A Study Report
on
Content Distribution Network’s Technology
& its
Financial Market.

Muhammad Irfan Younas Mughal
Mustafa Khan

This thesis is presented as part of Degree of
Master of Science in Electrical Engineering

Blekinge Institute of Technology
May 2009
ABSTRACT

With the advancement of the Internet age, the need for more and more data distribution to different users on different types of networks in short time and at a nominal cost has also increased significantly. To achieve these objectives several technologies have been used with different sorts of implementations but only few survive in today’s very competitive financial market. The objective of our thesis is to study the technology of the Content Distribution Network, which has up till now proven to be a very good and effective way to meet the always increasing demands of the rapidly developing Internet age.

In this study thesis, we will not only discuss the taxonomies of the Content Distribution Network or CDN but also its different types and implementations. CDNs provide the approach of eliminating congestion in the network and to solve the performance issues in a congested network. Their basic goal is to evade the congested links in the network. If the communication flow between client and server is not able to traverse a portion of the network (which is congested) then CDN is likely to have good performance. If the path is not congested, it might be possible for a client to get its services from a secondary server (surrogate server) with better performance.

The aim of our thesis project is to study and understand the technology of the CDN, the issues related to its types and implementations. We will also briefly study its financial market and the major companies providing CDN services.
ACKNOWLEDGEMENT

First and foremost we are really thankful to Almighty Allah without whose blessings none of this has been possible.
We are very thankful to our supervisor Alexandru Popescu for all of his help and guidance for our thesis.
We would also like to thank Professor Adrian Popescu (during the Telesystems class lectures) who provided us the interest and courage to study this technology in more detail.
Finally, we would like to show our gratitude and love for our parents who have always motivated and appreciated our efforts.
Contents

ABSTRACT ......................................................................................................................... 3

ACKNOWLEDGEMENT .................................................................................................... 5

LIST OF FIGURES ......................................................................................................... 10

LIST OF TABLES ......................................................................................................... 11

CHAPTER 1 .................................................................................................................... 12

1.1 Introduction .............................................................................................................. 12

1.2 Background ............................................................................................................. 12

1.3 Architecture of a CDN ............................................................................................ 13

1.4 CDN Functionality ................................................................................................. 14

1.5 CDN Types .............................................................................................................. 17

1.5.1 Internet CDN ..................................................................................................... 17

1.5.1.1 Overlay CDN ............................................................................................... 17

1.5.1.2 Peering CDN .............................................................................................. 18

1.5.1.3 Hosting CDN ............................................................................................. 18

1.5.2 Subscriber CDN .................................................................................................. 18

1.5.3 Enterprise CDN .................................................................................................. 19

1.6 For whom is CDN? ............................................................................................... 19

1.6.1 Enterprises ......................................................................................................... 19

1.6.2 Service Providers ............................................................................................. 19

CHAPTER 2 .................................................................................................................... 20

2.1 CDN Site Design .................................................................................................... 20

2.1.2 Server or Servers? ........................................................................................... 20

2.2 Architecture .......................................................................................................... 21

2.3 Load Balancing ..................................................................................................... 22

2.3.1 Front-End Load Balancer ............................................................................... 22

2.3.2 Broadcast and Filter ....................................................................................... 23

2.3.3 Smart Directory Server .................................................................................. 24

2.3.4 Smart Selection by Client .............................................................................. 24

2.3.4 Client Redirection Protocol ........................................................................... 25

2.3.5 Concurrent Parallel Access by Client ............................................................. 25
CHAPTER 3 .......................................................................................................26

3.1 Client Redirection......................................................................................................... ........................ 26

3.2 GENERIC WIDE AREA ROUTING MECHANISMS .............................................................. 26
  3.2.1 Extensions to Single Site Schemes ........................................................................ 27
  3.2.2 Triangular Routing .................................................................................................. 29
  3.2.3 Special Directory Servers .................................................................................... 30
  3.2.4 Client Redirection Protocol .................................................................................. 30
  3.2.5 Anycast .................................................................................................................. 31
  3.2.6 Wide Area Routing ................................................................................................... 31

What’s best?................................................................................................................................................ 33

CHAPTER 4 .......................................................................................................34

4.1 Selecting the Right Site ................................................................................................................ 34

4.2 GENERAL TECHNIQUES FOR SITE SELECTION ............................................................... 34
  4.2.1 Active Techniques with Per-Request Monitoring .................................................. 34
  4.2.2 Passive Techniques Using a Routing Table ............................................................ 35

CHAPTER 5 .......................................................................................................38

Data Management in CDN......................................................................................................... ................ 38

5.1 DATA SHARING SCHEMES.............................................................................................. 38
  5.1.1 Transparent Remote Access .................................................................................... 39
  5.1.2 Replicated Copies .................................................................................................... 40
  5.1.3 Caching ..................................................................................................................... 40

5.2 DATA CONSISTENCY SCHEMES...................................................................................... 42
  5.2.1 Periodic Synchronization .......................................................................................... 43
  5.2.2 Update Notification .................................................................................................. 44
  5.2.3 Consistent Updates .................................................................................................... 45

5.3 CACHING FOR DIFFERENT TYPES OF DATA.................................................................. 46
  5.3.1 Caching Static Content ............................................................................................ 47
  5.3.2 Caching Dynamic Content ....................................................................................... 47
  5.3.3 Caching Processed Data ........................................................................................... 47
  5.3.4 Caching Multimedia Streams ................................................................................... 48
  5.3.5 Caching Database Entries ........................................................................................ 48
  5.3.6 Caching Application Code ........................................................................................ 49

CHAPTER 6 .......................................................................................................50

Interconnecting the CDN Sites ........................................................................................................... 50

6.1 USING A SECONDARY NETWORK .................................................................................. 50
  6.1.1 Wired Secondary Networks .................................................................................... 51
  6.1.2 Wireless Secondary Networks ................................................................................. 51

6.2 OPTIMIZATIONS ON THE COMMON NETWORK ............................................................... 52
6.2.1 Private Transport Protocols ................................................................. 52
  6.2.1.1 TCP Performance ........................................................................ 52
  6.2.1.2 Sharing a TCP Connection ......................................................... 53
  6.2.1.3 Multiple Concurrent TCP Connections ..................................... 53
  6.2.1.4 Modified Congestion Control Schemes ..................................... 54

6.3 MULTICASTING IN CDN SITES ......................................................... 55
  6.3.1 IP Multicast for Overlay Interconnection ...................................... 55
  6.3.2 Application Layer Overlays .......................................................... 55

6.4 CDN MANAGEMENT ISSUES .......................................................... 56
  6.4.3 Security Issues in CDNs ................................................................. 56

CHAPTER 7 .................................................................................................. 58
Applications of Content Distribution Networks ........................................ 58
7.1 APPLICATION SUITABILITY FOR A CDN ....................................... 58
7.2 STATIC WEB PAGES ........................................................................... 59
7.3 STREAMING MULTIMEDIA DELIVERY ............................................ 60
7.4 ADVERTISEMENT GENERATION ...................................................... 61
7.5 CONTENT ADAPTATION ................................................................. 62
7.6 MAIL SERVER ..................................................................................... 63
7.7 SHOPPING SITES .............................................................................. 64

CHAPTER 8 .................................................................................................. 65
CDN Pricing .............................................................................................. 65
  Pricing the CDN ..................................................................................... 65
  Aggregate Usage ............................................................................... 65
  Percentile Based ............................................................................... 66
CDN Financial Market ............................................................................. 66
Major CDN Providers .............................................................................. Error! Bookmark not defined.
  Akamai Technologies ......................................................................... Error! Bookmark not defined.
    History ............................................................................................. Error! Bookmark not defined.
    Technology ...................................................................................... Error! Bookmark not defined.
    The Financial Market ..................................................................... Error! Bookmark not defined.

LimeLight Network ............................................................................... Error! Bookmark not defined.
Conclusion ............................................................................................ 67

CHAPTER 9 .................................................................................................. 69
Related & Future Work................................................................................................................................. 69

9.1 CONTENT DISTRIBUTION STANDARDS........................................................................................................ 69

9.2 PEER-TO-PEER CDNs.................................................................................................................................. 69

REFERENCES ........................................................................................................................................................... 71
List of Figures

Figure 1 Congestion Example
Figure 2 Basic Architecture of a CDN
Figure 3 Communication Flow in a Content Delivery Network
Figure 4 Using a Load Balancer
Figure 5 Global CDN
Figure 6 Communication scheme with front end and proxy site
Figure 7 Triangular routing mechanism
Figure 8 Connectivity through virtual site
Figure 9 Static webpage access method
Figure 10 Mail server setup
List of Tables

Table 1 Sample Routing Information Matrix

Table 2 Sample Routing Table
Chapter 1

1.1 Introduction

Data distribution over different computer networks was always of great importance. As the Internet era started and WWW (World Wide Web) became so popular among the end users. The user needs and demands for more and more data availability especially in a short time increased dramatically. Data streaming and quick downloading can be good examples of such needs. As the time passed the user needs also increased and the era of streaming media, Real Time audio/video and Triple/Quadruple play begun. But fulfilling such needs should not only be quick but they should also be reliable & cost effective. Many technologies & techniques have been developed so as to achieve this goal in a more effective & economical way. But every technology has its pros and cons.

Today, the most effective way of data distribution for such needs is by using different techniques of Content Distribution Network. A content distribution network can be defined as a group or groups of computers networked together through the Internet and they also work together to provide the data or content to the end user in a more effective way.

CDNs provide the approach of eliminating congestion in the network and to solve the performance issues in a congested network. Their basic goal is to evade the congested links in the network. If the communication flow between client and server is not able to traverse a portion of the network (which is congested) then CDN is likely to have good performance. If the path is not congested, it might be possible for a client to get its services from a secondary server (surrogate server) with better performance.

1.2 Background

The basic concept behind a CDN is shown in Figure 1 [1]. It shows the common representation of a client trying to access a server across a backbone network. The client to server path is congested, as indicated by the broken line connecting the client and the server in the figure. Suppose, however, that there is a secondary server (surrogate server) within the network, with which the client has an uncongested path, and the client can communicate with the surrogate server instead of the original server. In this way, the client can obtain better performance by connecting to surrogate server instead of connecting to the original server through the congested network path.
If we further improve this approach and direct more of our clients to communicate with the surrogate server, and design a system in which the surrogate server can handle more client’s request then we can avoid the congested path of the network and the user perceived response time would also improve. For any surrogate server, there would be a limit to handle client’s requests. But we can extend this solution to most of the clients by having more than one surrogate within the network. Such a network layout is illustrated in Figure 2. It shows three groups of clients, each of which is in a different city. All of these groups are accessing the same original server content but through their own surrogate servers in their own city. The original server may still be used to serve any remaining clients in the network.

1.3 Architecture of a CDN

The CDN consists of many surrogate servers located at different locations, so that it is highly likely that a client has good connectivity to at least one of the surrogate servers. Each client is directed to one of such servers, and obtains good performance even if the communication link between the client and the original server is congested. Figure 2 represents the basic architecture of a typical content distribution network (CDN) [2]. Surrogate servers which are at the same location are combined to form a surrogate site. A surrogate site may contain a single surrogate server or many surrogate servers that cooperate with each other to further enhance the performance of the content delivery network and to avoid any congested paths in the network. Moreover, the processing capacity of the surrogate servers is added to that of the original server, which results in the ability to handle many more clients than the original server might be capable of handling by itself. Thus, a CDN can considerably improve the scalability of any networked application. Another benefit of CDNs is an improvement in application
reliability. Even when the original server is down, one of the surrogate servers may be able to provide content to the client, thus improving the accessibility of the application.

![Basic Architecture of a CDN](image-url)

**Figure 2; Basic Architecture of a CDN**

One drawback of a CDN is its manageability. A Content Distribution Network consists of many surrogate sites that are geographically spread. The management and administration of such sites is much more complex and costly than managing a set of servers at a single location. For this reason, CDNs should be designed with a strong importance on the manageability of the surrogate servers and sites.

### 1.4 CDN Functionality

The fundamental concept of CDN is to distribute contents to the cache servers which are located close to the clients. This goal is achieved by placing multiple points of presences (PoPs) with clusters, which are called surrogate servers and they maintain copies of identical contents. Such an arrangement provides better balance between cost for content providers and QoS for clients. The CDN is made up of multiple such points of presences and they are deployed at multiple locations and in mostly in different backbones. The most current content is provided to the client by the transparent cooperation of PoPs with each other as shown in the figure 2.

When the content is moved closer to the user, by doing so the capacity sum of strategically place servers (surrogate servers or Point of Presence) can become higher than the network backbone capacity. Content delivery to the optimal surrogate servers is done by using IP routers and to make this more efficient overlay routing (which we will study in detail in later chapter) is also used. In CDN two communication flows and two types of data (Encoded-multimedia data and Metadata) are used. [3]
A CDN infrastructure is made up of:

1. A set of **surrogate servers** which are responsible for delivering copies of content to the users while combining different activities.

2. The **request-routing** consists of mechanisms to redirect content requests from a client to an appropriate surrogate server.

3. The **content distribution** consists of mechanisms to move contents from the origin server to the surrogate servers.

4. The **accounting** is responsible for tracking and collection of data on request routing distribution, and delivery functions within the Content Distribution Network. The accounting is also responsible for creating logs and reports of distribution and delivery activities.

5. The **origin server** which contains the original content, interacts with the Content Distribution Network in two ways (as illustrated in figure 3):
   a. It pushes new content to the surrogate or replica servers. Moreover, the replica servers also requests content updates from the origin server through the distribution infrastructure.
   b. It requests logs and other accounting data from the Content Distribution Network or the CDN itself provides the content to the origin server through the accounting infrastructure.

