**UNIT IV**

# DNS—The Domain Name System

DNS is hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names and e-mail destinations to IP addresses. To map a name onto an IP address, an application program calls a library procedure called the resolver, passing it the name as a parameter. The resolver sends a UDP packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller. Armed with the IP address, the program can then establish a TCP connection with the destination or send it UDP packets.

# DNS Name Space

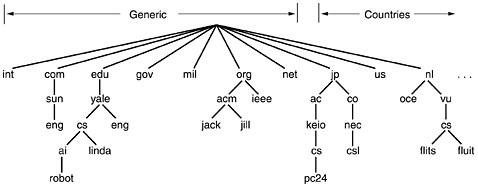


Figure: A portion of the Internet domain name space.

The Internet is divided into over 200 top-level **domains**, where each domain covers many hosts. Each domain is partitioned into subdomains, and these are further partitioned, and so on. All these domains can be represented by a tree, as shown in Figure.

* + The leaves of the tree represent domains that have no subdomains. A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.
  + The top-level domains come in two flavors: generic and countries. Each domain is named by the path upward from it to the root. The components are separated by periods (pronounced ''dot'').
  + Domain names can be either absolute or relative. An absolute domain name always ends with a period (e.g., *eng.sun.com*.), whereas a relative one does not. Relative names have to be interpreted in some context to uniquely determine their true meaning. In both cases, a named domain refers to a specific node in the tree and all the nodes under it.
  + Domain names are case insensitive, Component names can be up to 63 characters long, and full path names must not exceed 255 characters.
  + To create a new domain, permission is required of the domain in which it will be included.
  + Naming follows organizational boundaries, not physical networks.

# Resource Records

Every domain has a set of resource records associated with it. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. Thus, the primary function of DNS is to map domain names onto resource records. A resource record is a five-tuple.is as follows:

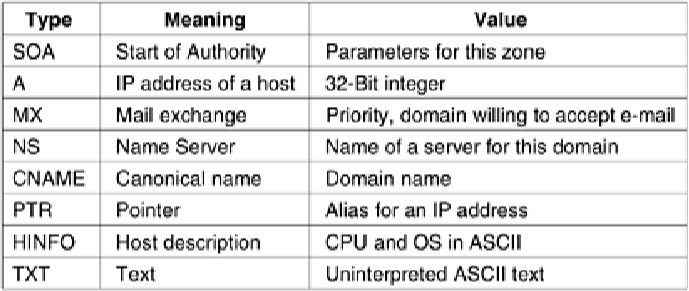
Domain\_name

Time\_to\_live

Class

Type Value

* + The Domain\_name tells the domain to which this record applies.
  + The Time\_to\_live field gives an indication of how stable the record is.
  + The third field of every resource record is the Class. For Internet information, it is always IN. For non-Internet information, other codes can be used.
  + The Type field tells what kind of record this is. The most important types are:



* + The Value field can be a number, a domain name, or an ASCII string. The semantics depend on the record type.

# Name Servers

To avoid the problems associated with having only a single source of information, the DNS name space is divided into nonoverlapping **zones**. Each zone contains some part of the tree and also contains name servers holding the information about that zone. Normally, a zone will have one primary name server, which gets its information from a file on its disk, and one or more secondary name servers, which get their information from the primary name server. To improve reliability, some servers for a zone can be located outside the zone. Where the zone boundaries are placed within a zone is up to that zone's administrator. This decision is made in large part based on how many name servers are desired, and where.

When a resolver has a query about a domain name, it passes the query to one of the local name servers. If the domain being sought falls under the jurisdiction of the name server, it returns the authoritative resource records. An **authoritative record** is one that comes from the authority that manages the record and is thus always correct. Authoritative records are in contrast to cached records, which may be out of date. If, however, the domain is remote and no information about the requested domain is available locally, the name server sends a query message to the top-level name server for the domain requested.

**LDAP** (**Lightweight Directory Access Protocol**): It organizes information as a tree and allows searches on different components.

