

Distributed Web-Based Systems (ch12)

Supervisor: DR:- Sheeren Ahmed

Prepared by

Nada Mohamed ahmed

Architecture

Aya Adel Mahmoud

Process

Magda Saif Shaban

Communication

Sally Ahmed Sayed

Naming and Synchronization

Hossam Hassan Ahmed

Replication

outline

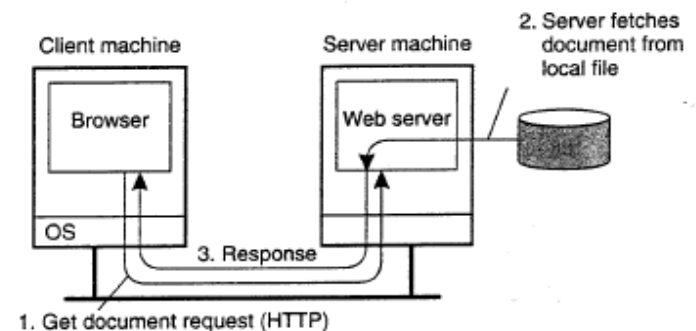
- Architecture.
- Process.
 - Clients.
 - The Apache Web Server.
 - Web Server Clusters.
- Communication.
 - HTTP Protocol.
 - SOAP Protocol.
- NAMING.
- SYNCHRONIZATION.
- Replication for Web Hosting Systems.

Architecture

- Distributed system: is a system whose component are located on diff networked computer, then communicate their actions by passing messages to one another
- World Wide Web (WWW).
 - A huge distributed system that consisting a million of clients and servers for accessing linked documents.
 - Servers maintain collections of documents, while clients provide users an easy-to-use interface for presenting and accessing those documents.
 - A document was fetched from a server, transferred to a client, and presented on the screen.

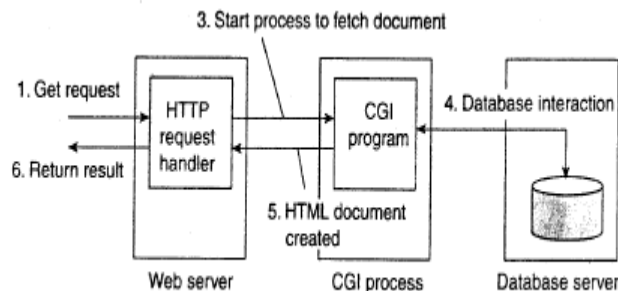
Architecture

- Traditional Web-Based Systems.
 - Many Web-based systems are still organized as relatively simple **client-server architectures**.
 - A client interacts with Web servers through a browser.
 - A browser accepts input from a user mostly by letting the user select a reference to another document, which it then subsequently fetches and displays.
 - The communication between a browser and Web server is : they both adhere to the Hyper Text Transfer Protocol (HTTP).
 - How to refer the web document?
 - Uniform Resource Locator (URL).



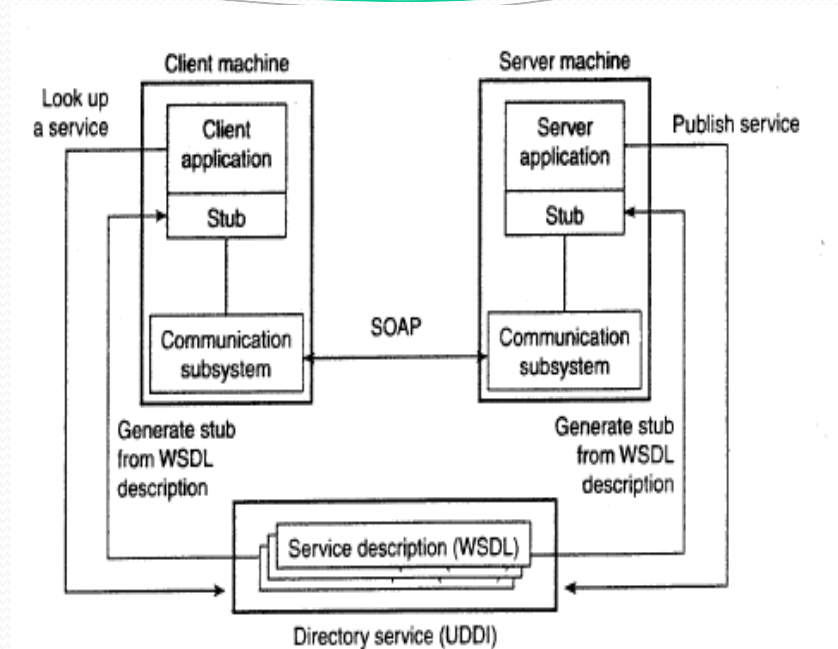
Architecture

- Web document.
 - hypertext markup language (HTML) and eXtensible markup language (XML)
 - multipurpose internet mail exchange (mime)
- Multitiered Architectures: Web document can build in 2 way:
 - Static.
 - Dynamic.
 - CGI



Type	Subtype	Description
Text	Plain	Unformatted text
	HTML	Text including HTML markup commands
	XML	Text including XML markup commands
Image	GIF	Still image in GIF format
	JPEG	Still image in JPEG format
Audio	Basic	Audio, 8-bit PCM sampled at 8000 Hz
	Tone	A specific audible tone
Video	MPEG	Movie in MPEG format
	Pointer	Representation of a pointer device for presentations
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in Postscript
	PDF	A printable document in PDF
Multipart	Mixed	Independent parts in the specified order
	Parallel	Parts must be viewed simultaneously