The clients interact with the Content Distribution Network through the request routing infrastructure and surrogates.
Figure 3 illustrates one of the possible ways of interaction between the clients, the access routers, the surrogate servers and the origin server.

The user agent sends:
1. a content request to the routing infrastructure, that redirects
2. the client request to a surrogate server, to which the client subsequently asks
3. the desired content.

The CDN works on a two communication flow:
   a. One between the origin server and surrogate server
   b. The other between the surrogate server and the client

The server-client flow is replaced by this approach in the CDN. The issues related to QoS, Content Multicasting and Multipath Routing makes the CDN very complex. For the CDN architecture, it is very important to choose the best placement for surrogate servers as it is critical for the content outsourcing. Such an arrangement could also help in reducing the number of surrogate servers to cover a specific CDN. Some algorithms are also proposed for the surrogate server placements such as Greedy, HotSpot and Tree Based replica.

Another important question is “which content to outsource?” This issue has a direct impact on the customer needs as it is always desirable to outsource only that content which is requested by the customer but its not easy task to achieve. The customer’s need could change with time and also the amount of content being requested by the customers.
An efficient management policy could solve this issue by considering the customer’s needs and grouping the content based on correlation figures or access frequency and replicate objects in units of content clusters. Moreover, depending upon the type of CDN, an adequate policy can be chosen for content outsourcing. Such policies are:
   a. Push Based
   b. Uncooperative Pull based
   c. Cooperative Pull based

Mostly uncooperative pulling is used in the CDNs being used today.

CDN pricing is also very important. The factors affecting the pricing are bandwidth costs, traffic variations, number of surrogate servers and other related parameters. It’s a common observation that the cost decrease as the investment in a specific technology increases. But in the case of CDN, its more complex as with the increase in the customers needs, the bandwidth cost also plays an important role. Higher the customer needs, higher the bandwidth requirement and ultimately higher CDN pricing.

1.5 CDN Types

The basic idea and service provided by a Content Distribution Networks are fundamentally the same, despite of type; a range of different Content Distribution Network types are present in the industry, regarding the way they are realized. The basic types of Content Distribution Networks realizations are Internet, Subscriber, and Enterprise.

1.5.1 Internet CDN

Internet CDN is the general name for the most common realization of CDN models. Its working is same as described earlier, using a distribution of single or multiple surrogate servers at various points close to clients. The service itself is sold to the Web sites to ensure them, that their web site and content will be readily accessible by the clients and would perform well. There are also subtypes of Internet CDN, although serving the same purposes but they do differ slightly in their realization and approach.
   a. Overlay CDN
   b. Peering CDN
   c. Hosting CDN

All of these subtypes are in brief described here.

a. Overlay CDN

Nowadays, the term “overlay” network is most commonly used to describe many of today’s Content Distribution Networks. A very good example of such an overlay network is Akamai. Akamai does not have layer 3 networks or data centers of its own. As a replacement for this, it has created a virtual network of servers spread across many other
provider’s layer 3 network Point of Presences and data centers. By overlaying this network of servers on top of the backbone networks of other providers, Akamai was able to create a new place for Content Distribution Networks services without building new networks or data centers and saving a great deal of expenses. With the help of this approach can maintain an extensive network of servers and caches around the world. Some disadvantages are also present with this approach. First, as with any solution made up through partnerships or outsourcing with other companies, overlay CDN providers must be careful of the quality of the facilities they use. Since they do not control the networks, POPs, or data centers that accommodate their equipment, they must be careful in assessing the quality and effectiveness. But if they are not able to maintain their requirements, they’ll need to shift elsewhere, which can be difficult and expensive.

b. Peering CDN
Peering CDNs also provide the same service but, as an alternative of placing content at POPs or data centers, equipment is positioned at the peering points between the service provider’s network and its peers. This type of approach works well for service providers that have excess backbone speed, capacity and substantial peering connections with other networks. Since backbone capacity is not an issue, locating content at peering points is a more economical way to distribute content, instead of placing equipment in every POP. A clear disadvantage is that content is only placed on that provider’s network (network which is being used) and, therefore, isn’t as close to the user in all cases, as it would be in an extensive overlay network (which locates content on many different networks). This type of approach could also be used by hosting service providers that does not employ a substantial backbone. As a result, they want to offload content requests as soon as they enter the network at the peering points.

c. Hosting CDN
A CDN can also be implemented by using the existing multiple data centers to house content for the CDN service. Even though this won’t have the reach of an overlay CDN, it will be much less costly to build and maintain by using the existing asset in data center facilities. In most cases, hosting service providers will employ a hybrid approach of both peering and hosting.

1.5.2 Subscriber CDN
Even though technically similar to Internet CDNs, subscriber CDNs is targeted at a different portion of market. An Internet access service provider can set up content delivery infrastructure at POPs across the network. Afterward, with that equipment in place, it can provide an improved service to access customers. Imagine being offered a service for five or ten dollars a month that would enhance user’s web surfing and content retrieval activities significantly. Instead of simply providing it as a free service of standard Internet access (as this is implemented as a cost-savings mechanism by service providers), service providers can sell this capability to their customers and can earn more revenue. The main targeted customers for a subscriber CDN are consumers and even some businesses.
1.5.3 Enterprise CDN

For this type of CDN, the clearly targeted customers are enterprises with distributed campuses in many different places. The enterprise CDN locates caching infrastructure in each major enterprise place and positions content for commonly used items such as files, corporate audio/video announcements, and other commonly used contents. While using an enterprise CDN, companies can employ more robust media, provide faster access to common files, and reduce WAN transport costs. Some companies might offer a managed enterprise CDN that’s basically the same type as enterprise CDN, but implemented and managed by the provider company. By doing so the corporation is free from the burden of learning the new technology and managing it properly and puts that burden on the service provider company.

1.6 For whom is CDN?

CDN is not so far seen as very important for technology infrastructures. However this is rapidly altering. In near future, the technology market will look back and question how they ever worked without features of Content Networking solutions. Understanding, for whom is Content Networking is the primary phase in the process of making these solutions.

1.6.1 Enterprises

No doubt, Enterprises can have many applications for CDNs. They can benefit from services purchased from a company that uses Content Networking technology. Moreover, others can implement these technologies themselves. To achieve the necessary performance, scalability, operational effectiveness, and availability for an Enterprise is through these advanced Content Networking functions. Enterprise CDNs is of at most use for Enterprises where more and more content is required, and as multimedia demands are at most.

1.6.2 Service Providers

Since many service providers are making a lot of effort for running their business after the 2001 economic downturn, they must continue to offer more valuable services to their customers to move toward profitability. Services like data center collocation and Internet access are commodity services with low (or even negative) margins. CDN enable service providers to offer valuable new services & capabilities which can benefit them more then the services they are offering today. Service providers like Data Center and Application Service Providers, Managed Security Service Providers and Internet Service Providers can benefit a lot from CDN capabilities.
Chapter 2

2.1 CDN Site Design

WWW (World Wide Web) has developed from a speculative medium to a healthy design that handles time sensitive business traffic. Industry pundits indicate that the number of e-commerce users over the Web increased to 142 million in 1998 and is forecasted to surpass 500 million and this increase in use is expected to drive commerce on the Internet to more that $1 trillion\(^1\). On the way to hold up this development, CDN sites must be able to make and keep up millions of connections every moment. To support such a design higher bandwidth, powerful and more servers, speedy access, and links with fault tolerance are required.

In today’s Internet era, so many sites are competing by offering quick response time, better & continuous availability of their services. These qualities (better response and availability of services) have become mandatory requirements for the users so as to have the best online experience. The most important issue requirement in all mission-critical web-computing designs is the dynamic scalability of server capacity to fulfill the client demand, while making sure of continuous service availability. To achieve these requirements, we have to address several challenges brought along with them.

2.1.2 Server or Servers?

In this regard, the most important question is that for a given site, what approach should be used i.e., whether we should use a single server or multiple servers. To answer this question we can start by looking at a single server site.

Let’s say that a single server for a given website is not able to provide services properly because of the growing demands of users which could be the result of high load or availability. An online user’s utmost desire is the quick response time and that’s what keeps him/her loyal to the service provider. Moreover, a single server is not able to provide any kind of redundancy or high availability. If the server is compromised or heavily loaded or shuts down for maintenance (because of too many requests), whether scheduled or unscheduled, the whole site is down. Such an event can adversely affect the business of any site and specially the business of an e-commerce site, which could lead to loss of valuable customers. If we further look into this problem we will notice that for any server site scalability is a very important factor because it ensures that growing demands are met. But in our case (single server)if any upgrades whether software or hardware require the server and in turn the site is also brought down for that time of upgradation. To solve this issue multiple servers can be used so that if any kind of upgradation or maintenance is required then the site would still be running as the other servers would be serving the customers during the time of upgradation and maintenance.
The Server Farm Approach (using multiple servers) provides solutions for many of the challenges presented for a single server site; but naturally it has issues of its own. For example the issue of traffic direction to the appropriate server, traffic distribution among the servers and resolving overloaded servers.

2.2 Architecture

The architecture of a scalable CDN site can be defined as creating an application that consists of multiple applications, which are in turn running on different servers but for an end user our this application should look like a single application running on a single machine. A client request is handled by the application running on one of the independent machines. To further study this issue, we will consider the various techniques that can be used, which can in turn make sure that the client request is handled by one of the servers within the network.

We can discuss these techniques by dividing them into two main categories

a. the one which involves client participation
b. the second in which client does not participates

We will only look at the first approach as it is of utmost importance with our topic. When client plays an active role, then the client machine has to select an appropriate machine among the available choices. If the control of both communicating application is possible then the client participation is a good approach. In order to have some control over the client side communication we can select proper intranet locations for the servers. Although, mostly the control of server side communication is possible in the network communications.

But, in general running the server side of an application on multiple machines in parallel is difficult to do for all applications. Nevertheless, a large class of server applications can be executed in parallel fairly without difficulty. But a stateless server application can be run on parallel machines without a lot of work because it is capable of fulfilling each request from the client without requiring any information concerning how such requests were processed previously. Such behaviour can easily be seen in a read-only application, which is simply responsible for providing read-only access to some data, performing transformations of data into a format understandable to the client. A web server containing static web pages is also a good example of such applications. If the data being requested is copied consistently across several servers, such applications can be run at the same time on many parallel machines. Another way to do this is to have all the machines access a common shared store for the data. By following such an approach and having a shared data store, applications that need state information, e.g., the result of a previous request can also be run in parallel as long as the state is maintained in a shared store. In further discussion we will suppose that the applications running at the server site are ones that authorize easy parallel execution on multiple machines.
2.3 Load Balancing

So, the problem is, how do we can easily cope up with the parallel processing of applications on multiple servers. The solution to this is to deploy server load balancers that provide the infrastructure to scale application-processing power, maximize server efficiency, and ensure high-application availability. Server load balancers are special purpose devices, based on a switching platform that consolidates multiple web infrastructure functions with server load balancing and multilayer switching. Examples of these functions include redirecting traffic to caches, load balancing traffic across multiple firewalls, packet filtering, and bandwidth management. [4]

2.3.1 Front-End Load Balancer

Front-end load balancer is the most common technique to allow use of multiple machines while providing the illusion of a single system to the client. The arrangement of such a technique is shown in figure 4.

![Figure 4; Using a Load Balancer](image)

Client requests are sent to the site using the external network. The entire site has a single machine address and this address is advertised for the entire site, clients communicate
with the server through this advertised machine address. Inside the server LAN, the different machines may have the same or different machine addresses, depending upon the transport protocols used and the needs of the network. Usually, the advertised address is the address the front-end load balancer. A load balancer can be easily set up between an existing client and an existing server transparently, and the client or server machines need not to be of the load balancer’s existence.

When a client sends a request, the load balancer processes the request and determines the appropriate server from the several servers at the site and the request is forwarded to that server. Similarly when a client wishes to establish connection to the site, the load balancer makes a choice among the servers and directs the client to the selected one. By doing so, the load balancer is able to maintain a uniform distribution of traffic load among the servers. When a client has been directed to a specific server, the load balancer creates some state information and saves them. This state information can be used for selecting the same server if the load balancer receives the subsequent requests or packets the same client. When the server is communication with the client then the resulting packets may flow through the front-end load balancer, or they may be able to take a different path. If the server chooses to communicate through the load balancer then would usually allow for a more efficient load balancing.

The distribution decision of the load balancer software may be based on a number of different criteria. In case of the server, decision is based upon the traffic load among the different servers and the information contained in a packet may also be used to distribute a request. The type of the software used in the load balancer can play an important role in selecting the amount of information use for decision making. For instance some load balancer soft wares only look at transport and network headers while others some look at application level information to make a more accurate decision. Moreover, some load balancers have a full-fledged application proxy for making their decision.

The ease of setting up a front-end load balancer and providing a transparent communication between the client and server makes it a very smart solution for load balancing in a CDN site design. On the other hand, the front-end load balancer can be a single point of failure because the load balancer has to process all the incoming client requests of the site, so very much care should be taken so as to avoid the load balancer from becoming a performance bottleneck.