# Electronic mail

1. **Architecture and Services**

Email systems consist of two subsystems: the user agents, which allow people to read and send e-mail, and the message transfer agents, which move the messages from the source to the destination. The user agents are local programs that provide a command based, menu-based, or graphical method for interacting with the e-mail system. The message transfer agents are typically system daemons, that is, processes that run in the background. Their job is to move e-mail through the system.

Typically, e-mail systems support five basic functions:

* + **Composition** refers to the process of creating messages and answers.
  + **Transfer** refers to moving messages from the originator to the recipient. In large part, this requires establishing a connection to the destination or some intermediate machine, outputting the message, and releasing the connection. The e-mail system should do this automatically, without bothering the user.
  + **Reporting** has to do with telling the originator what happened to the message. Was it delivered? Was it rejected? Was it lost? Numerous applications exist in which confirmation of delivery is important and may even have legal significance.
  + **Displaying** incoming messages is needed so people can read their e-mail. Sometimes conversion is required or a special viewer must be invoked,
  + **Disposition** is the final step and concerns what the recipient does with the message after receiving it. Possibilities include throwing it away before reading, throwing it away after reading, saving it, and so on. It should also be possible to retrieve and reread saved messages, forward them, or process them in other ways.

In addition to these basic services, some e-mail systems, especially internal corporate ones, provide a variety of advanced features like mailboxes, mailing lists, carbon copies, blind carbon copies, high- priority e-mail, secret (i.e., encrypted) e-mail, alternative recipients if the primary one is not currently available, and the ability for secretaries to read and answer their bosses' e-mail.

# User Agent

E-mail systems have two basic parts, as we have seen: the user agents and the message transfer agents. A user agent is normally a program that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes. Some user agents have a fancy menu- or icon-driven interface that requires a mouse, whereas others expect character commands from the keyboard. Functionally, these are the same.

## Sending E-mail

To send an e-mail message, a user must provide the message, the destination address, and possibly some other parameters. The message can be produced with a free-standing text editor, a word processing program, or possibly with a specialized text editor built into the user agent. The destination address must be in a format that the user agent can deal with. Many user agents expect addresses of the form *user@dns-address*.

Most e-mail systems support mailing lists, so that a user can send the same message to a list of people with a single command.

## Reading E-mail

When a user agent is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen. Then it may announce the number of messages in the mailbox or display a one- line summary of each one and wait for a command. After starting up the user agent, the user asks for a summary of his e-mail. A display appears on the screen. Each line refers to one message.

Each line of the display contains several fields extracted from the envelope or header of the corresponding message. In a simple e-mail system, the choice of fields displayed is built into the

program. In a more sophisticated system, the user can specify which fields are to be displayed by providing a **user profile**, a file describing the display format. After the headers have been displayed, the user can perform any of several actions, such as displaying a message, deleting a message, and so on.

# Message Formats

## RFC 822

Messages consist of a primitive envelope, some number of header fields, a blank line, and then the message body. Each header field consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value.

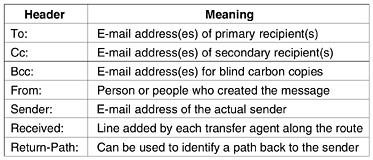


Figure: RFC 822 header fields related to message transport.

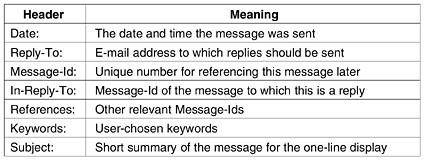


Figure: Some fields used in the RFC 822 message header.

## MIME—The Multipurpose Internet Mail Extensions

The basic idea of MIME is to continue to use the RFC 822 format, but to add structure to the message body and define encoding rules for non-ASCII messages. By not deviating from RFC 822, MIME messages can be sent using the existing mail programs and protocols. All that has to be changed are the sending and receiving programs, which users can do for themselves.

MIME defines five new message headers.