Architecture



- Web Services.
 - method of communication between two devices over network
- Web Services Components.
 - SOAP (Simple Object Access Protocol)
 - WSDL (Web Service Description Language)
 - UDDI (Universal Description, Discovery and Integration

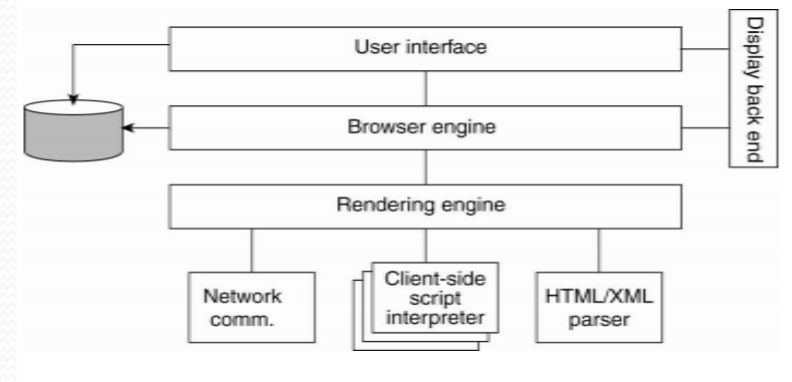
Clients

- **Web client** is a piece of software called a Web browser, which enables a user to navigate through Web pages by fetching those pages from servers and subsequently displaying them on the users' screen.
- A **browser** typically provides an interface by which hyperlinks are displayed in such a way that the user can easily select them through a single mouse click.

Clients cont.

➤ Important aspect of Web browsers:

1. Platform independent.
2. Should be easily extensible so that it, can support any type of document that is returned by a server.



The Apache Web Server

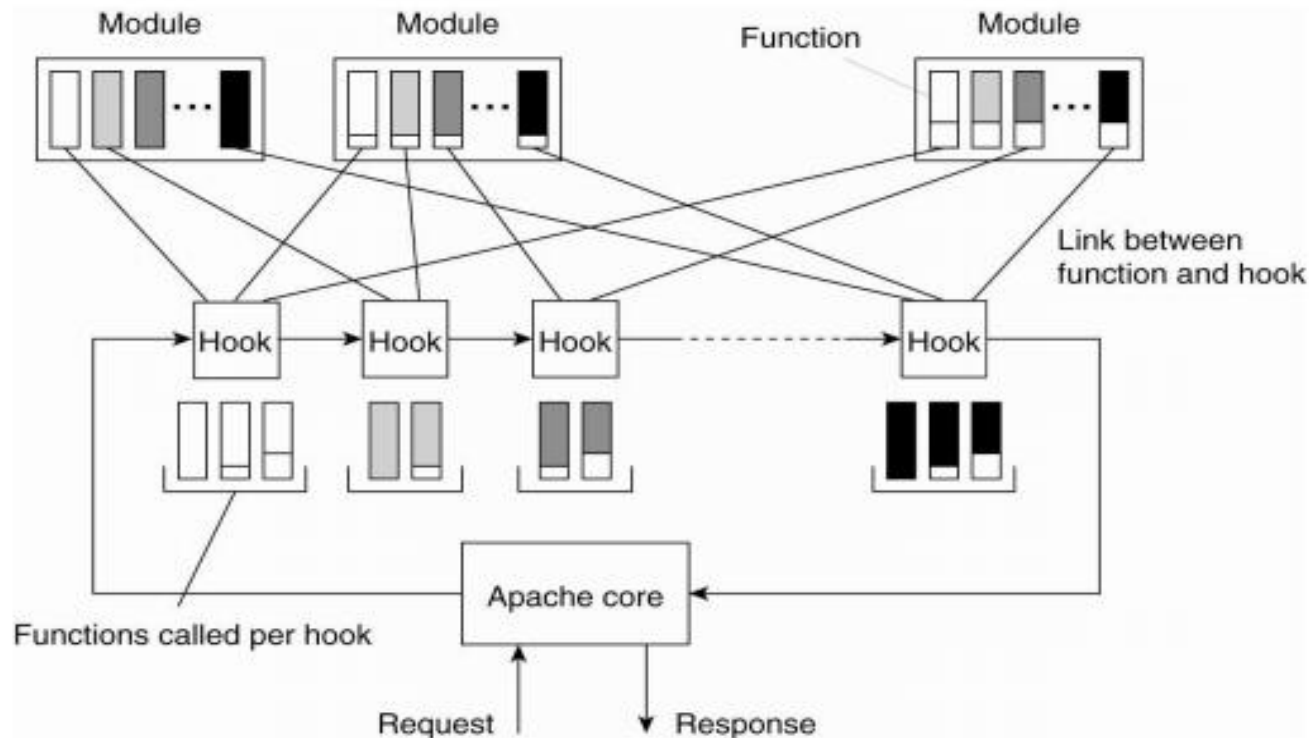


Figure 12-7. The general organization of the Apache Web server

The Apache Web Server

- By far the most popular Web server is Apache, which is estimated to be used to host approximately 70% of all Web sites.
- Apache's runtime environment, known as the Apache Portable Runtime (APR), is a library that provides a platform-independent interface for file handling, networking, locking, threads, and so on.
- The Apache core makes few assumptions on how incoming requests should be handled. Its overall organization. Fundamental to this organization is the concept of a hook, which is nothing but a placeholder for a specific group of functions. see that Apache enforces a specific flow of control concerning the processing of requests.
- The functions associated with a hook are all provided by separate modules.