2.3.2 Broadcast and Filter

To direct client requests to one of the many server machines, another way is to broadcast and filter method. In this method, the client is broadcasted to all of the servers within the site, and only one server process the request through the use of a collaborative protocol. For instance, a client network addresses list could be maintained at each server and the server will only accept requests from the clients present in the list and the requests from other clients are filtered out. Each client can be easily mapped to one of the servers by disjoining the set of client network addresses used by any two servers and also making sure that each client network address belongs to one of the servers. If the broadcast and filter approach is used then it eliminates the need for a special front-end processor and as
a result there is no single point of failure. But its biggest disadvantage is the largely unbalanced load distribution among the different servers. This disadvantage can be overcome by using the load balancer to distribute equitable traffic load to different servers but it is hard to implement filter rules that can be effective in a balancing load on the servers using a broadcast and filter approach.

2.3.3 Smart Directory Server

In most of the networking protocols, the concept of an address as well as a machine name is used and the clients can be directed to different server machines during the translation from machine name to machine address. If a client wants to communicate with the server then it perform a directory search to translate the machine name to a machine address. In order to direct clients to the different server machines different clients can be provided with different machine addresses. The advantage of using a directory server is that it eliminates the requirement of a separate system to direct the client requests to the different servers. The servers can therefore be positioned in different locations, and the directory scheme can be used not only for a CDN but also for a single site. But, sometimes the translation stage is skipped by the client applications because the machine address provided in the past by a directory server can be cached. This is also possible if the client might have provided the machine address instead of the machine name of the server. Different networking protocols call up the translation at different phases of the communication flow.

2.3.4 Smart Selection by Client

If the client is able to select the server by itself then the client can be provided with the set of possible servers that it can use for its communication. The client is then responsible for selecting one of the many servers that it finds appropriate for its communication needs. The choice of selecting the server can be made at the network translation stage, when the directory server is used to map the machine name to its network address. This selection of server can also be done when the actual communication is established. If the selection is made at the translation stage then the directory server is only responsible to send the client with the network addresses of all the servers available at network site. After getting these addresses the client can choose the server randomly or through a round-robin approach. But if the selection is made when the actual communication is made then the client sends its request to all of the servers at the site. Every server responds upon receiving the client request, and in turn the client chooses one of the responding servers for its further communication. For instance, the client will start to communicate with the first responding server and the connection establishment process is aborted with the rest of the other servers. This kind of abortion may either happen automatically in most communication establishment protocols (if the client does not
respond), or if the client sends an explicit message to terminate the communication process with the server.

2.3.4 Client Redirection Protocol

If the client is responsible for making the selection of the server then the servers rely on the client for making the right choice for the server. But if the client machines are compromised or are not properly secure, a misbehaving client can easily exploit the scheme, which in turn make the loads unbalanced on the different server machines. This issue can be resolved by using a redirection protocol on the server site to take control of the decision. In this way, the client requests to the site for establishing communication are handled by a machine at the site, which selects one of the many servers to whom the client is ought to be redirected. The client is then told to contact the selected server for further communication.

2.3.5 Concurrent Parallel Access by Client

A client can also maintain communication with more than one server in parallel if it has sufficient resources. By following this approach it is supposed that the client makes many requests to the server during the different stages of communications. The client then receives information about the different servers that are present at a site; it sends a request to all of these servers. Upon the completion of requests, the client again sends the next request to the same server. The less loaded servers can respond quickly to the client requests and can get a larger share of the requests from the client, and heavily loaded servers or slowly responding servers get fewer requests. This scheme can balance the load among the multiple servers, if all of the clients use this approach of parallel access. Considering the network needs and the performance issues one or more of the above techniques can be combined to come up with a solution that can be used to design a highly scalable CDN site.
Chapter 3

3.1 Client Redirection

Many geographic sites are combined to setup a CDN, and the combined sites may be
designed by using the architecture described in the previous chapter. Chapter 2 also
examined several ways for directing clients to one of the many servers that are part of a
site. But now the problem in hand is of directing clients to one of the CDN sites, which
are geographically at different locations. Previously discussed techniques described the
ways for directing clients to one of the servers within a CDN site but technique like
specialized directory servers, could also be useful for distribution in different CDN sites.
But other techniques will cause significant degradation in performance or reliability
issues. Some techniques are only meant for client redirection among the different CDN
sites such as anycast or wide area route manipulation (which we will discuss later in this
chapter). Moreover, we will also look at other various schemes and will discuss their
relative merits and demerits.

3.2 GENERIC WIDE AREA ROUTING MECHANISMS

To understand the routing mechanism in CDN sites or in wide area we can refer to Figure
5, which shows a CDN with multiple sites spread throughout a wide area network. Every
CDN site is illustrated as a small cloud to highlight the fact that they may consist of
more than one server at that site. The clients try to access the services offered by an
application hosted at the CDN. They can originate from anywhere within the CDN
network. Such examples are shown through two different groups of clients in the figure.
All of these clients need to be directed to the sites within the CDN network which can
provide them with best performance and availability for each of them. So each client
should be directed to the most appropriate CDN site. For instance, lets assume that clients
originating from the darker cloud area in Figure 5 should be directed to site A, and the
clients originating from the lighter cloud should be sent to site B.

But in this regard, there are two problems to face, first is to determine the most
appropriate site for the user and second is to find out that what mechanism can make sure
that the user picks the right site for the communications.

Every server in the network has a unique name and is identified uniquely by that name.
The clients look up a server directory to map the name of the server to its network
address. This network address is used to route packets sent by a machine to its
appropriate destination. Before the actual communications the clients would have a
handshake with the server and then begin the transfer of packets reliably. The following
subsections in the chapter would describe the schemes that can be used for client routing
in CDN network or wide area networks.
3.2.1 Extensions to Single Site Schemes

In considering our first approach to wide area network we can simply extend the solutions that we discussed for a single site. But in this regard, some of these techniques would work well and others may not work quite as well in the wide area network. We considered the following schemes for single-site

a. Front-end load balancer
b. Broadcast and filter
c. Smart selection by client
d. Smart directory server
e. A client redirection protocol and
f. Concurrent access by the clients.

Some of the schemes have considerable drawbacks if we consider them for wide area networks. But others can be considered for wide area environment as well, and the last three would require alteration of the software at the client side. [5]

The front-end load balancer unless used as a reverse surrogate or otherwise, is one of the better approaches for a single site distribution. In order to understand that why this same approach does not work well in the wide area, let us understand the network shown in Figure 6. Only one front-end load balancer would be present in the network which is assumed to be working as a surrogate for all of the sites within the network.
If the network is congested, all clients may not have good connectivity to the front-end load balancer. The purpose of the CDN is to have these clients (which have bad connectivity with the front-end load balancer) be served by a surrogate site that has good connectivity with these clients. The clients need to go through the front-end load balancer first so as to reach the surrogate site and that path is congested part of the network. It is quite possible that a well-connected surrogate is present in the same part of the network where the (unserved) client or clients are located. In this situation there are high chances that the surrogate site’s connectivity to the load balancer is also congested. The entire communication path which consists of four legs (as shown in Figure 6) and all of them are communicating through a congested part of the network. So, the better alternative is to go without all processing at the surrogate and do everything at the load balancer site. Such an arrangement would result in only two legs of the communication going through the congested path and would result in better performance of the network.

But if a surrogate site is placed close to the load balancer, or if a private high-speed network exists between the surrogate sites and the load balancer. Then, simply extending the single-site solutions would improve performance.

The four-leg congestion problem can be solved if only limited exchange of information is made with the front-end load balancer, i.e. the client only learns about the best surrogate site.

Figure 6; Communication scheme with front end and proxy site
The client redirection protocol uses this approach. Alternatively, the surrogate could send data directly to the client rather going through the load balancer. Such scheme would lead towards the triangular route solution (which would be discussed later).

The method of broadcast and filter does not work well in wide area network. Nowadays, the wide area links are more expensive and have less capacity than the local area networks. So LANs can easily handle broadcast packets but the wide area networks are composed of links with lower capacity and repeating packets on them causes needless waste of valuable bandwidth.

A client’s smart selection from many available servers at a single site is relatively easy. In most cases, a simple scheme such as selecting the servers randomly or in a round robin fashion could put a reasonably uniform load on the servers. But when we are considering wide area network, the client also needs to take care of the distance between each of the sites and itself while deciding that which site is most appropriate for communications. As wide area network consists of a large number of sites, so communicating this information to the client could be very difficult. The other single-site schemes that could work well in wide area network are discussed later in this chapter.

3.2.2 Triangular Routing

The front-end load balancer approach could work in wide area network if we limit its role and it does not sit along both the forward and reverse paths of the network communication. An advertisement is made and one of the many sites is advertise as the primary site for the server. The clients are redirected to the other sites at the start of the connection. On the other hand, the comeback does not flow through the primary site. The responses are sent back straight to the client by the selected site.

Figure 7 shows how the scheme works. The inner cloud is the congested part of the network and is performing poorly. The load balancer at the primary site has selected the site that is near to the client side and it sends the data received from the client to the surrogate site. The surrogate site sends back the packets directly to the client. As a result, the packets only traverse the congested region of the network in one round trip and the network performance is better than the original case.

Considering this arrangement, question of using the triangular routing may arise although content distribution is the primary solution being deployed in the network. For the long run, it is better to have a larger capacity at the server site than to set up a content distribution solution. But in the short run, triangular routing could work well for getting some extra processing capacity.
3.2.3 Special Directory Servers

A special directory server approach that can map a client to one of the many different sites can be implemented both in the wide area network and in the single site. In this approach, a special directory server translates the server name to a network address whenever a client wishes to communicate with the server. In case of using this approach in the single site, the directory server is only concerned about the loads on the different servers so that it can decide which server is most appropriate for the client. But in case of the wide area network, the directory server is also concerned about the location of the client because it needs to select the site (among several sites) that has good connectivity with the client.

If the location of the client is known to the directory server than it could probably choose the right site with more accuracy. But some network protocol does not provide the location of the client to the directory server which in turn degrades the performance of this approach.

3.2.4 Client Redirection Protocol

If a client is redirected to an appropriate surrogate site using a protocol (in order to communicate with the server) then this could be an advantage in the wide area network. Following this scheme, the client would at first contact the server to query about the identity of the site that it should select to communicate with. The surrogate server has
information which can determine the appropriate surrogate site for each client. When the surrogate server knows the identity of the client then it can direct the client to the right server. Such a redirection protocol can be designed to work at any layer higher than the transport layer.

3.2.5 Anycast

Another approach for selecting one out of several servers is to support this notion explicitly as part of the network routing protocols. In the traditional anycast method, a set of addresses are renowned by the network as being a special type of group communication. This set of addresses is an anycast address. Multiple servers could have or share an anycast address. It is the responsibility of network routing protocols that any packets sent by a client to the anycast address should be received by only one of the servers that shares the anycast address.

The anycast protocol can be implemented in many different ways. A simple way to work with the anycast protocol is to broadcast a message to all the servers in the domain. Every server responds with its network address and the network stack at the client side selects the first responding server. The client saves the identity of the server whose response is accepted on the first packet to be sent on the anycast address, and uses it for future communication.

Another way to implement anycast protocol would be for each router to save a link to the nearest server in the anycast group. When servers advertise their membership in the anycast groups, the routers update their routing tables so that packets can be sent to the nearest server. By doing so, the routing tables may send packets to a different host as new members leave and join the membership group. If the client wishes to continue the communication with the same server for the complete duration of communication then communication with only one server is required, and the client only have to remember the identity of the server which is chosen on the first packet.

The basic drawback of anycast approach is that it requires support from the underlying network. If the network is not owned by the company who wishes to deploy a content distribution solution then it would be difficult to put into practice this approach. If the company owns the network, it still have to make sure that the switches and routers in the network support anycast. If the routers and switches are not equipped with anycast feature then upgrading all the devices to support the anycast feature could be a hard task.

3.2.6 Wide Area Routing

The other scheme that can give effects similar to anycast routing without requiring support for anycast feature is to make use of the routing infrastructure within the network. The routing protocols tend to select the shortest path between two entities in the network. In connection-oriented networks, the routing is invoked when a connection is being made. But in connectionless networks, a background routing process on each network device makes a routing table which is used in forwarding process.
Most of the routing protocols first determine the shortest path between two entities within a single administrative domain, and secondly they determine the shortest route between entities which are in two different administrative domains. In a single domain, each entity maintains a full topology of the devices within the administrative domain or it maintains a simplified routing table containing the best routes to each of the entities within the administrative domain. In multiple administrative domains, the routing information is aggregated so as to provide a more scalable solution. The aggregation process being used depends on the type of protocol being used within the network.

Generally, in routing protocols it is assumed that a destination may be connected to multiple entities in the network. This is to hide the fact that the network is composed of multiple sites. The operator of the CDN advertises its connectivity to the network as if it appears to be a virtual site, which is connected to several points in the network. Figure 7 illustrates the working of this concept. A CDN consisting of four sites is depicted in the left part of the figure. The right part of the figure shows that in what way the connectivity of the site should be published by the operator of the CDN so as to set the routing tables to select the nearest site.

![Diagram](image)

**Figure 8; Connectivity through virtual site**
Even though we can use this routing approach with existing network infrastructure but some points must be kept in mind. A lot of routing schemes aggregate addresses into bigger units so as to decrease the volume of the routing tables being exchanged. The addresses which are used for the site should not be aggregated by the network during the computation of network addresses. It means that the provider of the CDN should work in close cooperation the provider of basic networking services. But if a very large network consisting of multiple domains is involved, then the administrative issues (involved in this scheme) can be more complex than the technological issues.

