HTTP

* Most famous for two-way conversation between web browsers and web servers.
* The world’s web browsers, servers, and related web applications all talk to each other through HTTP, the Hypertext Transfer Protocol.
* Billions of JPEG images, HTML pages, text files, MPEG movies, WAV audio files, Java applets, and more cruise through the Internet each and every day. HTTP moves the bulk of this information quickly, conveniently, and reliably from web servers all around the world to web browsers on people’s desktops.

Web Clients and Servers

* Web content lives on web servers. Web servers speak the HTTP protocol, so they are often called HTTP servers. These HTTP servers store the Internet’s data and provide the data when it is requested by HTTP clients. The clients send HTTP requests to servers, and servers return the requested data in HTTP responses
* HTTP response, along with the type of the object, the length of the object, and other information.
* HTTP request - “Get me the document called /index.html.”
* HTTP response - “Okay, here it is, it’s in HTML format and is 3,150 characters long.”

Resources

* Web servers host web resources. A web resource is the source of web content.
* Text files, HTML files, Microsoft Word files, Adobe Acrobat files, JPEG image files, AVI movie files, software programs that generate content on demand.
* A resource is any kind of content source.

Media Types

* MIME (Multipurpose Internet Mail Extensions)
* A MIME type is a textual label, represented as a primary object type and a specific subtype, separated by a slash.
* Content-type: image(MIME type)/jpeg
* Content-length: 12984
* An HTML-formatted text document would be labeled with type text/html.
* A GIF-format image would be image/gif.

URI

* Each web server resource has a name, so clients can point out what resources they are interested in. The server resource name is called a uniform resource identifier, or URI. URIs are like the postal addresses of the Internet, uniquely identifying and locating information resources around the world.
* <http://www.joes-hardware.com/specials/saw-blade.gif>
* URI specifies the HTTP protocol to access the saw-blade GIF resource on Joe’s store’s server. Given the URI, HTTP can retrieve the object.

URLs

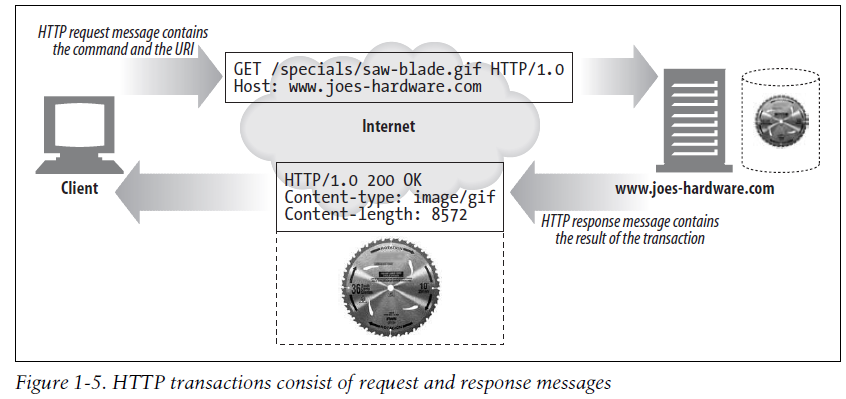
* The uniform resource locator (URL) is the most common form of resource identifier. URLs describe the specific location of a resource on a particular server. They tell you exactly how to fetch a resource from a precise, fixed location.
* Uniform resource locators (URLs) are the standardized names for the Internet’s resources.
* URLs point to pieces of electronic information, telling you where they are located and how to interact with them.

Most URLs follow a standardized format of three main parts:

* The first part of the URL is called the scheme, and it describes the protocol used to access the resource. This is usually the HTTP protocol (http://).
* The second part gives the server Internet address (e.g., www.joes-hardware.com).
* The rest names a resource on the web server (e.g., /specials/saw-blade.gif ).
* Today, almost every URI is a URL.

Transactions

* An HTTP transaction consists of a request command (sent from client to server), and a response result (sent from the server back to the client). This communication happens with formatted blocks of data called HTTP messages.



Methods

* HTTP supports several different request commands, called HTTP methods. Every HTTP request message has a method. The method tells the server what action to perform (fetch a web page, run a gateway program, delete a file, etc.).

Some common HTTP methods

HTTP method Description

GET - Send named resource from the server to the client.

PUT - Store data from client into a named server resource.

DELETE -Delete the named resource from a server.

POST -Send client data into a server gateway application.

HEAD -Send just the HTTP headers from the response for the named resource.

Status Codes

* Every HTTP response message comes back with a status code. The status code is a three-digit numeric code that tells the client if the request succeeded, or if other actions are required
* 2xx Success
* This class of status codes indicates the action requested by the client was received, understood, accepted and processed successfully.
* 3xx Redirection
* This class of status code indicates the client must take additional action to complete the request. Many of these status codes are used in URL redirection.
* 4xx Client Error
* 5xx Server Error
* Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request.