The Apache Web Server cont..

- For example:
- There is a hook to translate a URL to a local file name. Such a translation will almost certainly need to be done when processing a request.
 1. Likewise, there is a hook for writing information to a log,
 2. A hook for checking a client's identification,
 3. A hook for checking access rights
 4. A hook for checking which MIME type the request is related to (e.g., to make sure that the request can be properly handled).
- the hooks are processed in a predetermined order. It is here that we explicitly

Web Server Clusters

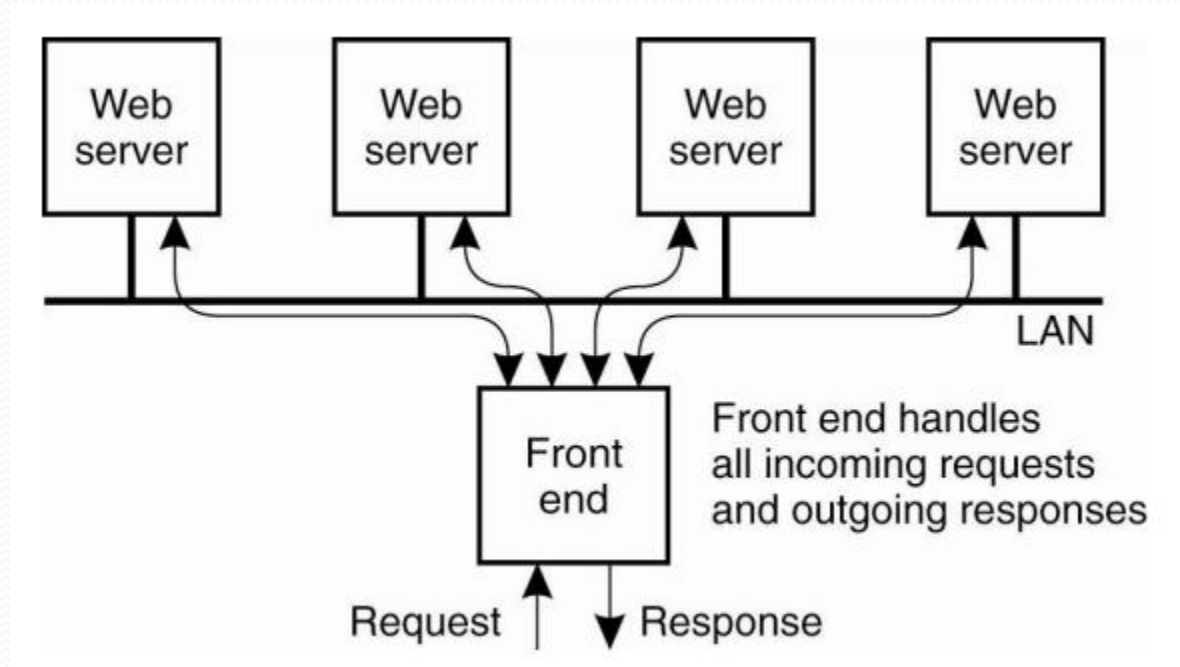


Figure 12-8. The principle of using a server cluster in combination with a front end to implement a Web service.

Web Server Clusters

- An important problem related to the client-server nature of the Web is that:-
 - Web server can easily become overloaded.
 - A practical solution employed many designs is to simply replicate a server on a cluster of servers and use separate mechanism, such as a front end, to redirect client requests to one he replicas.
- A crucial aspect of this organization is the design of the front end as it can become a serious performance bottleneck, what will all the traffic passing through it.
- In general, a distinction is made between front ends operating as **transport layer switches**, and those that operate at the level of the application layer.

Web Server Clusters cont...

- The main drawback of a transport-layer switch is that the switch cannot take into account the content of the HTTP request that is sent along the TCP connection. At best, it can only base its redirection decisions on server loads.

Web Server Clusters

- As a general rule, a better approach is to deploy content-aware request distribution, by which the front end first inspects an incoming HTTP request, and then decides which server it should forward that request to. .
- For example, if the front end always forwards requests for the same document to the same server, that server may be able to effectively cache the document resulting in higher response times.
- In addition, it is possible to actually distribute the collection of documents among the servers instead of having to replicate each document for each server.

This approach makes more efficient use of the available storage capacity and allows using dedicated servers to handle special documents such as audio or video.

Web Server Clusters

- Content-aware distribution has several advantages.
- A problem with content-aware distribution is that the **front end needs to do a lot of work.**

Ideally, one would like to have the efficiency of TCP handoff and the functionality of content-aware distribution

Web Server Clusters

- These two tasks can be distributed as
- **The dispatcher** is responsible for deciding to which server a TCP connection should be handed off.
- **A distributor** monitors incoming TCP traffic for a handed-off connection.
- **The switch** is used to forward TCP messages to a distributor.