**What’s best?**

The different approaches described above (most of them) can be implemented to direct clients to one of the many sites that make up a CDN. The operating environment is the governing authority in deciding that which of these approaches are best for the network. We can combine some of these techniques to come up with a better working solution. For instance, we can combine client redirection protocol and special directory server. The directory server can take care of the bulk selection and if it happens to end up with the wrong site, the client redirection protocol can take care of the correct site selection. Such practical examples of combined approaches can be seen in the different solutions are used to locate web servers in a wide area network.
Chapter 4

4.1 Selecting the Right Site

In Content Distribution Networks, many sites are grouped together and when a client sends a request to the CDN, the request should be directed to the site which is the most appropriate for the client. Now we will look at the different methods that can be implemented to find out the best matched and most appropriate server site.

The selection of best site depends on the purpose for which the Content Distribution Network is intended. Content Distribution Network can be deployed so as to reduce the response time to a client. For this scenario, the most appropriate site would be the one in the vicinity of the client and it also depends upon the amount of traffic load at each of the considered sites. For other scenarios, the Content Distribution Network is employed for scalability improvement of the server site and the client response (from a server) is of less importance. In these scenarios, the most appropriate site would be the one that is experiencing the minimum traffic load at a given time.

The methods that can be useful for the right site selection for a client are active probing, use of a static routing map, and the use of a dynamic routing map. Later on, we will discuss these techniques in detail.

4.2 GENERAL TECHNIQUES FOR SITE SELECTION

The techniques for selecting a site can be categories as:

a. active monitoring when a request is received
b. passive schemes using routing tables

c.

In the technique of active scheme, extra data packets are generated within the network, if a request is received. On the other hand, a passive technique does not generate extra data packets on receiving a request. But it creates a table which maps different clients and requests to the most appropriate site and this table is used direct clients appropriately. The routing table requires some extra packet flow with in the network for its operation and maintenance, but these extra packets are not attached to a specific request.

4.2.1 Active Techniques with Per-Request Monitoring

The extra packets generated in as active technique are used to measure the network characteristics and server load at the several sites within the network. This packet generation is invoked, when a client request arrives at a decision maker. For instance, if a directory-server-based scheme is used for load balancing of wide area, the special directory server would begin the active mechanism when it receives a query request from a new client. The directory server plays the role of decision maker in this scheme. But in
other schemes, a surrogate server or a front-end load balancer can play the role of a
decision maker.

The nature of extra packets being generated by this scheme is related to purpose of the
CDN. If the purpose of the CDN is to improve the server scalability, i.e., to handle as
much requests as possible, ten the request should be directed to the least loaded server.
To determine the least loaded server, an active technique would invoke the decision
maker so as to send a probe message to each of the sites. This probe message will be
answered by a performance monitoring server present at each of the sites and it will send
an estimate of the current load at the site. The server with minimum traffic load will be
selected by the decision maker.

If the purpose of a CDN is to decrease the user response time, then site in the vicinity of
the user should be selected. So for doing this, the performance monitoring server is
responsible for returning a metric to the decision maker, which represents the distance of
the client from its site. The decision maker would then make a selection depending upon
this metric and site with minimum distance is selected and then the client is informed
about this selection.

When a probe message is received by the performance monitor at the site, it can then
calculate the site distance to the client or its traffic load explicitly. Another way of doing
this is to learn the site distances to the client from past experience, and it could also
monitor the site traffic load periodically. By doing this, the performance monitor server
can send the most recent calculations to the decision maker.

4.2.2 Passive Techniques Using a Routing Table

In implementing a passive technique using a routing table, the decision maker creates and
maintains a routing table that maps the client requests to the best site for the client. This
routing table has two fields. One is for identifying a group of clients, and the other is for
identifying one of the Content Distribution Networks sites. When a decision maker
receives a client request, it looks up in the routing table to find the best site for the client,
and sends the location of this site to the client.

But some times, the forwarding decision is not only based on the location of the client,
but it also depends upon the resource a client may want to access. For such times, another
routing table may also be maintained that is mapping a specific resource present at sites
in a CDN to the client request for that resource. In such a routing table, three fields would
be present, first will identify the client, the second will identify the resource, and the third
one will identify the nearest site. To server every client request, the location of the client
and the request for a specific resource are identified and they are used to determine the
appropriate site from the routing table.

The routing table is created from the routing information matrix that provides a distance
metric between each client and the site or the resource at the site. The distance metric
could also contain the round trip latency of the client to the site, the traffic load on a
specific site, or could be the combination of both metrics. To further understand this, consider a wide area CDN with three sites situated in New York, Stockholm, and Tokyo. Clients for this Content Distribution Network may come from Europe, the Americas, Asia, or Africa. Let us first take the simpler approach in which routing is based on the geographical location of the client. The routing information matrix for this CDN is depicted in Table 1, where the distance metric is the average round trip latency between a client and the closest site [6]. By considering this routing information matrix, the decision maker can make up a routing table, which is shown in Table 2, and will route users in the Americas to New York, users in Europe and Africa to Stockholm, and the users in Asia to Tokyo.

<table>
<thead>
<tr>
<th>Client Location</th>
<th>Surrogate server</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>New York</td>
<td>120</td>
</tr>
<tr>
<td>The Americas</td>
<td>New York</td>
<td>30</td>
</tr>
<tr>
<td>Asia</td>
<td>New York</td>
<td>160</td>
</tr>
<tr>
<td>Africa</td>
<td>New York</td>
<td>160</td>
</tr>
<tr>
<td>Europe</td>
<td>Stockholm</td>
<td>30</td>
</tr>
<tr>
<td>The Americas</td>
<td>Stockholm</td>
<td>120</td>
</tr>
<tr>
<td>Asia</td>
<td>Stockholm</td>
<td>160</td>
</tr>
<tr>
<td>Africa</td>
<td>Stockholm</td>
<td>100</td>
</tr>
<tr>
<td>Europe</td>
<td>Tokyo</td>
<td>160</td>
</tr>
<tr>
<td>The Americas</td>
<td>Tokyo</td>
<td>200</td>
</tr>
<tr>
<td>Asia</td>
<td>Tokyo</td>
<td>60</td>
</tr>
<tr>
<td>Africa</td>
<td>Tokyo</td>
<td>120</td>
</tr>
</tbody>
</table>

Table 1; Sample Routing Information Matrix

It is sometimes possible that some resources are not available at every site. In this case the distance metric can be different for each type of resources in the routing information matrix. For instance, in our CDN example, a streaming video server may only be present in New York and Stockholm and not in Tokyo. To solve this issue, the routing information matrix would add another column indicating the resource present at the sites, and in turn the distance metric would include the availability or the performance of these resources available at different sites.

<table>
<thead>
<tr>
<th>Client Location</th>
<th>Surrogate Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Americas</td>
<td>New York</td>
</tr>
<tr>
<td>Europe</td>
<td>Stockholm</td>
</tr>
<tr>
<td>Asia</td>
<td>Stockholm</td>
</tr>
<tr>
<td>Africa</td>
<td>Tokyo</td>
</tr>
</tbody>
</table>

Table 2; Sample Routing Table

The routing table derived from the static information matrix may be static for the entire system. But, this won’t be as effective as if the routing information matrix is created in a dynamic manner so as to indicate the most recent network and server states.
All the sites which are monitoring the performance of clients accessing them may maintain dynamic routing matrix information. The sites can monitor the client’s traffic passively or it can also do the same by active means, i.e., by probing the client, invoking the client to generate a probe packet of some type. Several performance monitoring sites can then send their collective performance statistics to the decision maker. The decision maker can then merge the different statistics reports together so as to maintain a routing information matrix. After doing this, the decision maker can derive a routing table from the routing information matrix for the network.

As an alternative to a centralized approach and relying only on a single decision maker to incorporate the performance statistics collected by the sites into the routing information matrix, a distributed approach could be helpful in making the routing information matrix and routing table. In this approach of distributed routing, different sites would exchange their performance statistics with other sites; in turn each site would maintain a composite routing table from the information received from other sites. The routing table or the routing information matrix for the decision maker can be obtained from its local site. This approach will enable any of the CDN sites to act as a decision maker.

The approach of passive routing table can work well if good performance statistics of a client’s access to a site are obtained. If a client or another client in the same area has accessed the Content Distribution Network before, then it is likely to show up in the routing table thus formed. Although, when a client visits the decision maker for the first time then no information about the client location may be available. Active and the passive approaches can be used together for such a case. But if the client information is available in the routing table then this information is used to route the client. If information is unavailable then an active approach is used in its place, or a default site is selected.

Many different approaches can be used to create the routing table in a distributed manner. A very simple approach is that all of the sites participate in a single multicast group. Each site broadcasts its performance characteristics to the multicast group, and each of the sites can then merge characteristics to build a distributed routing table. By doing so, each of the sites will perform the functions of the central routing entity on its own. The distribution of the routing information could also be made by creating a set of overlay connections among the sites. The routing table could then be shared through the overlay links. This approach could be of great advantage if multicast feature is not supported on the wide area network sites.
Chapter 5

Data Management in CDN

As we have discussed earlier that a Content Distribution Network is composed of one origin site and one or several surrogate sites. The origin site and the surrogate sites share the same data and code through different ways and schemes. The data present at the origin site can be remotely accessed by the surrogate sites, they can copy all or some of the data (as per need) from the origin site. This copied data can be cached all or some part of it at the surrogate sites. The techniques used to implement this idea will be discussed with their own merits and demerits later in this chapter (section 5.1).

The cached copies of data from the origin site are maintained at different surrogate sites and they can become inconsistent. Therefore, the consistency of copied data (with the original at origin site) is of much importance so as to deliver the most current and updated content to the clients. In order to achieve this goal many techniques have been developed for data consistency in a CDN and we will look at these techniques in detail in later section. (5.2)

After examining the different techniques of data consistency, we will describe the different types of data, which constitute a typical application and the various schemes through which a surrogate server can maintain the relevant types of data (in section 5.3).

5.1 DATA SHARING SCHEMES

In a broad sense, data sharing schemes which we can implement at different surrogates can be divided into three categories, namely;

a. transparent remote access,
b. replicated copies,
c. caching.

In first scheme, a single copy of data at the origin site is accessed by the surrogate sites remotely. In second scheme, an independent copy of data is maintained at different sites and each of these sites act as peers and different techniques are implemented manage the consistency among them. In last scheme, the origin site data copy is the official copy, and the surrogate sites maintain a cached copy of a subset of the data.

The schemes, which we will discuss in detail, are

1. Transparent Remote Access
2. Replicated Copies
3. Caching [7]
5.1.1 Transparent Remote Access

In this scheme, only one copy of the data maintained at the origin site and this data is accessed by the surrogate sites. This access is made transparently through a middleware layer, such as a special file system, or a RPC (remote procedure call) facility. We can also employ the remote mounting facility available in most common file systems, in which a file system is physically resident at the origin site but it is mounted as a network file system by machines at the surrogate site. The machines at the surrogate site can also access the data by using remote procedure invocation and it can hide the location of a file system or a database from the application.

For the transparent remote access scheme it is not required to know applications that are executing in a Content Distribution Network. The technologies which are used for remote access are well established in the market (e.g., CORBA, Java-RMI, etc.). The consistency is not a big issue in this scheme as only a single copy of the data is maintained.

This approach is best for the network, in which latency between the users of the data and the copy of the data is low. When several machines are connected together in a fast local area network than any one of them can export its file system, which is mounted over the network. Thus, access latency would be relatively small. In a similar manner, a database can be hosted at a machine and the others may access it using Java RMI. This scheme of having a single copy of data with remote users works fine as long as the latency between the users and data is small.

On the other hand, if the network latency between the users and the copy of the data is relatively high, this approach may not work very well. Since, the basic reason for having a CDN is to have performance gains by decreasing the client’s need to go to the origin server because the network path between the client and origin server is likely to be congested. The surrogate to origin site link is also likely to be congested, as both are likely to be in the same area of the network. So, our scheme of remote access is not likely to decrease the client-perceived response time of the system because the delay between the surrogate and the origin sites is large and further delays (because of the remote access to the data) can add up considerably.

This approach can still perform well in some cases of high latency between the different sites. If the total processing time includes only a small amount of data access latency in an application then such a scheme will work. For instance, an application which is responsible for displaying a complex multidimensional graphics of a database. So, the data access time from the database is quite low as compared to the time required for displaying the graphics. In this case, it is quite acceptable to maintain a single data site and retrieve data when needed.

In an Internet-based Content Distribution Network, the CDN sites access may be provided by a private network with a faster access between the sites than similar access over the open Internet. Further more, in considering the scheme of maintaining a single copy of data as an adequate solution also depends upon the characteristics of the private network.
5.1.2 Replicated Copies

In this scheme, each CDN site holds a complete copy of the entire data and has equal authority over the data. When any application needs to access the local copy of the data, it is completely free to perform its reads or writes on the data. If several individual computers are present in a CDN site, then internal access of the site is done by employing a transparent access scheme. Replicated copies scheme of the data provides adequate performance as far as the latency in performing an operation on the data is concerned.

Any user request can be divided into two phases: first is reading and formatting data and second is modifying or changing the data (maintained at a site). The formatting refers to changing the format in which data is presented. Formatting data means to make the data or database entries compatible or suitable for presenting it the user side (e.g., in a browser). Replication offers substantially gained in performance and scalability if most of the requests were for reading and formatting data. But for an update request, the distribution of the data has two more issues: one is to ensure synchronization between conflicting updates from two different caching sites and the other is data consistency between the various surrogates caching the data. Data consistency means to make sure that a surrogate already holding a cached piece of data gets the new copy of the same data (when an update is made to the data). So, such updates should be propagated consistently in order to make the data consistent at different sites. This can be done by employing a variety of schemes for update notifications of the data to all the sites in a CDN.