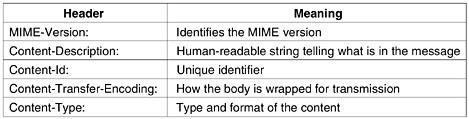
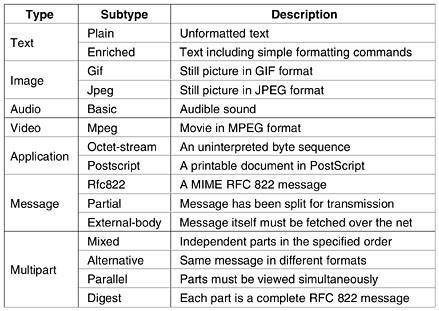


Figure: RFC 822 headers added by MIME.



# Message Transfer

Figure: The MIME types and subtypes

The message transfer system is concerned with relaying messages from the originator to the recipient. The simplest way to do this is to establish a transport connection from the source machine to the destination machine and then just transfer the message.

## SMTP—The Simple Mail Transfer Protocol

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on that port (25). After successfully establishing the TCP connection the client process sends the mail instantly.

The SMTP model is of two type:

* + End-to- end method
  + Store-and- forward method

The end to end model is used to communicate between different organizations whereas the store and forward method is used within an organization. A SMTP client who wants to send the mail will contact the destination’s host SMTP directly in order to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver’s SMTP.

The client SMTP is the one which initiates the session let us call it as client- SMTP and the server SMTP is the one which responds to the session request and let us call it as receiver-SMTP. The client- SMTP will start the session and the receiver-SMTP will respond to the request.

# Final Delivery

## POP3

Post Office Protocol version 3 (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline. Note, that when you use POP3 to connect to your email account, messages are downloaded locally and removed from the email server. This means that if you access your account from multiple locations, that may not be the best option for you. On the other hand, if you use POP3, your messages are stored on your local computer, which reduces the space your email account uses on your web server.

## IMAP

The Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client. IMAP and POP3 are the two most commonly used Internet mail protocols for retrieving emails. Both protocols are supported by all modern email clients and web servers.

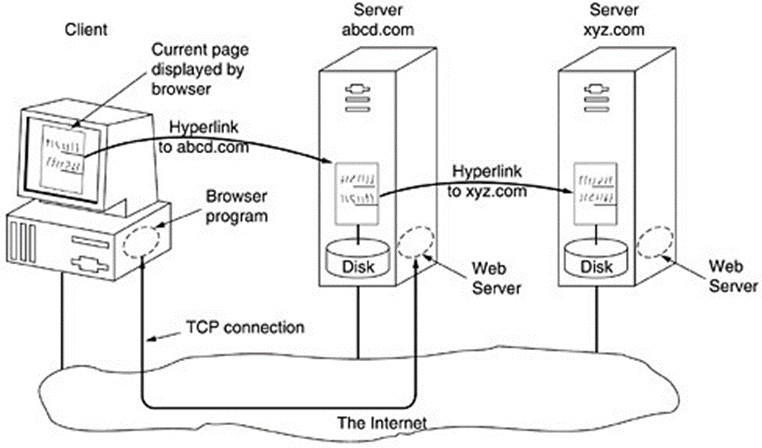
While the POP3 protocol assumes that your email is being accessed only from one application, IMAP allows simultaneous access by multiple clients. This is why IMAP is more suitable for you if you're going to access your email from different locations or if your messages are managed by multiple users.

# World Wide Web

The Web, or World Wide Web (W3), is basically a system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called HTML (HyperText Markup Language) that supports links to other documents, as well as graphics, audio, and video files.

This means you can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web.

# Architectural Overview



## The Client Side

A browser is a program that can display a Web page and catch mouse clicks to items on the displayed page. When an item is selected, the browser follows the hyperlink and fetches the page selected. Therefore, the embedded hyperlink needs a way to name any other page on the Web. Pages are named using URLs (Uniform Resource Locators).

## The Server Side

The steps that the server performs are:

* 1. Accept a TCP connection from a client (a browser).
  2. Get the name of the file requested.
  3. Get the file (from disk).
  4. Return the file to the client.
  5. Release the TCP connection.