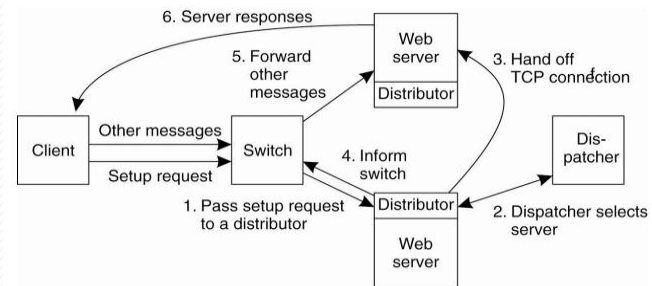


Figure 12-9. A scalable content-aware cluster of Web servers.

communication

HTTP (Hypertext Transfer Protocol)

- Is the standard protocol for exchanging messages.
- Used for traditional web systems.
- HTTP is a relatively simple client-server Protocol.
 - a client sends a request message to a server and waits for a response message.

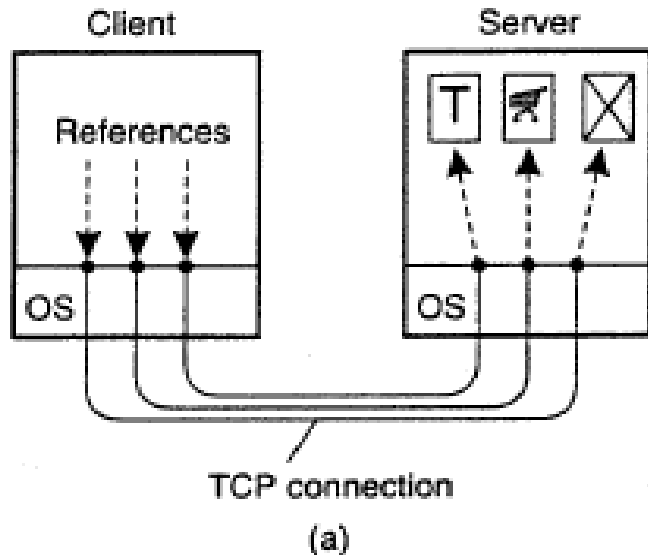
Http topics :

- HTTP Connections.
- HTTP Methods.
- HTTP Messages.

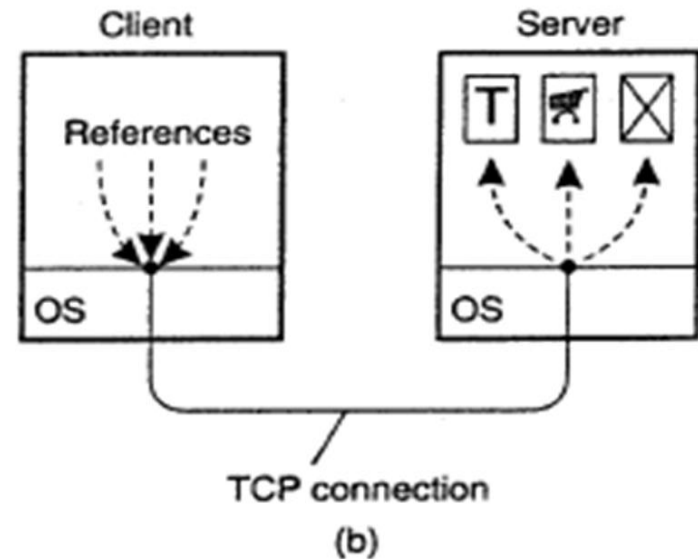
HTTP Connections

-HTTP is based on TCP.

- HTTP version 1.0
- **Nonpersistent** connections



- HTTP version 1.1
- **Persistent** connections



HTTP Methods

- Some operations that added to request to be carried out at the server

A list of the most commonly-used request messages is given in Fig. 12-11.

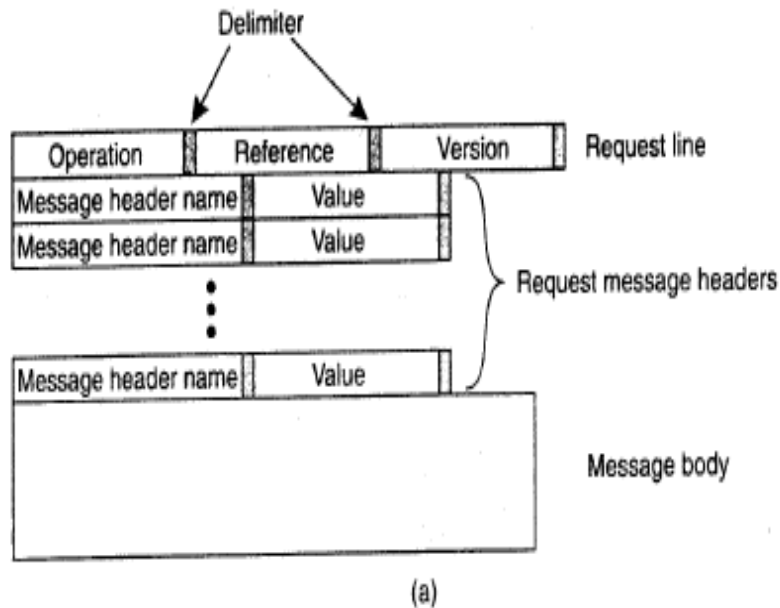
Operation	Description
Head	Request to return the header of a document
Get	Request to return a document to the client
Put	Request to store a document
Post	Provide data that are to be added to a document (collection)
Delete	Request to delete a document

Figure 12-11. Operations supported by HTTP.

