used in CDNs for accounting, billing and maintenance purposes. The average cost of charging of CDN services is quite high , often out of reach for many small to medium enterprises (SME) or not-for-profit organizations. 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

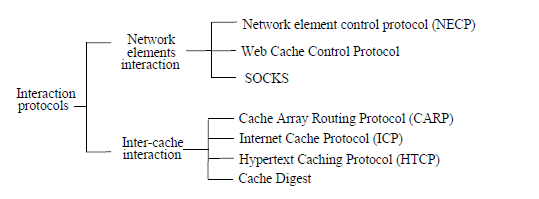
**Layered Architecture of CDN:**



The architecture of content delivery networks can be presented according to a layered approach. We present the layered architecture of CDNs, which consists of the following layers: Basic Fabric , Communication & Connectivity , CDN and end-user . The layers are defined in the following as a bottom up approach :-

* Basic Fabric is the lowest layer of a CDN. It provides the infrastructural resources for its formation. This layer consists of the distributed computational resources such as SMP, clusters, file servers, index servers, and basic network infrastructure connected by high-bandwidth network. Each of these resources runs system software such as operating system, distributed file management system, and content indexing and management systems.
* Communication & Connectivity layer provides the core internet protocols (e.g. TCP/UDP, FTP) as well as CDN specific internet protocols (e.g. Internet Cache Protocol (ICP), Hypertext Caching Protocol (HTCP), and Cache Array Routing Protocols (CARP), and authentication protocols such as PKI (Public Key Infrastructures), or SSL (Secure Sockets Layer) for communication, caching and delivery of content and/or services in an authenticated manner. Application specific overlay structures provide efficient search and retrieval capabilities for replicated content by maintaining distributed indexes.
* CDN layer consists of the core functionalities of CDN. It can be divided into three sub-layers: CDN services, CDN types and content types. A CDN provides core services such as surrogate selection, request-routing, caching and geographic load balancing, and user specific services for SLA management, resource sharing and CDN brokering. A CDN can operate within an enterprise domain, it can be for academic and/or public purpose or it can simply be used as edge servers of content and services. A CDN can also be dedicated to file sharing based on a peer-to-peer (P2P) architecture. A CDN provides all types of MIME content (e.g. text, audio, video etc) to its users.
* End-users are at the top of the CDN layered architecture. In this layer, we have the Web users who connect to the CDN by specifying the URL of content provider’s Web site, in their Web browsers.

**Interaction and Communication Protocols:**



These are some interaction protocols which are implemented in Content delivery networks architecture .

Such interactions can be broadly classified into two types: **interaction among network elements** and **interaction between caches**. This Figure shows various interaction protocols that are used in a CDN for interaction among CDN elements. Examples of protocols for network element interaction are Network Element Control Protocol (NECP), Web Cache Coordination Protocol and SOCKS. On the other hand, Cache Array Routing Protocol (CARP), Internet Cache Protocol (ICP), Hypertext Caching protocol (HTCP), and Cache Digest are the examples of inter-cache interaction protocols. These are explained below :-

**NECP:** The Network Element Control Protocol (NECP) is a lightweight protocol for signaling between servers and the network elements that forward traffic to them. The network elements consist of a range of devices, including content-aware switches and load-balancing routers. NECP allows network elements to perform load balancing across a farm of servers and redirection to interception proxies . Rather, this protocol provides methods for network elements to learn about server capabilities, availability and hints as to which flows can and cannot be served. Hence, network elements gather necessary information to make load balancing decisions .

**WCCP:** The Web Cache Coordination Protocol (WCCP) specifies interaction between one or more routers and one or more Web-caches. It runs between a router functioning as a redirecting network element and interception proxies. The purpose of such interaction is to establish and maintain the transparent redirection of selected types of traffic flow through a group of routers. The selected traffic is redirected to a group of Web- caches in order to increase resource utilization and to minimize response time. WCCP allows one or more proxies to register with a single router to receive redirected traffic.

**SOCKS:** The SOCKS protocol is designed to provide a framework for client-server applications in both the TCP and UDP domains to conveniently and securely use the services of a network firewall . The protocol is conceptually a "shim-layer" between the application layer and the transport layer, and as such does not provide network-layer gateway services, such as forwarding of ICMP messages. When used in conjunction with a firewall, SOCKS provides an authenticated tunnel between the caching proxy and the firewall .

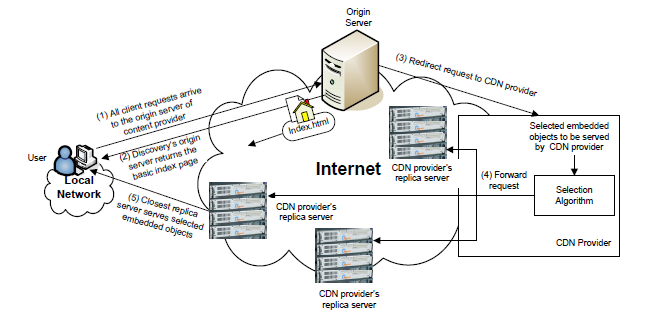
**CARP:** The Cache Array Routing Protocol (CARP) is a distributed caching protocol based on a known list of loosely coupled proxy servers and a hash function for dividing URL space among those proxies. An HTTP client implementing CARP can route requests to any member of the Proxy Array. The proxy array membership table is defined as a plain ASCII text file retrieved from an Array Configuration URL .