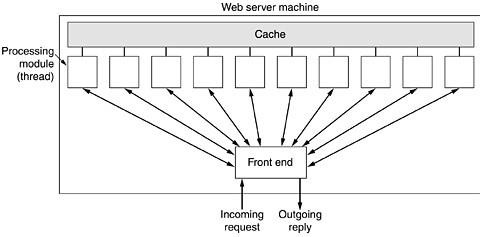


Figure: A multithreaded Web server with a front end and processing modules.

The front end passes each incoming request to the first available module, which then carries it out using some subset of the following steps, depending on which ones are needed for that request.

1. Resolve the name of the Web page requested.
2. Authenticate the client.
3. Perform access control on the client.
4. Perform access control on the Web page.
5. Check the cache.
6. Fetch the requested page from disk.
7. Determine the MIME type to include in the response.
8. Take care of miscellaneous odds and ends.
9. Return the reply to the client.
10. Make an entry in the server log.

## Uniform Resource Locator

URL stands for Uniform Resource Locator. A URL is a formatted text string used by Web browsers, email clients and other software to identify a network resource on the Internet. Network resources are files that can be plain Web pages, other text documents, graphics, or programs.

URL strings consist of three parts (substrings):

* protocol designation
* host name or address
* file or resource location

These substrings are separated by special characters as follows:

protocol :// host / location

The protocol specifies how information from the link is transferred. The protocol used for web resources is HyperText Transfer Protocol (HTTP). Other protocols compatible with most web browsers include HTTPS, FTP, telnet, newsgroups, and Gopher. The protocol is followed by a colon, two slashes, and then the domain name. The domain name is the computer on which the resource is located. Links to particular files or subdirectories may be further specified after the domain name. Directory names are separated by single forward slashes.

# Static Web Documents

Web pages are static, that is, are just files sitting on some server waiting to be retrieved

## HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

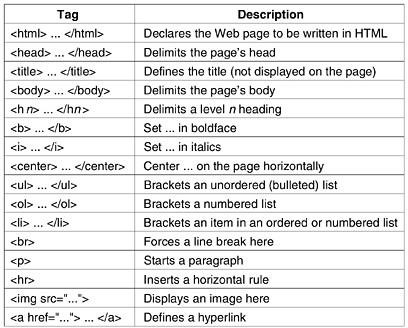


Figure: A selection of common HTML tags. Some can have additional parameters.

## Forms

Forms contain boxes or buttons that allow users to fill in information or make choices and then send the information back to the page's owner. They use the <input> tag for this purpose. It has a variety of parameters for determining the size, nature, and usage of the box displayed. The most common forms are blank fields for accepting user text, boxes that can be checked, active maps, and submit buttons

## XML and XSL

XML stands for eXtensible Markup Language. It is a markup language much like HTML. It was designed to store and transport data and was designed to be self-descriptive. It is a W3C Recommendation

XSL stands for Extensible Styles Language and is a very powerful language for applying styles to XML documents. XSL has two parts —

* Formatting language: allows you to apply styles similar to what CSS does. Browser support for the XSL formatting language is limited at this stage.
* Transformation language. It is known as XSLT (XSL Transformations). XSLT allows you to transform your XML document into another form.

## XHTML—The eXtended HyperText Markup Language

XHTML 1.0 is HTML 4.0 redefined to meet the XML standard. XHTML has been replaced in modern web design with HTML5 and the changes that have come since. You are unlikely to find any newer sites using XHTML, but if you are working on a much older site, you may still encounter XHTML out there in the wild.

There aren't a lot of major differences between HTML and XHTML, but here is what you will notice:

* XHTML is written in lower case. While HTML tags can be written in UPPER case, MiXeD case, or lower case, to be correct, XHTML tags must be all lower case. (Note - many web professionals write HTML in all lowercase, even though it is not technical required).
* All XHTML elements must have an end tag. Elements with only one tag, such as and need a closing slash (/) at the end of the tag: <hr / >

<img / >

# Dynamic Web Documents

## Server Side Dynamic Web Page Generation

* Dynamic web pages involve the execution of a program on the web server.
* The program on the server usually generates an HTML document which is sent to the client to be displayed in the web browser.
* Dynamic web pages are often associated with web forms that provide input data to the programs executed on the server.