The replicated copies of complete data require a lot of storage space across the entire CDN site. If the cost of storage space is not an issue (for any given CDN) then replication can result in substantial performance gains. It’s a very good scheme for relatively static data (i.e., data is not likely to change very much) because the hard part is to manage data consistency which arises due to data modifications by the users. The replicated copies scheme improves performance as the ratio of read requests to write requests increases. For virtually static data such as an old songs or old photographs archive, replicated copies provide best solution.

5.1.3 Caching

This scheme is a combination of the replicated copies scheme and the transparent remote access scheme. In this scheme, an official copy of data is maintained at the origin site, and the surrogate sites depend upon the origin site data copy to obtain the required data (same as in transparent remote access scheme). Further more the data is maintained locally at the surrogate site (similar to replicated copies scheme). But, in caching scheme, the surrogate data need not be a full replication of all the origin site data and only the needed data is replicated at the surrogate.

When surrogate site receives user requests, it attempts to satisfy the request from the locally cached data. If the requested data is not present at the surrogate then the data is
retrieved from the origin site. This copy of data is cached for satisfying future requests. This method is for the pull mode cache, in which the cache pulls data from the origin site as per request. The other method is the push mode cache, in which the origin site initiates data distribution to the cache and data is pushed to the cache. In this mode, if a request is not served at the cache then it is directed to the origin site. Practical caches can employ any of the two modes or a combination of both, in which data can be pushed and pulled (from the origin site) as per request.

The caching scheme is based on the principle of locality. This principle originates from virtual memory systems in computers and it describes that computer programs do not access all the pages of their address space uniformly because some pages are accessed more frequently than others. So, caching the most accessed pages in the main memory improves the program performance, which is running in virtual memory environments.

This principle is appropriate for general caching in wide area networks so any application can be considered as programs accessing data items. Applications may not use data uniformly and it can make repeated access to the same set of data at any time. So we can say that many applications display locality of reference. For the geographically distributed applications, locality can be of three types.

1. Temporal locality means that at any given time, some portions of data will be repeatedly accessed by the application.
2. Spatial locality means that an application accessing a portion of data is more likely to access other portions of related data.
3. Geographical locality means that some data types are more likely to be accessed from one geographical location than another.

So, caching can result in considerable performance benefits, whenever locality is high.

When a cache has less storage space, it discards the older entries to make room for new entries. Space management in cache can be vital for performance if the required space at the cache is small for the working set of data needed by users. The working set of data is the amount of data that is accessed by the current set of active applications. When large objects are accessed by an application then working set could exceed the available cache storage. In other situations, the application’s required working set could be smaller than the available cache storage. In the latter situation, cache replacement policies won’t be needed.

The classic least recently used policy is mostly used for cache replacement schemes. In this scheme, the least recently accessed cache entry by any user is replaced. If all the cache objects have same size then this scheme performs quite well. But when the subjects are of different sizes, policies that replace larger objects are helpful for some workloads. Workloads which access small objects more repeatedly than large objects and working sets which are larger than the storage available at the cache may have better performance if large objects are replaced more frequently. By replacing large objects makes more room for many small objects and replaces objects that are less likely to be accessed for the upcoming requests.
The cache requests that modify the data can cause inconsistent data present at the origin site and the cached surrogate data. The caches must be consistent with each other, for doing so data consistency schemes are used. These schemes make sure that the data cached at different surrogate sites and the origin site is consistent with each other. Caches tend to improve systems in which the read requests are much more frequent than the write requests. So, many caching schemes may not handle update cache requests and forwards them to the origin server. But an update can invalidate cached data at other surrogate servers so consistency schemes must be used to maintain the data consistency at different surrogate sites.

5.2 DATA CONSISTENCY SCHEMES

Many schemes have been developed which can be used to maintain the data consistency among multiple distributed sites. The nature of replicated data is the most important factor in determining, which scheme to use. Some replicated data may require data consistency at all times, others may tolerate inconsistency for some time, and some may not require consistency at all. e.g., data of a bank account may require consistency of data at all times to make sure the correct representation of money available in the bank account. If data is used to represent an individual’s postal address then a scheme that can make sure that the address is made consistent within a couple of days would be sufficient. The data which does not require consistency could be a set of colors used for rendering. If the color scheme at the origin site is changed, even then the surrogate site can use the old scheme because it wont effect the rendering much. Moreover, the data for archives wont need consistency as it will not changed once it is published. The environment (in which data is used) plays an important role in determining the consistency requirements for the similar types of data. For instance a consistent color scheme could be very important for some environments, but could be of less importance or even unnecessary in other environments.

In the later sections, we will examine the different types of consistency schemes for managing consistency among replicated copies of data.

1. Periodic synchronization, which works well for data that can tolerate a limited time window of inconsistency.
2. Update notification, which lessens the time window of data inconsistency up to the same magnitude as of network latency.
3. Consistent updates is the scheme that ensures that replicated copies of data are kept consistent with each other at all times.
5.2.1 Periodic Synchronization

In periodic synchronization, multiple distributed copies of data in the network are synchronized with each other at periodic intervals. The synchronization could be done at regular periodic intervals, or it can be triggered by another event which is occurring at irregular intervals. For instance, synchronization of a cached data item may be done on every fourth request for that item, or even on every request, or the same data item is retrieved from the origin server after period of every five minutes.

Most commonly used type of periodic synchronization is writethru caching. In this approach, any update request of data is forwarded to the origin site because the official data copy present at the origin site is always current with all the updates. In turn, the other caches synchronize themselves with the official copy of data in periodic intervals. This synchronization with the official copy can be done either by replicating copied data or by deleting cached copies of the data at periodic intervals. If replication scheme is used all surrogate sites replicates the official copy of the data in periodic intervals. This replication could be initiated by each of the surrogate sites independently, or by the origin site. Moreover, expiration time is used by each surrogate site for its cached data. As the expiration time is met, the cached entry is expires and is removed from the cache. Upon receiving any requests for the expired entry, the surrogate site retrieves the official copy of the data and caches it once again. Periodic replication and expiration time serve the same purpose of making any cached entries consistent after a fixed time period.

Another approach is modified readthru caching. In this, requests for reading the cached data items are first validated with the official copy at origin server to check for a new copy availability (if any). Even though read-thru caching does not decrease the response time to a client when the data item being retrieved is comparatively small, it can save substantial network bandwidth (and the related decrease in response time) when the large data items are cached.

Another scheme, in which each surrogate site has its own replicated data copy and performs read operations or update operations to its own data copy. During regular intervals, the different data copies are merged together to provide a consistent data copy across all of the surrogate sites. The data copies at each site contains multiple data records and each data record has a global time stamp (which represents the time when the data record was updated). The global time stamp comprises of the local time of data copy update and the identity of the data copy where the record is located. If the replica clocks are roughly synchronized, the global time stamp then represents the relative order in which the record updates are made. When replication is performed, each data record is updated with the version that has the highest value of global time stamp among all the replicas available. By doing this, it is made sure that all the replicas of the data are synchronized with each other periodically.

Periodic synchronization is an ideal solution for data that can tolerate limited windows of inconsistency. For instance, in WWW, caching surrogates use expiration time to cache web pages for a limited time. Periodic synchronization is not appropriate for data, which requires consistency at all times.
5.2.2 Update Notification

In periodic synchronization, the data inconsistency window depends upon the synchronization period that is selected in a random fashion. The data can thus become inconsistent in between the synchronization period. But this problem can be solved and copies of data can be made consistent by exchanging a few messages with each other. This message exchange can reduce the inconsistency window to a small multiple of the network latency between any two sites within the Content Distribution Network.

Such message exchange approach is called update notification. In this approach, whenever there is an update in the data copy at any surrogate site, this data updating notice is sent to all other sites in the CDN. Moreover, this notification may also contain the location information of the most recent copy of the data and the other sites then synchronize their records with the updated data copy.

The easiest way of using update notification is to employ write-thru caching with a single official data copy. The official site is responsible for keeping track about the updates and processes them as they are received. Whenever an update is detected, the official site notifies all the sites that a specific data record has been changed. Upon receiving this notification, each site invalidates the local copies of the changed record (if any), and retrieves the new copy from the origin server.

There can be many variants of the basic update notification scheme. Such as, the notification message can carry the actual updated record rather the informing that a record has changed. If the data records have small sizes, then this approach could provide a more effective way for update notifications to the CDN sites. Another approach is that the CDN site would forward the update request instead of the official site. This would reduce the processing required at the origin site. The updates can be sent through UDP or through TCP, if the delivery mechanism requires acknowledgment for update notifications sent to the surrogate sites.

If any specific data record is cached at only few CDN sites, then global notification of the update won’t be necessary because it will only consume network bandwidth and processing cycles at each site. So for such cases, it’s beneficial for the site that generates the notification to maintain a list of sites that may have kept a specific record. In turn only the listed sites are informed about any updates to a specific record. This list is maintained by the official site along with write-thru caching because the official site sees all the requests for the data record and for caching. This approach decreases network bandwidth usage and the number of notifications received at each surrogate site, by slightly increasing complexity at the origin site.

As the number of data records is large, number of such lists could also be very large. The number of lists can be reduced by defining a limited set of categories and the list of sites for notifications is maintained on category basis. So each record would belong to one or more of the categories being defined for the CDN. When a data record is modified, the update notification along with data record identity is made to only those sites that are
related to the data record category (which is being modified). Every site checks if the modified record has been cached locally and if so, it is thus invalidated. This technique shows a middle path between generating notifications to all of the sites for each update and maintaining a separate list for each data record.

If the data records are depending on each other, e.g. a data record containing the worth of different properties may depend on the current value of the property taxes and market price of the property. So, any changes in the taxes may cause an invalidation of several properties worth records. This dependency can be captured by means of a dependency graph, in which each node consists of a data record and a directed edge connects a record to the other records that may depend on it. This graph can then be used to determine the data record dependencies. The official site can then make a decision to send the suitable notification for the dependent records as well and the process is repeated (if needed.).

The asynchronous notification approach works well for CDNs that can tolerate inconsistency windows for few tens of milliseconds e.g. stock quotes and current scores in a sports event It should be kept in mind that the update notification process is not designed to promise data consistency; it simply reduces the inconsistency window. If the data record is changed before the notification process is initiated, and similarly it is possible that some systems may not receive the update notification on time because of network losses. If guaranteed data consistency is required, the schemes discussed in the later section should be considered.

5.2.3 Consistent Updates

The data, which requires consistency at all times, schemes are employed to ensure synchronization and consistency of data within the distributed databases. For instance, data items such as balances in bank accounts, in which inconsistency can cause serious financial problems. The distributed databases schemes can make sure that serialization and consistency of updates is maintained across multiple sites. This enforcement of atomicity, serialization, and consistency is known as transactional semantics. A transaction is defined as a read operation or writes operation on one or more data record in a system.

Atomicity means that a transaction is either successful or failed at all. Serialization implies that two transactions initiated at several sites that may be proceeding at the same time should have the identical outcome as if one of them was executed after the other. The execution order of transactions is normally not important. In this regard, data consistency implies that data records in all replicated copies should contain the same value after the transaction completes.

In case of a single site, atomicity and serialization is achieved through mutually exclusive locks that have to be acquired before performing any data operation. The obtaining order of these locks determines the execution order of transactions. Locks could be defined on several levels (depending upon the granularity required), i.e. per data record, one per type of record, or over a complete database. As the granularity increases the number of locks
in the system also increases but it permits more concurrency, thus user response time is improved. The fundamental locking method is improved by operating system and programming language constructs such as monitors or the synchronized tag in Java, which hide the basic working of locks from most of the programmers. For a single site, consistency is not an issue.

For multiple sites with copies of the same data, atomicity, serialization, and consistency are achieved by considering the concept of locking in a distributed environment. The traditional move toward this goal is the use of the two-phase commit. As the name suggests it works in two phases, the preparation phase and the commit phase. During the preparation phase, the request originator sends out a message to all other sites. Upon receiving this message, all sites will obtain the suitable locks for performing the transaction, and send back a positive acknowledgment to the originator. If any of the sites is sends a negative acknowledgment, then the transaction is aborted. But if the originator receives a positive acknowledgment from all the sites then, the originator will start the commit phase. When the commit phase is initiated, the originator asks each site to execute the transaction. Each site sends back a positive acknowledgment to the originator upon completing the transaction. If all sites send positive acknowledgements to the originator, it will ask them to release the locks as a final step. If any one of the site sends a negative acknowledgement to the originator, it will ask all of the sites to roll back the transaction.

This method is best for obtaining consistent updates. But as the number of exchange messages for the updates grows with the number of sites involved, hence the time required to do this update will also grow. Although the method works quite well for distributed databases connected through a low latency local area network, but the performance could degrade quickly over wide area links. So, this method must be carefully examined for performance implications before implementing it in a wide-area CDN.

**5.3 CACHING FOR DIFFERENT TYPES OF DATA**

The caching paradigm is most commonly used to share data in existing CDN service providers. Let us see how different types of data are typically cached within a Content distribution network. We will examine different types of data and the suitable caching schemes for them.

First, we will examine static data or data that is not likely to change or changes very slowly. Then we will look at dynamic data, which changes frequently. Further more we will consider the type that is generated because of processing queries or transformation of existing data records. Then we will discuss steaming multimedia streams and structured data (the content of relational databases and LDAP directory servers). Finally, the program codes or application logic that must run at the different CDN sites are discussed.
5.3.1 Caching Static Content

Static data is easily cached at a surrogate site within a Content Distribution Network because of its simplicity. So, static data can be pushed to the origin site for caching, or pulled on request from the origin site. The cached item currency is maintained by attaching an expiration time with it. If a cached item is relatively large and tends to change then the cache can check with the origin site to look for the changes. Thus, retrieves the current version (if changes are made to the cached object) from the origin site and forwards it to the user. Changes in static data items are relatively slow, so a periodic synchronization scheme for data consistency works quite well. Static item caching can decrease the user-perceived latency if the item is likely to be requested repeatedly from the cache. So this caching scheme also works very well for applications with a high degree of temporal locality and is likely to improve the user perceived latency in retrieving an object. Similarly, it’s a good approach to prefetch related data items (which are likely to be accessed in the subsequent requests) for applications with a high degree of spatial locality.