Number Meaning

200 OK

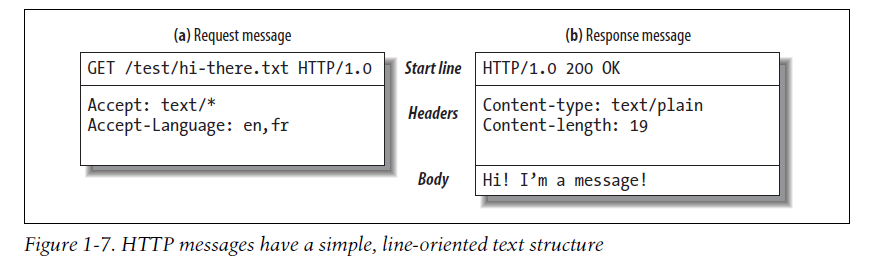
301-303 page has moved (permanently or temporarily)

403 you are forbidden to access this page

404 page not found

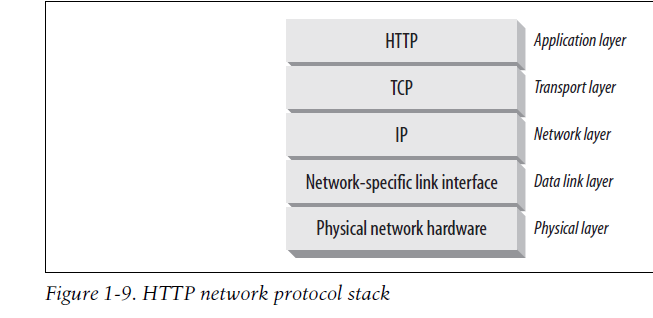
500 internal server error

Messages



* HTTP messages sent from web clients to web servers are called request messages.
* Messages from servers to clients are called response messages. There are no other kinds of HTTP messages.

TCP (Transmission Control Protocol)



Connections, IP Addresses, and Port Numbers

* Before an HTTP client can send a message to a server, it needs to establish a TCP/IP connection between the client and server using Internet protocol (IP) addresses and port numbers.
* Setting up a TCP connection is sort of like calling someone at a corporate office. First, you dial the company’s phone number. This gets you to the right organization. Then, you dial the specific extension of the person you’re trying to reach. In TCP, you need the IP address of the server computer and the TCP port number associated with the specific software program running on the server.

Host Name

* The hostname is just a human-friendly alias for an IP address. Hostnames can easily be converted into IP addresses through a facility called the Domain Name Service (DNS).

Here are the steps:

(a) The browser extracts the server’s hostname from the URL.

(b) The browser converts the server’s hostname into the server’s IP address.

(c) The browser extracts the port number (if any) from the URL.

(d) The browser establishes a TCP connection with the web server.

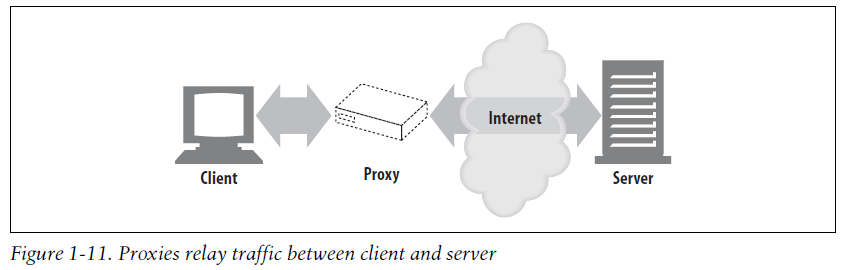
(e) The browser sends an HTTP request message to the server.

(f) The server sends an HTTP response back to the browser.

(g) The connection is closed, and the browser displays the document.

Proxies

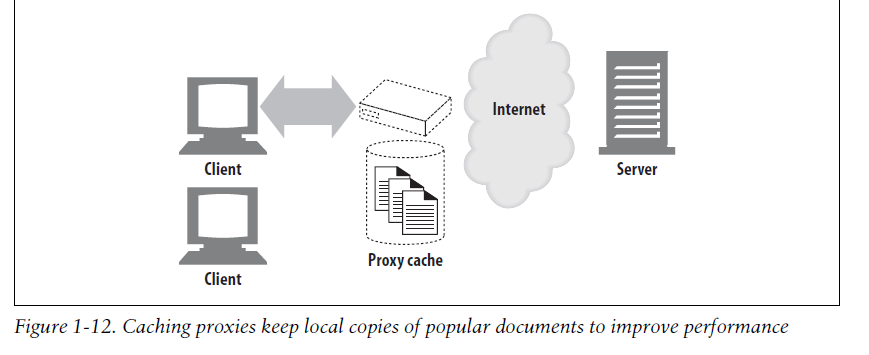
* HTTP intermediaries that sit between clients and servers
* Receiving all of the client’s HTTP requests and relaying the requests to the server (perhaps after modifying the requests)
* These applications act as a proxy for the user, accessing the server on the user’s behalf.