Methods for creating Dynamic Documents

* Common Gateway Interface (CGI)
* Active Server Pages (ASP)
* Java Servlets Common Gateway Interface
* CGI is a standard defining how a program will interact with a server to generate dynamic documents.
* Can use any programming language.
* Can generate any type of output, although HTML is most common.

Passing Data to the Server

* Forms are web pages that send the contents of fields to an active document •Each field has a name defined by the author.
* When you press the “submit”button, an HTTP command is sent to the server requesting execution of a program.
* The form data is sent with the HTTP command.

Different Methods

* GET –Data is passed to the server application as an environmental variable.
* POST –The application can read the form data as input from standard input.

Active Server Pages

* Active Server Pages is a Microsoft developed technology for sending dynamic Web content to the client.
* An ASP file (extension .asp) is a serverside text file processed in response to a client browser request.

ASP Execution

* When a server receives an HTTP request from a client, it directs it to be processed by the appropriate Active Server Page.
* VBScript is the de facto language for ASP scripting.
* ASP often involves interacting with a database. •The server then returns its result to the client.
* This is normally in the form of a HTML document –but other formats (e.g., images) are possible.

## Client-Side Dynamic Web Page Generation

To respond to mouse movements or interact with users directly, scripts are embedded in HTML pages that are executed on the client machine rather than the server machine. Starting with HTML 4.0, such scripts are permitted using the tag <script>. The most popular scripting language for the client side is JavaScript.

JavaScript is a scripting language, very loosely inspired by some ideas from the Java programming language. It is definitely not Java. Like other scripting languages, it is a very high level language. For example, in a single line of JavaScript it is possible to pop up a dialog box, wait for text input, and store the resulting string in a variable. High-level features like this make JavaScript ideal for designing interactive Web pages.

JavaScript is not the only way to make Web pages highly interactive. Another popular method is through the use of applets. These are small Java programs that have been compiled into machine instructions for a virtual computer called the JVM (Java Virtual Machine). Applets can be embedded in HTML pages (between <applet> and </applet>) and interpreted by JVMcapable browsers.

# HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers. Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification

specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

## Basic Features

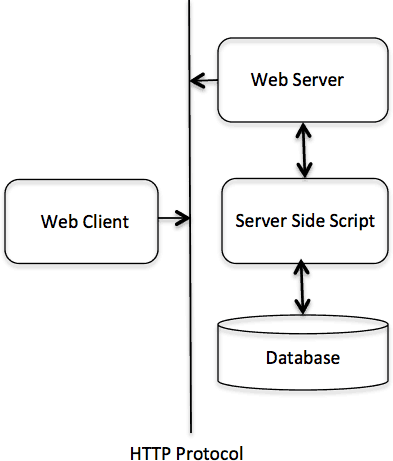
There are three basic features that make HTTP a simple but powerful protocol:

* **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response. The server processes the request and re-establishes the connection with the client to send a response back.
* **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
* **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges.

## Basic Architecture

The following shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

## CLIENT

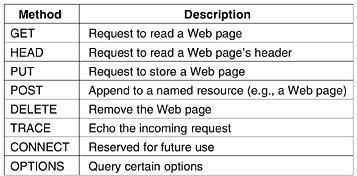
The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

## SERVER

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

# Wireless Web

***Figure: The built-in HTTP request methods.***



## WAP—The Wireless Application Protocol

WAP is essentially a protocol stack for accessing the Web, but optimized for low-bandwidth connections using wireless devices having a slow CPU, little memory, and a small screen. These requirements are obviously different from those of the standard desktop PC scenario, which leads to some protocol differences.

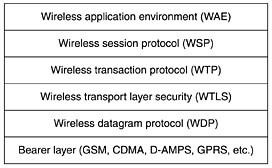


Figure: The WAP protocol stack.