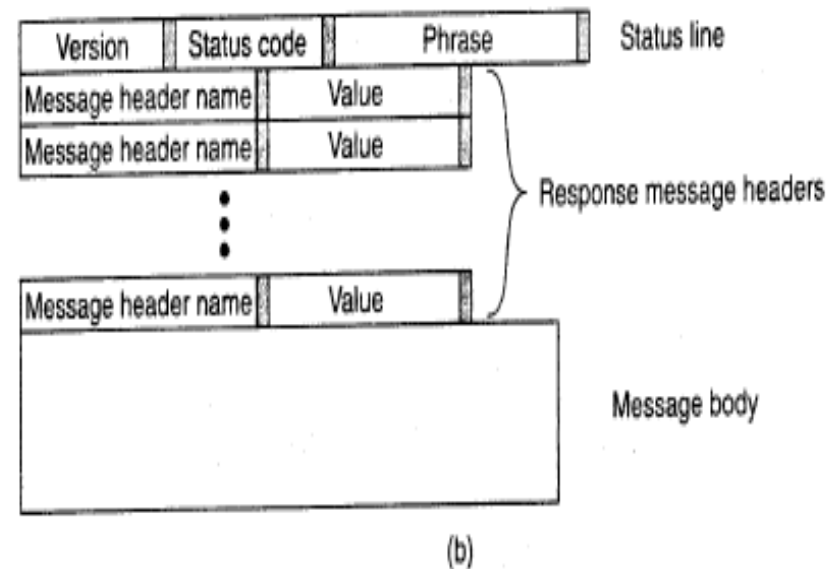
HTTP Messages

- All communication between a client and server takes place through messages.
- Messages consists of three parts
 - Message Start-Line
 - Header Fields
 - Message Body(optional)

- A request message consists of three parts.
 - The request line.
 - Request Header.
 - Message Body



- A response message consists of three parts.
 - The status line.
 - Response Header.
 - Message Body



Some http message headers

Header	Source	Contents
Accept	Client	The type of documents the client can handle
Accept-Charset	Client	The character sets are acceptable for the client
Accept-Encoding	Client	The document encodings the client can handle
Accept-Language	Client	The natural language the client can handle
Authorization	Client	A list of the client's credentials
WWW-Authenticate	Server	Security challenge the client should respond to
Date	Both	Date and time the message was sent
ETag	Server	The tags associated with the returned document
Expires	Server	The time for how long the response remains valid
From	Client	The client's e-mail address
Host	Client	The DNS name of the document's server
If-Match	Client	The tags the document should have
If-None-Match	Client	The tags the document should not have
If-Modified-Since	Client	Tells the server to return a document only if it has been modified since the specified time
If-Unmodified-Since	Client	Tells the server to return a document only if it has not been modified since the specified time
Last-Modified	Server	The time the returned document was last modified
Location	Server	A document reference to which the client should redirect its request
Referer	Client	Refers to client's most recently requested document
Upgrade	Both	The application protocol the sender wants to switch to
Warning	Both	Information about the status of the data in the message

Figure 12-13. Some HTTP message headers.

SOAP

(Simple Object Access Protocol)

- SOAP forms the standard for communication with **Web services**.
- this protocol is designed with the assumption that two communicating parties have very little common knowledge.
- Based on this assumption, SOAP messages are based on XML.
 - “XML is a meta-markup language”
 - Meaning that an XML description includes the definition of the elements that are used to describe a document.
- this means that the definition of the syntax as used for a message is part of that message.

SOAP Messages

- A SOAP message generally consists of **two parts**, which are jointly put inside what is called a **SOAP envelope**.
- The body
 - contains the actual message.
- the header “optional”
 - containing information relevant for nodes along the path from sender to receiver.

SOAP envelope

- Is the root element in every SOAP message.
- Everything in the envelope is expressed in XML, that is, the header and the body.
- SOAP envelope does not contain the address of the recipient. Instead, SOAP explicitly assumes that the recipient is specified by the protocol that is used to transfer messages.
 - Such as → “HTTP -SMTP” protocols