5.3.2 Caching Dynamic Content

Data that can change frequently and quickly becomes outdated is called dynamic content. Caching dynamic data is helpful, when the data is likely to be accessed many times before it becomes obsolete, e.g. scores in a tennis match change quite quickly within a few seconds. If the match is watched globally then thousands of people are likely to request the score before it changes. So, caching the current set of scores can be an effective approach. But if only few reads are expected before the data changes again, then it is better to service all requests from the origin sites instead of caching. Another approach to handle dynamic data is to broadcast the new changed data to all of the caches as soon as it changes. If the network round-trip latency is less than hundred milliseconds, then caches would receive the data quickly, and can provide it to the users. Dependency graphs and category-based notifications can be used to further enhance the efficiency of sending the updates for dynamic data to others.

5.3.3 Caching Processed Data

For application processed data items, it could be advantageous to cache them, rather than the raw data itself. The processing function performed may be executing a query on a set of data items or reformatting a data item, such as translating a web page from Swedish to English. If the same processing function is most likely to be performed in the subsequent request, it may be useful to cache the results instead of raw data.

If the data processing cost is high, caching the processed data for repeated requests can lower this cost. For instance, if the English translation of an article published by a Swedish editor is in high demand, caching the English translation of the article can get rid
of the need to do the translation for subsequent requests. In the same way, if a search for certain books of stock market is frequently observed at on-line book stores, caching such search results can save considerable processing and improves system performance.

### 5.3.4 Caching Multimedia Streams

Multimedia content is made up of audio and visual data that can be used by a client in one of two ways. It can download the whole file and then play the downloaded file from the local disk. But, for larger files, downloading can take a long time. So, to reduce this time before one can play the multimedia file, streaming mode is used. In this mode, the client can start playing the clip as soon as it has downloaded a segment of the file for a fixed period of time (pre-defined in the system). During the playback, other segments are downloaded from the server to be played when the current segment finishes playing. For interoperability, the client and server use a common streaming protocol.

This multimedia caching is similar to caching static objects, except that multimedia objects are much larger in size. For the caching of streaming multimedia streams, the surrogate cache has to employ the utility of the streaming server. The surrogate can decide by considering the granularity of a segment of the multimedia file rather than the whole file. For highly demanded multimedia files, all the segments can be kept in the cache. For multimedia files that are not requested much, the cache can only keep the initial few segments, and download (from the network) the remaining ones upon receiving a client request. This can be beneficial in saving the storage requirements at the surrogate cache.

### 5.3.5 Caching Database Entries

A database is a storage area of several data items with an arrangement. The mainly common databases today, are relational databases. It organizes information into one or more tables, and each table contains a fixed number of columns. Each entry in the database is defined by a row of the table. Information is accessed using a query language that allows the selection of a subset of the entries in the database using a sophisticated set of expressions, including loops and if-conditional statements.

Even though distributed databases are also a choice, but they need to maintain a consistent update operation which is hard to maintain. So, to eliminate the consistent update problem, a database cache would normally process only queries that do not modify the database entries.

Caching the query results is one approach to build a database cache. This can be an effective method that can be built into the access protocol accessing a database. But, the query language complexity implies that a canonical representation of the queries may be difficult to achieve. On the other hand, for a limited subset of queries, query result caching can be done in a relatively simple manner. For applications with simple queries, query result caching for databases can be used effectively.

The other method is to cache a subset of the database records at the surrogate sites and run query operations against the local subset of data. Updates are sent to the origin site.
database. A subset of data can be defined which is cached at the database. Each query needs to be analyzed to confirm that the results can be obtained from the presently cached data at the local server. Updates to the origin database should be sent to the cached copies as soon as possible. The most important actor in database caching is to maintain the highly reliability and consistency without affecting performance.

5.3.6 Caching Application Code

Like other data elements, the program part or code can also be cached at the surrogate site to enable some of the processing functions to be performed there. The programs that are used to serve the user requests for an application have the characteristic that don’t change quickly. So, it means that much of the application processing at the origin site can also be offloaded and can be executed at the surrogate site of a CDN. By caching application programs, distributed program execution with a single point (where software is installed and configured) can be used. However, the executing programs may result in some errors and these errors needs to be sent back to the originating site so that they can be analyzed.
Chapter 6

Interconnecting the CDN Sites

We discussed earlier the Content Distribution Network (CDN) includes a number of alternatives to serve the common application or set of applications. Cooperation is needed in different CDN sites to communicate with each other. Interconnection of different programs can be used to CDN sites. The main communications sites in CDN increases with the surrogate sited is probably the source of CDN sites, but the representative agent communication is also possible in many cases. Can not meet the requirements of customer’s proxy site, and asked that the information can be found in the local cache, the source communications need to receive the point.

When the CDN networks which includes a number of surrogate sites, together with a common application service or a group of applications. Cooperation’s in this field in different locations, the CDN need to communicate with each other. In this chapter we expect to integrate different programs can be used to credit the website. Major communications site in Canada is likely to be with the surrogate sites and origin site of sources, but alternative communication is possible in many cases.

When surrogate sites are unable to meet the customer's requirements, since the necessary data is not available in the local cache, the source communications need to receive the point. Proxies site originating Web site to communicate the notification to happen given the site of origin, or are pushed into the source of the cache site. The quality of the link between the surrogate’s site and the site of origin has a significant impact on the performance of the CDN. This section examines the various programs to improve the link between the sites, so that plus.

Because the customer is informed of the source site and a number of the proxy site, it is necessary that there is a common network that connects customers to various locations in CDN. We referred to this common network sharing network. Common connections on different sites, as well as one, but the performance of this path can not be very good. Access to CDN following the performance of the agent close to customers in the path between the public network and the origin of the alternative site is likely to be filled, because the path to the client and the network. It is better chance of getting a job, if the contact points of surrogate sites and sites of origin can be better than the public network.

6.1 USING A SECONDARY NETWORK

The best links to sites so that make up a CDN, plus when one can run a specific network connections in different locations within the CDN. It is a private network infrastructure leading to better performance for two reasons: (1) If the budget permits, is likely to be linked to a site connected to the source is a surrogate origin site to connect faster than
customers of the sources of those sites; (2) There is no competition outside the private network traffic sources.

Such private networks can create a wired network or using wireless communications technology, such as those based on the satellite. Matters, the two networks processed the subsections below.

### 6.1.1 Wired Secondary Networks

Is the alternative a private server and the site is the ideal way (with the exception of the cost problem) to connect the surrogate sites to the origin sites within a CDN. The ability to choose the only means of access is faster than the ability of normal use between the agent and server on the public network. Delays in communication between agents and the site holding a private network, but may also be less than the common network.

Technology shared Internet service provider can also take advantage of the operator's web site (a CDN surrogate sites) are a means of independent users to visit on the Internet. If the site is connected to multiple Internet service provider network, connect it to all the ISP's customers are likely to be significantly improved. As a result, many sites to choose form a number of Internet service providers. This allows for faster access to the website also provides a fail-safe mechanism in question is an Internet service provider's experience of failure.

### 6.1.2 Wireless Secondary Networks

In other hand cable network is a wireless network in secondary between in different sites. The wireless network may be by satellite or radio link, according to the actual separation of the network in different places. The satellite network uses a geo-stationary satellite in the sky as a central router. Each site sends the information via satellite, which will include its aims. Radio contact to set up a direct sight line between the structures of the website and can be used only in a different Web site, where it is relatively close. The wireless network may be a viable option, of which the content of the site are located in remote areas with good links. These include mountain in Africa and the United States and many parts of Asia infrastructure, wired connection is relatively low.

Wireless networks in secondary schools usually do not have the same characteristics as the cable network. Satellite networks often have higher latency than wired networks due to the fact that each transmission via satellite. While this lack of a satellite-based wireless networks, the basic Regulation, the interests of secondary cable infrastructure that multicast communications. The satellite-based infrastructure, that the site sends the same data, all the receivers at the same time. The data relay satellites can receive and deal with all sites at the same time.

Most of the communication takes place a credit web site is multicast in nature. When the proxy server's cache is a surrogate-site data are likely to be resolved, the information required to multicast to all sites. When the data update a valid notice meeting all the
cache and when copy the contents regularly, the source site's content is sent to all the surrogate sites. To increase the use of any of these techniques to manage in CDN, wireless satellite network can be a good solution for an auxiliary network.

6.2 OPTIMIZATIONS ON THE COMMON NETWORK

The environment of network setup in some cases, the second private networks are not possible because of cost. Two network operators are very expensive, and any performance increase from private network operators can not be a good price. In such circumstances, it is necessary to draw up a plan to improve the communication between two locations to participate in a CDN. In this section we will see some of the technical, you can make effective communication.

Site Map more a result of the network shared in CDN, they have access to the same physical resources, the remaining units, using a common network. Internet, each CDN site is used in the credit standards of the TCP or UDP communication protocol. However, when a site that is necessary and CDN need to communicate with each other, they can be used by private agreement effective standard protocols.

Reviewing a number of private contract, can be used, we focus on the Internet, where communication takes place using the standard TCP / IP protocol suite. Application layer protocol, these agreements vary depending on the running applications. Optimizing can change to any agreement over the transport layer. Transport Layer is the first layer, are only explained the communication between endpoints, not the routers in the network.

6.2.1 Private Transport Protocols

Lots of traffic, makes the Internet and corporate intranet including TCP flow. TCP protocol is time-tested agreement has proved the stability and fairness of different network conditions. Connect to exchange a large amount of information; its performance is strongly influenced by the TCP congestion control of its program. Connect to exchange a small amount of information, the TCP connection performance, and the strong calls for the establishment of the connection.

6.2.1.1 TCP Performance

TCP connection to performance the impact of the context of the time, flow control program and its congestion control. Flow control refers to the aim is to ensure that the data sender does not spill in the receiver buffer. Congestion control refers to programs whose aim is to ensure that the node path between sender and recipient are overcrowded (i.e., more than the buffer can be used for packet forwarding scale).

Applications to change a small amount of data (e.g., data, less than one TCP segment), congestion control performance are not a bottleneck. A key factor in their performance on
the time required to establish a TCP connection. Connection set up time is required for at least one return, and data transfer can not be started until the connection established.

TCP is a reliable protocol, to sends the data unit called a TCP segments. The receiver will send a confirmation of each component of the sender, it will receive. If the sender does not receive a confirmation after the timeout period, it knows that a loss has occurred. If the certificate of receipt, the sender knows the package is over.

6.2.1.2 Sharing a TCP Connection

In any case the communication of different sites for a number of CDN a small amount of information, they are available on the performance of TCP re-links the various stock exchanges. Rather than set up a TCP connection, each of the exchange, they can set up one connection, so to pay the cost to establish a connection only once, all to communicate. This reduces the time required for each Exchange.

For example, consider a situation, the CDN, which the surrogate site in HTTP proxy server agent. Proxy server can use multiple clients. It can set up different scenarios for each client's requirements are not available in cache, disconnected from the source site when a customer breaks the surrogate origin site. However, if the file is downloaded, it is very small; the surrogate origin site may be considered in this regard. When a new customer set up a proxy server to connect, it can send a request form and a link to the original set up costs can be avoided.

Sharing TCP connections mean that these two site's content distribution network through a common application-layer multiplexing program can tunnel several requests for a single TCP connection. Communication between the parties shall be compatible with the common plan, to review the normal operation of the operation.

6.2.1.3 Multiple Concurrent TCP Connections

Communications to exchange large amounts of data, connectivity is not the bottleneck in base overhead costs. Connecting to the performance impact of interest rate restrictions, the status result, and tone of congestion control and management system uses TCP connections. If the amount of information on flight restrictions, the receiver buffer is the receiving end, then more data in flight can be sent using two or more parallel sessions of the two endpoints. Under normal circumstances, and each connector has a separate receive buffer pool, and a parallel meeting in these circumstances, can effectively double the amount of information, during the flight.

When the communication between the content distribution networking sites are the poor, the main reason congestion path between two locations, which operates a number of simultaneous TCP connections can be helpful in certain situations. If you have multiple TCP flow of a congested link to each perform the same congestion control scheme, and it is roughly the same round-trip time of each flow, eventually taking its fair share of link bandwidth congestion. If the communication between the units decides to have a number
of simultaneous TCP connections, as one, it is ultimately a larger part of the link congestion, which has a better efficacy. There are two interpretations must be taken into consideration when using multiple links in order to have equitable sharing of communication bandwidth content distribution network in place.

The first mention the crowded conditions of contact may not result in Content Distribution Network website. If the congestion caused by traffic between the CDN sites, mobile distribution between the dominant sites and to increase the flow of each share of reduced communication bandwidth for each link. For example, consider the situation, the only communications link congestion is due to TCP flow two CDN site. If there is only one TCP connection to use the site, open the congestion window until it is to use the entire bandwidth of the link. If there are two TCP connections to, and each may have half the bandwidth, and effective performance, the two will not increase. The results may be even lower than one connection, as indirect participation in the management of extra connection endpoint.