**ICP:** The Internet Cache Protocol (ICP) is a lightweight message format used for inter-cache communication. Caches exchange ICP queries and replies to gather information to use in selecting the most appropriate location in order to retrieve an object. Other than functioning as an object location protocol , ICP messages can also be used for cache selection. ICP is a widely deployed protocol.

**HTCP:** The Hypertext Caching Protocol (HTCP) is a protocol for discovering HTTP caches, cached data, managing sets of HTTP caches and monitoring cache activity. HTCP is compatible with HTTP 1.0, which permits headers to be included in a request and/or a response. This is in contrast with ICP, which was designed for HTTP 0.9 .

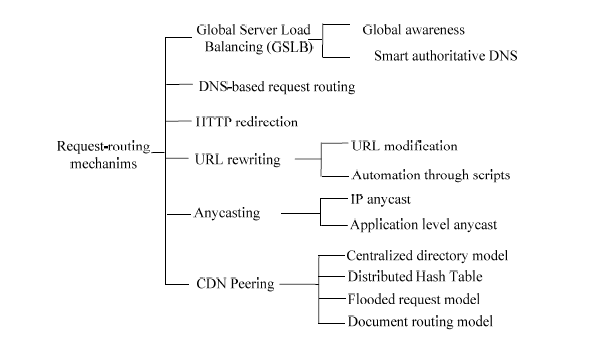
**Cache Digest:** Cache Digest is an exchange protocol and data format. Cache digests provide a solution to the problems of response time and congestion associated with other inter-cache communication protocols such as ICP and HTCP. They support peering between cache servers without a request- response exchange taking place. Instead, other servers who peer with it fetch a summary of the content of the server (i.e. the Digest) .

**Request and Response Routing and their Mechanisms:** A request-routing system is responsible for routing client requests to an appropriate surrogate server for the delivery of content. It consists of a collection of network elements to support request-routing for a single CDN. It directs client requests to the replica server ‘closest’ to the client. However, the closest server may not be the best surrogate server for servicing the client request [151]. Hence, a request-routing system uses a set of metrics such as network proximity, client perceived latency, distance, and replica server load in an attempt to direct users to the closest surrogate that can best serve the request .



Above Figure provides a high-level view of the request-routing in a CDN environment. The interaction flows are: (1) the client requests content from the content provider by specifying its URL in the Web browser. Client’s request is directed to its origin server; (2) when origin server receives a request, it makes a decision to provide only the basic content (e.g. index page of the Web site) that can be served from its origin server; (3) to serve the high bandwidth demanding and frequently asked content (e.g. embedded objects – fresh content, navigation bar, banner ads etc.), content provider’s origin server redirects client’s request to the CDN provider; (4) using the proprietary selection algorithm, the CDN provider selects the replica server which is ‘closest’ to the client, in order to serve the requested embedded objects; (5) selected replica server gets the embedded objects from the origin server, serves the client requests and caches it for subsequent request servicing.

**Mechanisms:** Request-routing mechanisms inform the client about the selection of replica server, generated by the request-routing algorithms. Request-routing mechanisms can be classified according to several criteria. We classify them according to the variety of request processing. As shown in Figure below , they can be classified as: Global Server Load Balancing (GSLB), DNS-based request-routing , HTTP redirection , URL rewriting , anycasting , and CDN peering .



**Best Practices or Recommendations for CDN:**

* **Make sure you have a solid business reason to outsource content delivery:** There are many reasons to consider using CDN . Since, CDN’s distribute your content from many locations around the world, customers can receive content faster and servers are under less load and costs will come down. Research for the needs of using CDN’s.
* **Identify where your customers are; investigate latency to their location:** If servers are in California and customers in New York, look at how long pages will load in California and compare them to New York .There are services that will help us to analyse the response, latency and transmission and other components that make up the end-to-end time a page a page needs to be entered to the customer’s browser. You may find that the components outside of network transmission are better targets to start with.
* **Make sure to identify which areas of the site can be put onto CDN:** There are two main aspects that can be cached on a CDN ; images and other assets and even HTML. As non-HTML content normally comprises 70-90% of the weight of most pages, there is some for improvement in caching across a CDN. However, on the HTML side, many e-commerce sites have dynamic data which is difficult to cache on CDN.
* **Carefully think about the benefit of using CDN:** How much does and improved site will help to increase in sales? Does it enables to shop faster? Is the current site so slow, such that it drives sales away? Is the traffic on CDN is cheaper than current ISP? Determining the answers to such questions, like these will help to understand what benefit you will achieve.
* **If using a CDN, try to address the slowest page components first:** Remember the slowest loading asset will dictate how fast pages load. If you could have load 99% of your site which could load in 1 second, and thee other 1% loads in 20 seconds, the total page load time will never be better than 20 seconds. Optimisation of the slowest components may be done by CDN.
* **Monitor Performance and continually make adjustments:** Once we have implemented a CDN, we will need to continually monitor and improve our site performance.
* **Don’t use CDN to make up for an un-optimised site:** If the site has extremely heavy pages, that are unnecessary then first address these pages, the revisit the CDN question.
* **Fully optimised pages may not be used on CDN:** If the site is producing dynamic pages, from the application server, the server may generate these pages individually for each request and CDN would not be able to cache them for performance gains.