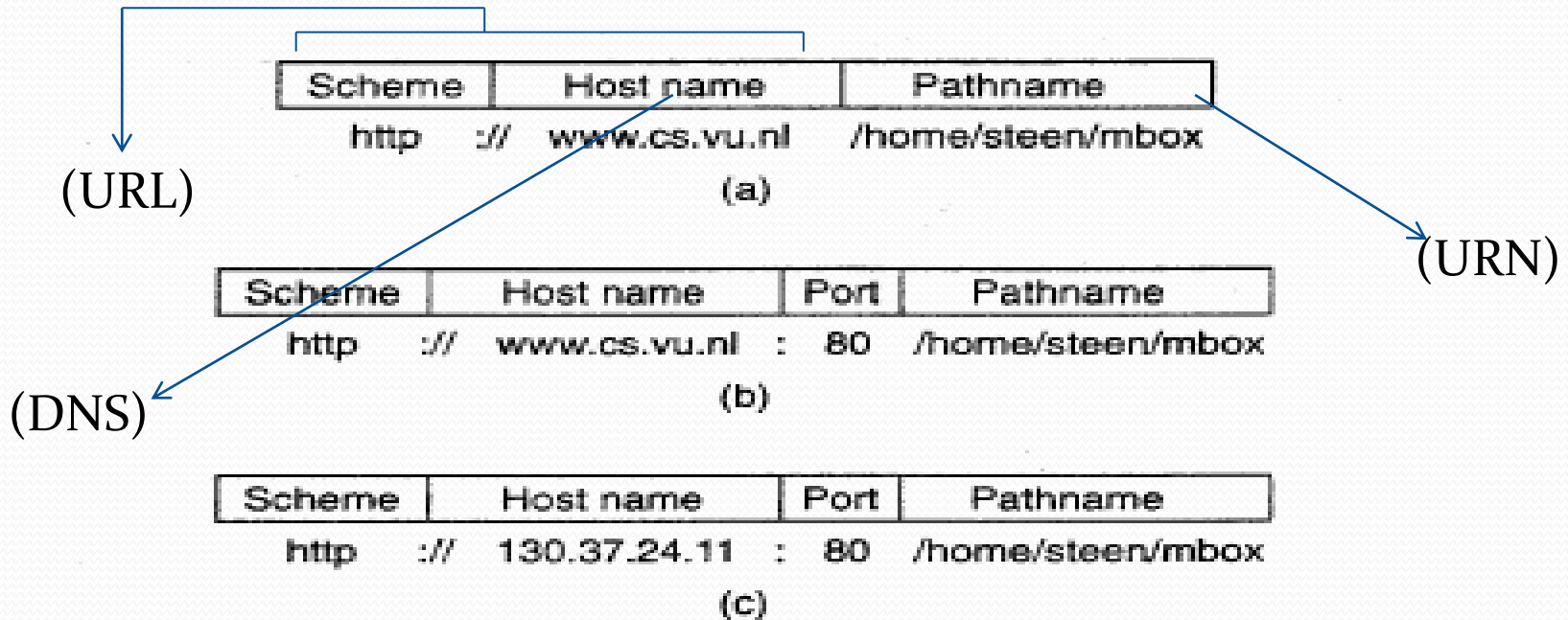
```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

Figure 12-14. An example of an XML-based SOAP message.

NAMING

- The Web uses a single naming system to refer to documents. The names used are called Uniform Resource Identifiers (URI)
- URIs contain URL and URN:-
 - 1- (URL) Uniform Resource Locator that including information on how and where to access the document. (**How** such as http or ftp) and **Where** a document is located is embedded in a URL by means of the DNS (Domain Name Server) name of the server to which an access request can be sent.
 - 2- (URN) Uniform Resource Name is used as a globally unique and persistent reference to a document.

NAMING cont.



- Often-used structures for URLs.
- (a) Using only a DNS name.
- (b) Combining a DNS name with a port number.
- (c) Combining an IP address with a port number.

NAMING cont.

Name	Used for	Example
http	HTTP	http://www.cs.vu.nl:80/globe
mailto	E-mail	mailto:steen@cs.vu.nl
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:/edu/book/work/chp/11/11

Examples of URIs.

SYNCHRONIZATION

- Synchronization has not been much of an issue for most traditional Web based systems because the traditional Web can be considered as being a read-mostly system and Updates are generally done by a single person.
- Now, The Web should provide support for concurrent updates of documents by a group of collaborating users or processes.
- Distributed authoring of Web documents is handled through a separate protocol (WebDAV) Web Distributed Authoring and Versioning and provides a simple means to lock a shared document, and to create, delete, copy, and move documents from remote Web servers