### 6.2.1.4 Modified Congestion Control Schemes

TCP is a very polite protocol, each sender to reduce interest rates so that the network congestion. However, participants in the content distribution network in the Web site can use their own congestion control protocol it can be more active TCP. If the effective throughput of modified TCP congestion protocol is better than what can be offered, and the communications content distribution network to improve the site. The most positive factor for things to do is to remove the congestion control. This means that it ignores the congestion window and sends the data packets have the highest rate permitted by the flow control receiver window. If it is not crowded road router, there is no damage, since normal TCP congestion control to accelerate the speed of the final.

The project has been positive, only the sender can be ignored. If the number of active senders becomes large, the entire network of drowning the lack of congestion controls. In addition, if the sender is a positive large number of network traffic, they can cause more congestion and losses from all senders comply with TCP. Non-use of congestion control is non-actors, such as reduced queue in front of the restaurant. If only a small number of people to do so, it will reduce the waiting time, the expense of others. However, if it started before the jump the queue, there is no doubt, what is worse, the waiting time for each person, if everyone followed the rules above.

Another option is to use rate-based congestion control mechanism. Rate-based congestion control system determine the speed of the sender, it can transfer data, and only send interest rates lower. Decided to send the correct tax rate is check your network connection set up, and the contribution limit may vary over time.

The revised design for congestion control scheme must be careful. The revised plan may not be so aggressive so as to affect the normal TCP connection, very much. If the plan is too radical, the machinery of normal TCP connection, it is their own experience of severe environmental degradation, new congestion control scheme used to select the machine or
application. If you want to show some of the normal application for their own work, and that would be a good idea, so that they are in a reasonable performance. Otherwise, many of the normal business of content distribution network between the sites may be affected adversely.

6.3 MULTICASTING IN CDN SITES

To many cases, the content of the sites includes the broadcast of information the site of origin of substitute sites. When the cache is the substitute of the site or amount of time replication occurs, it is a great amount of information on the site of origin status of surrogate sites. If the substitution of objects implementing the cooperative caching option, they can flow multicast query searches for a file that is not present in the cache. Thus, a significant part of the online content of the site consists of a message sent to the many participants on the site. In this section we will consider various options for multicasting, which can be used for CDN.

6.3.1 IP Multicast for Overlay Interconnection

IP standard provides a particular group of addresses, which represent multiple receiving machines. The packet is sent to all multicast group of machines. Machines can be members of one or more multicast groups. Accessions and the hosts in these groups are dynamic, using the Internet Group Management Protocol (IGMP).

In order to use IP Multicast and the content network sites is a private multicast address, which they use. All contents networks use multiple groups to send information to all participants. The network will ensure routing of the packets to a destination. IP Multicast works great information that can tolerate a limited loss.

Multicast protocol enables reliable single sender reliably transfer data over a single receiver. The fact that the sender communicates with more than one receiver is a multicast complex than the one (single-sender one recipient) protocols such as TCP.

Reliable multicast protocol enables a sender and reliable data transmission over a single receiver. In fact, the authors provide more than one receiver to allow more complex multicast (sending a single recipient) protocol such as TCP connections.

6.3.2 Application Layer Overlays

Problems in an IP multicast are that it is not always available. Multicast packet is forwarded and more complex activities Routers forward packets from unicast. Internet Group Management Protocol also created a series of networks. Performance problems due to many network operators turn off the multicast network. Network - if the operator environment, it's in line for the broadcast of a program is necessary for the establishment, but the content of the implementation of the network nodes themselves multicast.
Application-level multicast, each node to take participates in the single-link communication. Take part in the relay node information issued by the other members of the site. Mesh topology is from each of the participating nodes, which was created to connect a subset of other nodes. Send each message to one or more neighboring countries covered, and the message forward to other users.

6.4 CDN MANAGEMENT ISSUES

A related problem with the content distribution system is a machine the whole wide area. Each site's introduction of the content distribution networks and each machine is a content distribution network required to manage it. Management, including management of the installation, configuration and software updates to various web-based content distribution network, and the consolidation of the use of statistical and accounting information of the locations at one site. The second part of the content distribution network management is to ensure that the communications and web-based content distribution network to maintain security. This chapter looks at ways that can be used to solve these problems.

6.4.3 Security Issues in CDNs

Three different types of security issues must be taken into account in the design, add:

(1) To ensure that each site, increase, it is an integral part.

(2) To ensure the security of communications between the CAD website.

(3) To ensure safe content, the cache or copy of a Web agent.

In order to ensure a secure content distribution networks to ensure that the Web is no different from any secure Internet at the network. A secure web-based content distribution network is to provide one or more firewalls, to prevent various attacks and unauthorized access.

All encryption is at the cost of degraded performance. It is a common infrastructure is already very full; the encryption makes the content of the visit between the distribution networks, and even more expensive. Therefore, the content distribution networks are likely to be effective in the environment, its content is accessible to the public, or for other web-content distribution network to a minimum.

There are three ways to deal with sensitive content site of the CDN. The first is to ensure that no sensitive information is stored in a safe location can not be very strong. This means that all users’ requirements for safe handling of the page or asks the user to provide sensitive information (such as credit card number, and e-mail address) should
only be the country of origin of the web site, with sufficient safeguards to deal with sensitive information correctly.
Chapter 7

Applications of Content Distribution Networks

In various parts of the content distribution network and a variety of methods that can be used to create these components. They do not address the use of the Canadian dollar, and the types of applications to be held in Canada. In this chapter we look forward for some applications, typically running a central server, but can benefit from using the Content Distribution Network.

The use of CDNs is serving image and multimedia clips from surrogate sites. The architecture of CDN can be use in many different applications. For example use for increasing the user response time and increasing the scalability. Through the number of users that can be supported concurrently with a particular response time we can measure scalability.

In this chapter we discuss the advantage of the CDN and discuss the design of CDN network for better advantages and find out which application suitable for CDN in the predominance of the web at the current time.

7.1 APPLICATION SUITABILITY FOR A CDN

The most important achievement is a credit institution is to improve the user's response to improve the speed and scalability of the system. To implement these goals is by removing some of the functionality to replace the close to end-users. Progress and scalability is the user's reaction depends on how good the content of the surrogate mother are in the service.

The client request, the surrogate must be correctly service without needing to access the origin server. The requirement of requests that are working with surrogate is high; the surrogate will be able to provide better performance. Many type of caching schemes use for surrogate (stream caching, file caching, database caching, application caching, etc.) to increase the amount of requests that it can locally service. Surrogate to services in the local data type is currently visiting the request should be cached, and the likelihood of the required information on existing proxy cache would be very high.

Date of request to the requirements of living in surrogate to fulfill their high probability of a number of different assumptions. For example, the source of information throughout the site promoting the site's authors, agents must always be informed. If only a small part of the data source's Web site to promote (or cache) is the surrogate mother, there will be some cache miss. However, if the pattern of use revealed preference customers in a limited part of the data, then the cache can be quite effective. In other words, the customer usually receives information more frequently than others.
Each time, keep multiple copies of data, there is a risk that the loss of control of these tracks to synchronize to each other. Thus, the cached data has changed relatively slowly, and a number of inconsistencies to be acceptable. Slow down, if the data changes, the system can use the data for consistency of methods, to ensure that the information ultimately consistent. Speed / slow changes should be compared with the effectiveness of the planned introduction of cache coherence. Since most of the data plan would lead to some inconsistencies in the window, the limited window of inconsistency should not lead to the application.

In short, an application may be a good candidate for a CDN if

- It has a high ratio of reads compared to writes
- Client access patterns tend to access some set of objects more frequently
- Limited windows of inconsistent data are acceptable
- Data updates occur relatively slowly
- An application is not a good candidate for a CDN if
- It requires strong consistency of data
- It modifies shared data and requires concurrency control
- It has security or licensing needs that require centralization

The above list provides a rough guidelines and what kind of applications can help the CDN environment for credit. The application also can plan their way to make some or all of the benefits of implementing the CDN.

7.2 STATIC WEB PAGES

The first application, we review page is provided via the Internet or corporate intranet. Any object can be a Web browser, is related to the Web site. Is the object Web page, HTML, text and links to other objects. Contact an object can be divided into two categories. Visit a number of related objects through the browser page, List, while others are related to object access to only certain user's actions. The first two categories, known as embedded objects in the linked object, usually graphics and images that are used to create a stunning visual impression of all mobile users. Embedded objects and links to web pages are mostly point to objects, and Web pages, the same server, but they can also point the objects in any other server on the network.

The majority of users consider postponing the visit to the page, because the network is the time take to download the embedded object. Improve efficiency, these objects can be a surrogate site in a CDN, and not place of origin itself. This process is automatic if the URL object reference suitable substitute location. The only problem is the more suitable sites for different customers in different locations depending on the customer’s network.

There are two common approaches, with each customer directly to the server. The first option is that the web site and all the objects using a common name or IP address. Consider, for example, the embedded object contained in the website
www.originalsite.com the URL of the page / images/image1.gif or its equivalent, a website www.originalsite.com/images/image1.gif. This URL may change in the reference www.surrogatesite.com/images/image1.gif. When a customer tries to get converted into a URL, the DNS-based IP routing of a regulation can be directly used in an appropriate growth in the customer's site. Surrogate mother must be translated into binding surveying government and Mapping and the corresponding transfer image receiver requirements.

Figure 9; Static webpage access method

Another way is to the origin site, and maintains different versions of pages, each version of graphics to guide customers to download one of the surrogate sites. Map of place of origin to maintain routing table to determine which version of the page should be sent to each customer. This approach allows more routing the customer's representative. Site map origin to know the identity of the client request, and can choose the representative can not be more specific than the DNS approach, it is not necessary to coordinate closely with the network provider is not in practice, the IP routing. However, it requires more care site of origin.

7.3 STREAMING MULTIMEDIA DELIVERY

The multimedia content (audio, video and animation) that the website can be used in two ways: either by downloading the entire contents of the client, and after playing this game, local and / or used in a media streaming server demo. Streaming Server Client to start
browsing the content a lot faster, is the preferred method, the long-term real-time streaming, or watches the events online.

If the clients substitute the functionality of the surrogate sites, the multimedia streaming server, this service can provide a replacement site, but that customers can access the website source. Network performance can play a key role in the mass flow to the customer. Who will play all the multimedia playback of streaming Internet encounter stage, the network congestion has frozen the flow or quality severely deteriorated. As an alternative location closer to the customers, visit the cached copy of a flow can provide better performance.

Different types of digital rights architecture is developed for various industry associations and individual companies. However, the regulation at the level where it contains three main functions: (i) way that the author, content, restrictions on the use of designated, (ii) for restricted use of combustible contents, (iii) approach to the implementation of access restrictions. Any digital rights management architecture to implement these tasks in various types.

To play surrogate site, the surrogate master surrogacy website to support two digital rights management features. They are limited to using the multimedia content and control over these restrictions. Digital Rights Management system may restrict the use of hand-in-hand description of the multimedia content can provide the mapping is a central location. Implementation of digital rights is usually done as a player or browser software, content, is the client. Although some law enforcement agencies can also make a server, restricting access to authorized customers, it may not be able to prevent reproduction of the contents of the client.

**7.4 ADVERTISEMENT GENERATION**

What are often integral parts of banner ads page usually header or footer of the page. These sites are usually the ads are in the script appearing in your web sites or their country of origin, in many cases, a third-party advertising companies like Double Click’s [77]. In order to create the ads are effective, it must take into account the preferences for selected users, the Web site to place ads.

In the internet era, there are two types of system-generated advertisement. One type of ads generated Web content is based entirely on the user is visiting. Other types of image to keep the user and to take the user to determine that your use of the content to produce the ads. User configuration files for a file can be in some cases through a user completing a registration form. In other cases, personal information, users can browse and Monitoring of users are likely to follow. In most cases this is likely to produce ads that surrogate master website in CDN.

A useful application that is surrogate site to create a website in an ad aimed at local users. If additional resources are CDN routing mechanism to maintain some degree of rough
geographical proximity, it is likely that most customers use the surrogate mother from the same geographical area. Therefore, we can have a proxy server; the ads are targeted at local audiences. Proxies server is located in Stockholm may be ads for the local football team, as long as the Sports page of the newspaper's visit and the surrogate server is located in Lund, Karlskrona, produces a different ad groups.

Advertising agency, in order to generate these sites, proxy sites need to know how to map a URL category, select the ads fall into this category. If we take into account the logic of mapping the classification of the site a large scale, surrogate mother, that the site can be a part of this table in the cache, and to obtain the correct mapping of the URL cache. There are two ways to handle the URL cache type mapping table. If a credit institution is encoded in a relatively simple process and procedures, it is possible to implement the same procedures, the holder of the proxy site. Otherwise, they may use a surrogate mother a site plan to cache query results, the output of the program (type) is cached for each input (URL) that the surrogate site to see so far. If you have a web site, how it is, and this cache can be quite effective. Table or procedures to define URL mapping will be updated regularly category new statistics. However, since this change is not expected to have more than once a day, the cache consistency with the simple mechanism (for example, a limitation one day) would be sufficient to produce a reasonable and accurate web site to type mapping.

Sharing your personal data may be used for information which ads surrogate server and customized web pages according to individual user preferences.

7.5 CONTENT ADAPTATION

Client to obtain information on the Internet (or corporate intranet), including users of personal computers and the use of handheld devices and PDAs. Because of different access devices, simple HTML format web page is no longer the needs of all users. Access to web-page, portable devices need to use the data used in the WAP protocol instead of HTTP to use the browser's usually on the desktop. Even for the contents of the computer and the standard HTTP, the user's personal preferences may vary. Users Sweden hopes that the website in Swedish and users in the United States would like to see all pages in English. The content that the server must respond to user needs.