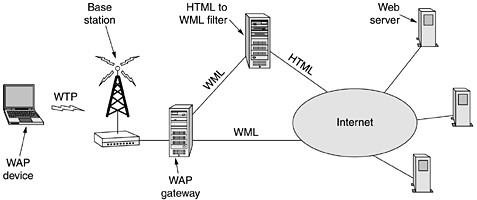


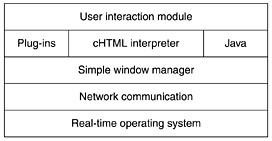
Figure: The WAP architecture.

## I-Mode

The i-mode system has three major components: a new transmission system, a new handset, and a new language for Web page design. The transmission system consists of two separate networks: the existing circuit-switched mobile phone network (somewhat comparable to DAMPS), and a new packet- switched network constructed specifically for i-mode service. Voice mode uses the circuit switched

network and is billed per minute of connection time. I-mode uses the packet-switched network and is always on (like ADSL or cable), so there is no billing for connect time. Instead, there is a charge for each packet sent. It is not currently possible to use both networks at once.

Figure: Structure of the i-mode software.



## Second-Generation Wireless Web

WAP 2.0 has new features. The most significant ones are:

* 1. Push model as well as pull model.
  2. Support for integrating telephony into applications.
  3. Multimedia messaging.
  4. Inclusion of 264 pictograms.
  5. Interface to a storage device.
  6. Support for plug-ins in the browser.

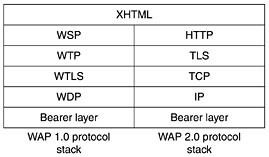


Figure: WAP 2.0 supports two protocol stacks.

# Multimedia

1. **Digital Audio**

Digital audio is audio, or simply sound, signal that has been recorded as or converted into digital form, where the sound wave of the audio signal is encoded as numerical samples in continuous sequence. In a digital audio system, sound of an analog electrical signal is converted with an analog-to-digital converter (ADC) into a digital signal, typically using pulse-code modulation. This digital signal can then be recorded, edited, modified, and copied using digital audio workstation computers, audio playback machines and other digital tools. A digital-to-analog converter (DAC) performs the reverse process, converting a digital signal back into an analog signal, through an audio power amplifier and sending it to a loudspeaker.

Digital audio systems may include compression, storage, processing and transmission components. Conversion to a digital format allows convenient manipulation, storage, transmission and retrieval of

an audio signal. Unlike analog audio, in which making copies of a recording results in generation loss, a degradation of the signal quality, when using digital audio, an infinite number of copies can be made without any degradation of signal quality.

# Streaming Audio

The Internet is full of music Web sites, many of which list song titles that users can click on to play the songs. The most straightforward way to make the music play is illustrated in figure

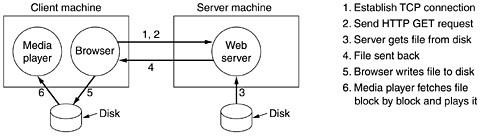


Figure: A straightforward way to implement clickable music on a Web page.

The process starts when the user clicks on a song. Then the browser goes into action.

* + Step 1 is for it to establish a TCP connection to the Web server to which the song is hyperlinked.
  + Step 2 is to send over a *GET* request in HTTP to request the song.
  + Next (steps 3 and 4), the server fetches the song (which is just a file in MP3 or some other format) from the disk and sends it back to the browser. If the file is larger than the server's memory, it may fetch and send the music a block at a time.

The media server uses RTSP (Real Time Streaming Protocol).

## VOIP

Voice over Internet Protocol (VoIP) is a technology for communicating using "Internet protocol" instead of traditional analog systems. VoIP converts the voice signal from your telephone into a digital signal that can travel over the Internet. If you are calling a regular telephone number, the signal is then converted back at the other end. VoIP calls can be made from a computer, a special VoIP phone, a traditional phone with or without an adapter, or using a wireless phone, depending on the type of VoIP service you subscribe to.