SYNCHRONIZATION cont.

- To synchronize concurrent access to a shared document, WebDAV supports a simple locking mechanism.

There are two types :-

- First : An exclusive write lock can be assigned to a single client, and will prevent any other client from modifying the shared document while it is locked
- Second : There is also a shared write lock, which allows multiple clients to simultaneously update the document. Because locking takes place at the granularity of an entire document, shared write locks are convenient when clients modify different parts of the same document. However, the clients, themselves, will need to take care that no write-write conflicts occur.

Replication for Web Hosting Systems

- What is replication challenges ?
 - 1- Replica Placement
 - 2- Consistency Enforcement
 - 3- Request Routing
- What is content delivery networks (CDNs)?

CDNs is a Web hosting service, providing an infrastructure for distributing and replicating the Web documents of multiple sites across the Internet

Documents are automatically distributed

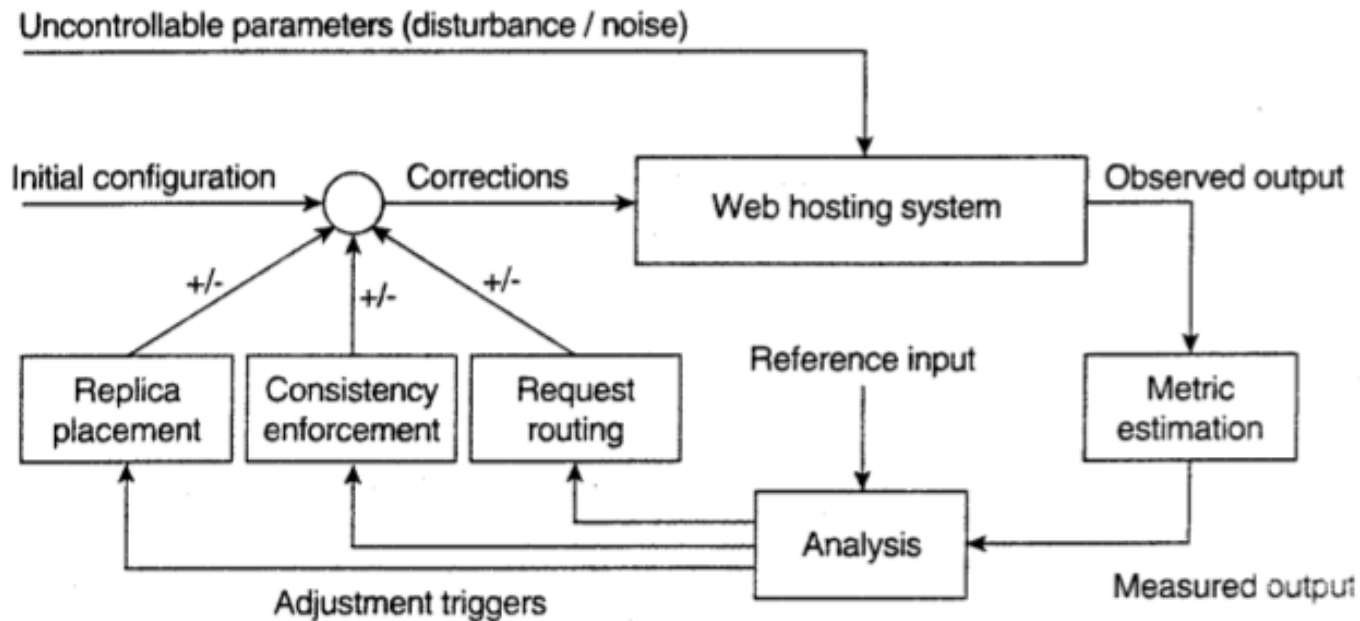


Figure 12-18. The general organization of a CDN as a feedback-control system (adapted from Sivasubramanian et al., 2004b).

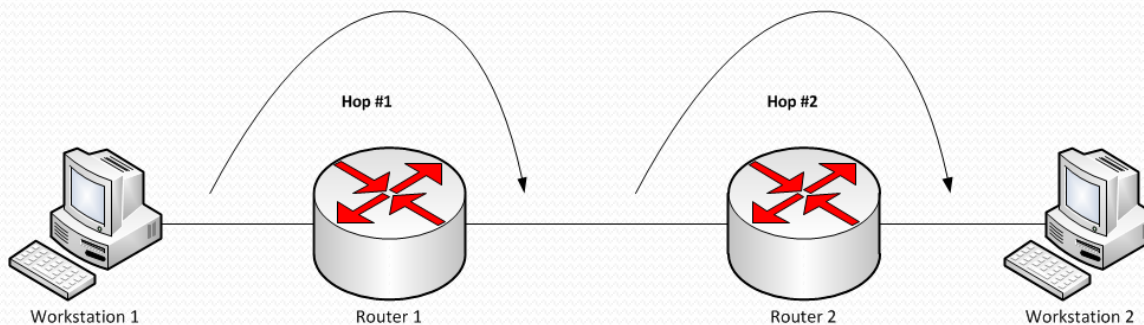


Documents are automatically distributed cont.

- Metric Estimation
- Adaptation Triggering
- Adjustment Measures

Metric Estimation

- Latency metrics:- time is measured for an action
- Spatial metrics:- measuring the distance between nodes in terms of the number of network-level routing hops.



Metric Estimation cont.

- Network usage metrics:- Computing how often the document is read, how often it is updated, and how often it is replicated