If the server knows the client preferences, it may take the form of content, are ideally suited to accept the client. However, in many cases, the server can not know the preferences. Server is located in Lund, Sweden, visit the user world. It is very difficult to know whether the mother tongue of the client to server, if the information is present in the user's personal information, or a cookie. Not too many servers require this information to the user. On the other hand, if the Contents Distribution Networks, as well as the many cases in which the proxy server, an option the possibility of increasing the default language that customers prefer the local language.
It is the content delivery network, where all the surrogate mother has a WAP site to support the function means that the server does not have to worry about support for wireless devices. If the same network as the proxy sites are used to support many of the origin of the spot, is not the source of the site must support WAP.

7.6 MAIL SERVER

Although much less than the actual volume of traffic to the internet and internal network users web, e-mail remains a key application of opening the internet technology. The mail server is a machine, storage, e-mail users, enabling them to read and send e-mail needs. Large number of different agreements that can be used for the users and the mail servers. Products such as Microsoft Exchange and Lotus Notes e-mail to provide professional services to the use of the client and the server. Some of the open protocol that can be used between the customer's mail server, including POP (Post Office Protocol) and IMAP (Internet Message Access Protocol). IMAP allows a user database, e-mail access to remote servers. POP is a protocol that allows users to download e-mail database of licensed remote server is on a local file system. Some mail servers have access to the user's e-mail, Web browsers via HTTP. When the e-mail server is required in Exchange e-mail belonging to different users, they are usually used in SMTP (Simple Mail Transfer Protocol).

When the user in the remote location, how the mail server can become rather slow. In this case, the performance can be significantly improved mobile Web servers closer to Client by exploiting a CDN. In such cases, the Web servers to provide better e-mail into the user's cache content closer to them.

![Figure 10; Mail server setup](file)

63
Two features require special attention to access e-mail proxy server in place. The first is the issue of registration; the other is the client authentication. When users arrive at the new system should be recorded on the original site or through a proxy server for your site? Both solutions are technically viable, but the personal information by e-mail users will see the CDN in the second case the site. If additional CDN to the sites belonging to different administrative domain (for example, the CDN provides a third party), but the limitations of this medium-registration site is the best approach. When the same environment of origin, the site may be a surrogate mother does not want to use a password to their users. In this case, the plan describe the secure content caching, the use of hidden on the e-mail user agent's website. This method is used, you must first verify the origin of your own site. Origin of the site, the customer has tickets available for a surrogate mother to visit a Web site's e-mail database. Tickets are available at these example cookies, the HTTP protocol.

7.7 SHOPPING SITES

The most common types of website on Internet provide e-marketing projects, books, CDs, software, tickets, and even our daily food items in the stores. Many web sites can be significantly more effective use of metadata. There are some views; you can do about the majority of online shops. Most online shops, most users tend to browse the directory, rather than the actual purchase. Between 80-90% throughout the visit, the e-content and search requests through a catalog. Only a small number of users actually buy the merchandise in stores. However, the browser that the user can interfere with the performance of the buyer, and each may have a poor performance; many users are simultaneously active in the online store.

Navigational features are read-only operation, may be transferred to the surrogate site grows within a CDN. Users can browse the entire catalog or search for specific pages. Proxy’s site meets these requirements to retain copies of the directory to copy or use the query cache. The survey provides a cache of good work has many different types of queries more often than others.

A real online store, which is necessary for operation and navigation, is to keep the shopping cart. Client projects in addition to the online shopping cart while browsing the directory. Mobile browsing the web site is the surrogate sites is not effective, mobile properties, and the basket. Fortunately, the manipulation Cart necessary to update or change information, it must be unique for each user, and it does not require synchronization. It can therefore be considered as one alternative to the basket to carry out the client browser. When customers buy items in shopping cart ready, he will take back-end web site.
Chapter 8

CDN Pricing

Content Delivery networks play an important role in media delivery over the internet. As with time the broadband adoption and the shift of customers towards the multimedia content delivery. The CDNs are a unique part of this valued supply chain of content and hence are used by a large number of content providers. Content providers are having very significant economic benefits by employing CDNs in their content delivery. As CDN moves the content to the edges of the network and making the content delivery more accessible and speedy. This technique aggregates the customer traffic across several sites (within the CDN) and hence they help in achieving the economies of scale in infrastructure. This cost of delivering content to the customers (using CDN) is much lower then the cost of delivering the content (by a small content provider) on its own.

Traffic aggregation also helps in decreasing the impact of demand variation for the content. As the different content providers are cooperating with each other (in a CDN) to for content delivery so it is unlikely that all the content providers are surged at the same time because of a sudden increased demand of any content within the CDN. Moreover, as multiple servers are present in a CDN and it can serve the customer from any one of them, so there is no single point of failure in the CDN. As result CDN is not compromised during sudden increased demand or denial of service attacks.

The maintenance of a content delivery network and hence the content delivery to the customers requires a cost for the continuation of a better and reliable service. So CDN pricing is of very much importance in order to make sure that the content providers can continually get the valued service of CDN and similarly CDN providers have incentives in deploying and managing the infrastructure.

In CDN market CDN pricing has gained significant importance and the pricing policies have great effects on the CDN providers and the subscribing content providers. Issues such as bursty traffic impacts on pricing, use of large volume discounts and the need of new pricing policies are of great interest. Later on, we will look at different CDN price models and the issues related to them.

Pricing the CDN

So many pricing methods are used in the CDN market and all of these methods have a usage component. Lets examine the two most common pricing methods used for CDN.

Aggregate Usage

This pricing model is one of the simplest models. In this approach, a content provider tells the CDN operator about its level of usage e.g. 70 TBs/month. The CDN operator then determines the cost on the basis of this usage which is in terms of per GB delivered.
For this usage plan, there is also a volume discount e.g. 0.5 euros/GB for a usage level of 70TBs/month and 0.3 euros for a usage level of 100 TBs/month. So the price of the CDN decreases as the volume usage an increase that’s one of the key advantages of using CDN. But if the usage level is exceeded some CDN operators charge a penalty from the content provider in order to make the correct estimate of its usage level which in turn also helps the CDN operator in better capacity planning of CDN.

**Percentile Based**

In this method, the subscriber is charged on the basis of the 95th percentile of usage level, the CDN operator notes the usage level of the subscriber from time to time. The operator then calculates the 95th percentile of subscriber’s usage level at the end of each month. The subscriber is then charged per Mpbs on the basis 95th percentile of usage level. Most of the time, both methods (aggregate based & 95th percentile based) are offered by the CDN operators and it’s the choice of subscriber to select any one of them according to its usage level requirements.

**CDN Financial Market**

Content delivery network’s have long being in the field of content delivery to the users but most of the time in a different way of implementations such as in ISPs and web servers. But as the need for more bandwidth and multimedia content increased with time. The content delivery was no more just a side business o any ISP and the commercial market begin to realize the its importance as a separate bussiness and a valuable service not only to individual users but also to big companies starvi ng for faster and economical content delivery services.

The first commercial CDN appeared in 1998 when the Companies started to realize that the use of a CDN service for their websites could save them money and they will also get increased reliability and scalability without employing expensive hardware. As more and more companies started to put their website on CDNs and began to avail the CDN services. The CDN business got momentum, so at the same time other companies like Akamai and Mirror Image employed large CDNs and started to establish a highly profitable CDN business. They became the specialists in providing fast and reliable delivery of Web content and at the same time earned large profits from their valuable CDN services.

In year 2000, the CDN business begins to expand rapidly and captured a lot of market shares. In U.S only, CDNs made a huge market generating $905 million with the expectation to reach $12 billion by 2007. This high growth in the CDN market made it so popular among the customers and continued to grow rapidly. The CDN also proved to be reliable in the event of flash crowd (numerous users accessing a website at the same time). Such an event happened on Sept 11, 2001 when users flooded popular news sites with requests about the terrorist attacks in the U.S., resulted in serious caching problems since sites had typically become unavailable. Flash events transfer more dollars to CDN
sales income, since CDNs provide the desired level of protection to Web sites against them.
This rise in growth impressed more companies and in 2002 large-scale ISPs (such as AT&T) tend to build their own CDN functionality, providing customized services to their users. In 2004, more than 3,000 companies use CDNs, spending more than $20 million monthly [8]. Marketing research [8] shows that CDN providers have doubled their revenue derived from their streaming media operations in 2004 compared to 2003. Furthermore, many CDN providers are trying to move Web services (such as Microsoft .NET and Java 2 Platform Enterprise Edition) closer to users. Furthermore in 2005, CDN revenue for both streaming video and Internet radio is estimated to grow at 40%, spending more than $450 million for delivery of news, film, sports, music, and entertainment.
In today’s financial market of CDN, the major companies are:

1. Akamai Technologies
2. Limelight Networks
3. Mirror Image

Akamai is the up till now standing as the leader in the CDN financial market and continually providing new and faster services. Other CDN providers are also fast runners and trying to catch up with the leading companies. As the internet is getting more and more bandwidth hungry, the companies are trying to offer more and more to its clients and trying their best to coup with the Internet. This competition among the different companies is very useful for the customers as they have a lot of choices for getting their needs fulfilled and also at a less expensive cost.

**Conclusion**

The life of any technology depends on the amount of research work made for it and most importantly its demand in the open market. Today the CDN technology has both of these aspects in its favor. Everyday new bandwidth hungry softwares and services are coming into market so users need more from their ISPs and web servers. As this trend seems to increase in the near future so do the need of CDN will be there to provide contents at in less time and good quality. If we look at the new interactive websites such as video streaming and gaming websites the need for best content delivery is always there. As these sites are very time sensitive and little delay in the content delivery could result in inconsistent display. So to take care of such issues the website provider has to be very cautious in selecting the right type of technology for delivering its content.

CDN financial market is constantly expanding and growing. With new companies coming into this field, the CDN market has grown to over billions of dollars (U.S) and has achieved a quite stable financial position. The ongoing competition in the financial market has benefited the users very much and they can easily select and use the service according to their needs.
After studying the CDN technology and having a brief look at its financial market. We can easily say that CDN can still last for a longer period in the market. Although new technologies are always emerging and making their way to the market.
Chapter 9

Related & Future Work

Content delivery needs are increasing everyday and every now and then new companies are coming into market. As every company has its own way of working. So In order to maintain good and viable services and to also take care of the legal issues the CDN is standardized to meet the best possible service to the customers. The following section will discuss these standards briefly and the new work being carried out in the field of content delivery.

9.1 CONTENT DISTRIBUTION STANDARDS

These standards relate to the content of the progress of some of the industry league and forums, mainly in IETF (Internet Engineering Task Force). IETF standards for the determination of the working groups and specific topic of interested. Before the task force made up of the subject has been discussed in BOF (birds of a feather) meeting. This time, there is a working group and two treatment-related BOFs problem CDNs. This two BOFs task force is expected to be followed.

The WEBI (Web Intermediaries) working groups are trying to determine protocol to maintain the network resources can be used to update the usual web-based proxy server. Meanwhile, the group also attempted to define the requirements of Protocol will need to find an intermediary proxy server. OPES converter will examine opportunities to establish a working group OPES (Open extensible service). The Working Group defines the OPES architecture scalability agent, which can perform various functions of the website to add surrogate site in the CDN. CDI (Content Distribution Network Interconnection), the converter will examine issues relating to peer-to-peer content distribution networks and the identification of mechanisms, distribution and routing services at the request of the agent.

9.2 PEER-TO-PEER CDNs

The word CDN and sometimes applied to an ad hoc peer-to-peer network, can form around a machine independent communication, the formation of overlay network. Peer-to-peer networks has received great popularity because it is used to exchange digital music files on the Internet, some of the projects are specific to them. Lot of the computer peer-to-peer networks are interconnected by an ad hoc basis. Replace the traditional client-server model, which is one of the client retrieval of documents known as a server; the document is available on any machine to participate in peer-to-peer network. In order to use the content of peer-to-peer network, the user begins with a description of the property information, he can find. The next step is to find a network engineer responsible for the documents needed for the property. And the document search.
To find the right content is two methods can be used. The first way is to use a medium-index server. Each participant was to report on types of cases the index server; anyone can look for information on the search index to a server matching documents and their location. This approach is used in the Napster file-sharing programs. Another way is to form overlay network linking all the participants. The survey participants are broadcast networks, along until the files corresponding to the query results. And the document is posted to the original query.

Broadcast message can also be done on many fronts. The simplest solution is to use flooding. This is the method used in the Gnutella software. Each participant must submit the question to all other participants and connect. Flood program to avoid the cycle of forever package, for example, limit the total number of hops investigation will be forwarded to each node or only the search process to meet for the first time. The first can be done with the use of time and life of research, which has increased each node; Studies discarded if an increase beyond the realm of the limit. IP before using the same technology, but the data packets from a time value of the survival of the sector and its value decreases each jump, and if the package has rejected “0”.

Each node, if necessary, to deal with only one query, a node must not forget that he has the necessary investigation or research, which includes a node, is over. Two systems have the time, whether it or the hub. This means that the limited space requirements. Since the space to open the window, the great cycle of existence, they have been used in the survival time. For example, if the question for the survival of K hops of nodes less than necessary to remember, K. If the question contains a list of nodes that have changed a few copies will be able to process up to the question occurs at the same place on two different ways.

The flood is an extreme form of travel in a little coordination between the participants. The Less extreme form of routing to forward the request as the routing mechanism the flood is an extreme form of travel in a little coordination between the participants. A less extreme form of routing to forward the request as the routing mechanism to find way to implement various layer’s multicasting.
References