* Proxies are often used for security, acting as trusted intermediaries through which all web traffic flows. Proxies can also filter requests and responses; for example, to detect application viruses in corporate downloads or to filter adult content away from elementary-school students.

Caches

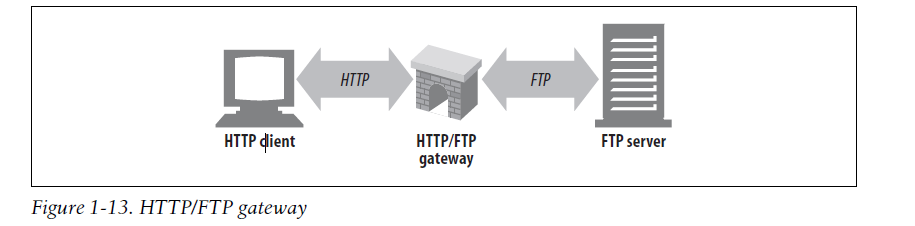
* HTTP storehouses that keep copies of popular web pages close to clients
* A web cache or caching proxy is a special type of HTTP proxy server that keeps copies of popular documents that pass through the proxy. The next client requesting the same document can be served from the cache’s personal copy



* A client may be able to download a document much more quickly from a nearby cache than from a distant web server

Gateways

* Special web servers that connect to other applications
* Gateways are special servers that act as intermediaries for other servers.
* They are often used to convert HTTP traffic to another protocol.
* For example, an HTTP/FTP gateway receives requests for FTP URIs via HTTP requests but fetches the documents using the FTP protocol (see Figure 1-13). The resulting document is packed into an HTTP message and sent to the client



Tunnels

* Special proxies that blindly forward HTTP communications

Agents

* Semi-intelligent web clients that make automated HTTP requests

Port

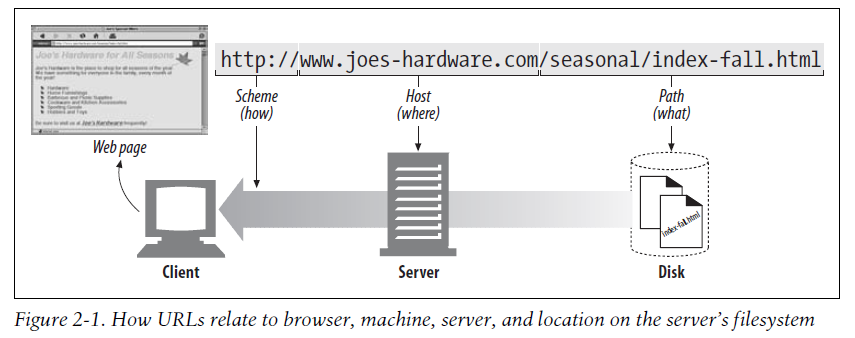
* A port (noun) is a "logical connection place" and specifically, using the Internet's protocol, TCP/IP, the way a client program specifies a particular server program on a computer in a network.
* When referring to a physical device, a hardware port or peripheral port is a hole or connection found on the front or back of a computer. Ports allow computers to access external devices such as printers.
* When referring to a network or to the Internet, a software or network port is a location where information is sent. For example, port 80 is the http network port.

Some Common port numbers:

* echo 7
* daytime 13
* qotd 17 (Quote of the Day)
* ftp 21
* SSH 22
* telnet 23
* smtp 25 (Simple Mail Transfer, meaning e-mail)
* time 37
* nameserver 53
* nicname 43 (Who Is)
* gopher 70
* finger 79
* WWW 80

Process of Fetching Information from Server

* Example - <http://www.joes-hardware.com/seasonal/index-fall.html>
* The first part of the URL (http) is the URL scheme. The scheme tells a web client how to access the resource. In this case, the URL says to use the HTTP protocol.
* The second part of the URL (www.joes-hardware.com) is the server location. This tells the web client where the resource is hosted.
* The third part of the URL (/seasonal/index-fall.html) is the resource path. The path tells what particular local resource on the server is being requested.



FTP

* File Transfer Protocol
* Way of transferring files to and from servers.
* Used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.
* A format and set of rules for transferring files from a host to a remote computer.

Websites and Web Applications

* Web sites are more about displaying static (unchanging) content and web applications require a "smart" server running code (that you're going to learn to write) to produce dynamic content.