- Consistency metrics:-what extent a replica is deviating from its master copy
- Financial metrics:- It's closely related to the actual infrastructure of the Internet.

Adaptation Triggering

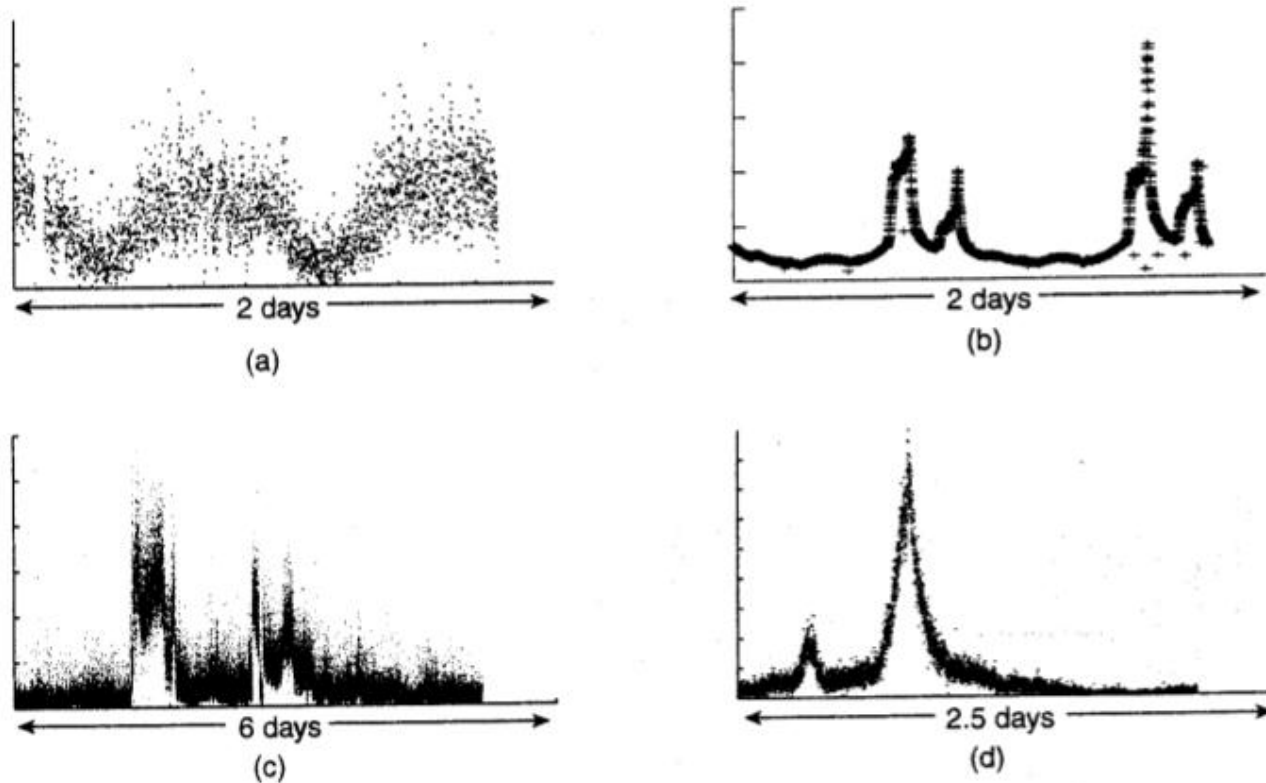


Figure 12-19. One normal and three different access patterns reflecting flash-crowd behavior (adapted from Baryshnikov et al., 2005).

Adaptation Triggering

Adaptation trigger in three models:-

- 1- Periodically
- 2- Flash-crowd predictor:-that will provide a server enough time to dynamically install replicas of Web documents, after which it can redirect requests.
- 3- simple linear extrapolation technique:-
continuously measure the number of requests to a document during a specific time. If the number of requests are predicted to exceed a given threshold, an alarm is raised

Adjustment Measures

- there are essentially only three (related) measures that can be taken to change the behavior of a Web hosting service:-
 - 1- Changing the placement of replicas
 - 2- Changing consistency enforcement
 - 3- Deciding on how and when to redirect client requests

Adjustment Measures Example

- Idea:- Each Web document consists of a main HTML (or XML) page in which several other documents such as images, video, and audio have been embedded.

To display the entire document, it is necessary that the embedded documents are fetched by the user's browser as well. The assumption is that these embedded documents rarely change, for which reason it makes sense to cache or replicate them

Adjustment Measures Example

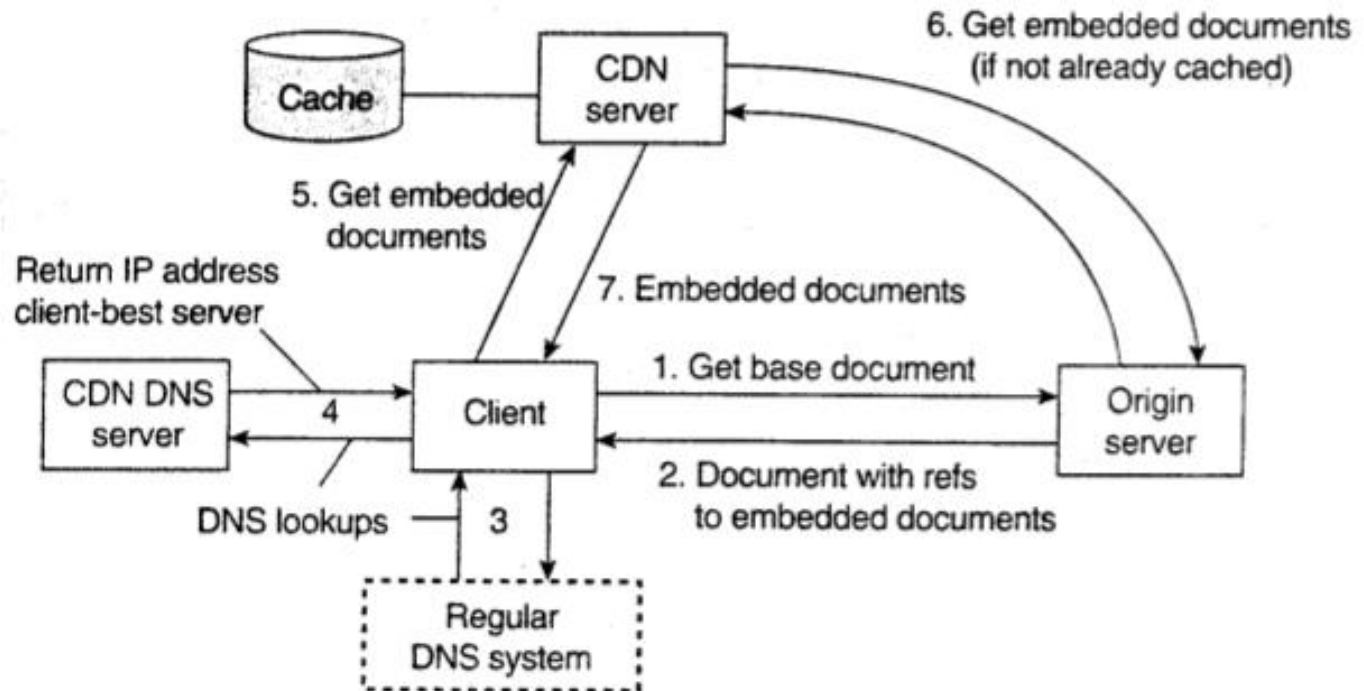


Figure 12-20. The principal working of the Akamai CDN.



Thank you