VoIP protocols include Session Initiation Protocol (SIP), H.323

**H.323 protocol**

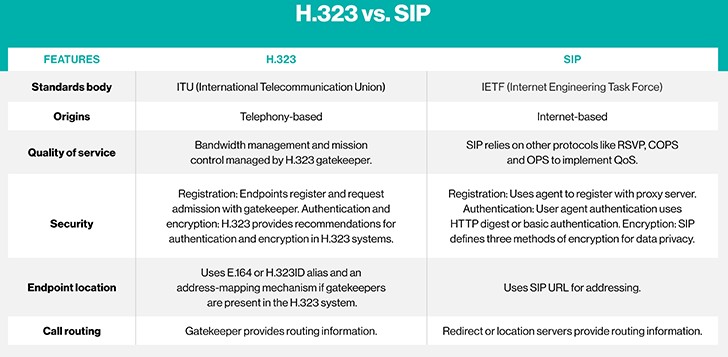
H.323 is a binary-based standard developed by the International Telecommunication Union to support rich-media communications over IP networks that was initially focused on video conferencing, but now includes audio and video conferencing. H.323 is a well-defined and well-structured protocol, with specific definitions for establishing sessions, which can be loosely compared to calls, services and session components. Because of its rigid services definition that comes out of the telecom-based standards body, all H.323 implementations are generally interoperable.

While H.323's rigid definition is an advantage in terms of interoperability, that rigidity can be its greatest challenge, because vendors are limited in their ability to layer additional features or services not supported by the protocol.

**Session Initiation Protocol**

SIP was developed by the Internet Engineering Task Force. It was designed to set up a session between two points and to be a flexible component of the internet architecture. Unlike H.323, SIP's initial focus was voice communications, rather than video, but it was expanded to include video, application sharing, presence, instant messaging and other common communications applications.

SIP is an ASCII text-based standard leveraging much of the existing design of HTTP. SIP's text format, however, can result in large messages that aren't as suitable for networks that may have bandwidth, delay and processing issues. SIP is highly extensible, which allows developers to expand or add to its capabilities, and it supports rich-media communications, as well as data transfer. SIP uses the Session Description Protocol (SDP) to define the characteristics of a session, which enables the use of encryption, transport protocol, the selection of voice and video codecs, and compression.



# Video on Demand

Video on demand (VoD) is a system that allows users to select and watch video content of their choice on their TVs or computers. Video on demand is one of the dynamic features offered by Internet Protocol TV. VoD provides users with a menu of available videos from which to choose. The video data is transmitted via Real-Time Streaming Protocol.

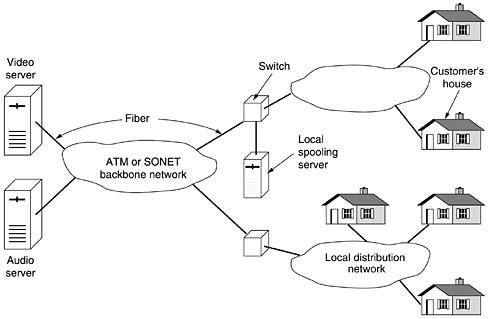


Figure: Overview of a video-on-demand system.

## Video Servers

To have (near) video on demand, we need video servers capable of storing and outputting a large number of movies simultaneously.

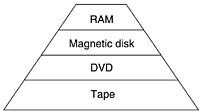


Figure: A video server storage hierarchy.

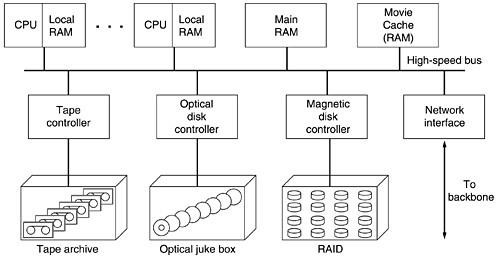


Figure: The hardware architecture of a typical video server.

## The Distribution Network

The distribution network is the set of switches and lines between the source and destination. It consists of a backbone, connected to a local distribution network. Usually, the backbone is switched and the local distribution network is not. The main requirement imposed on the backbone is high bandwidth.