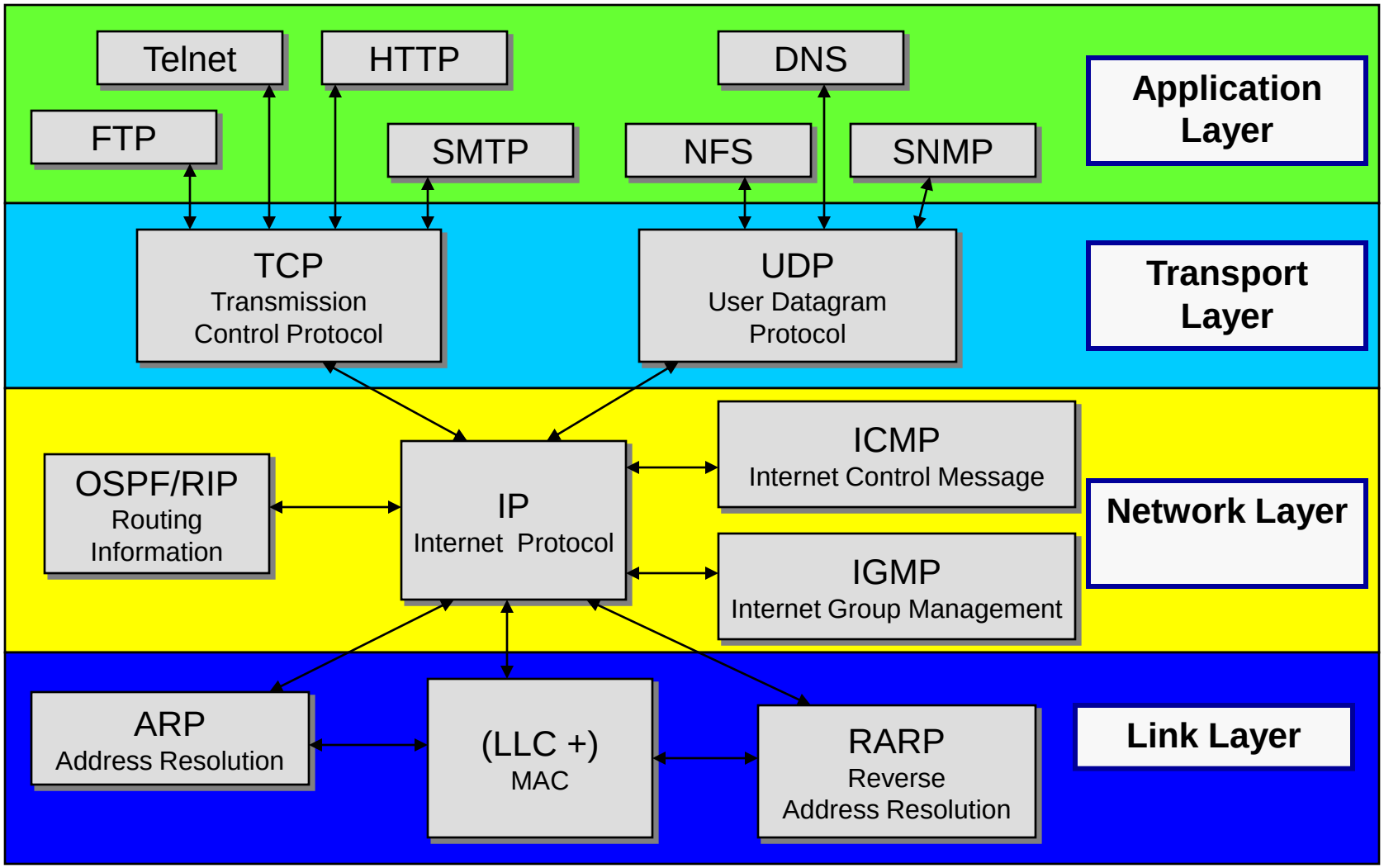
CGS 3175: Internet Applications
Fall 2009

Introduction to the Internet – Part 2

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cgs3175/fall2009

School of Electrical Engineering and Computer Science
University of Central Florida

Internet Reference Model



UDP

- UDP (User Datagram Protocol) is an alternative protocol to TCP that also builds on IP (extends the functionality of IP).
  - The main feature that UDP adds to IP is the port concept that is the same as we saw with TCP.
- Unlike TCP, UDP does not provide two way-connection or guaranteed delivery.
- Its advantage over TCP is speed (for simple tasks).
  - For example, if all you want to do is send a short message to another computer, you're expecting a single short response message, and you can handle resending if you don't receive the response within a reasonable amount of time, the UDP is a good alternative to TCP.

UDP

- One Internet application that is often run using UDP rather than TCP is the Domain Name Service (DNS).
- While every device on the Internet has an IP address (such as 132.170.7.155), humans generally find it easier to refer to machines by names, such as www.cs.ucf.edu.
- DNS provides a mechanism for mapping back and forth between IP addresses and host names.
- Basically, there are a number of DNS servers on the Internet, each listening through UDP software to a port (port 53 if the server is following the current IANA port assignment standard).

UDP

- When a computer on the Internet needs DNS services (for example to convert a host name such as www.cs.ucf.edu) to its corresponding IP address, it uses the DUP software running on its system to send a UDP message to one of these DNS servers, requesting the IP address.
- If all goes well, this DNS server will then send back a UDP message containing the IP address.
  - Recall that it took three messages just to get the TCP connection set up, so the UDP approach is much for efficient for sporadic DNS queries.
  - UDP is sometimes referred to as a lightweight communication protocol and TCP as a heavyweight protocol. In general, in computer science, the terms lightweight and heavyweight are used to describe alternative software solutions to some problem, with the lightweight solution having less functionality but also less overhead.

DNS

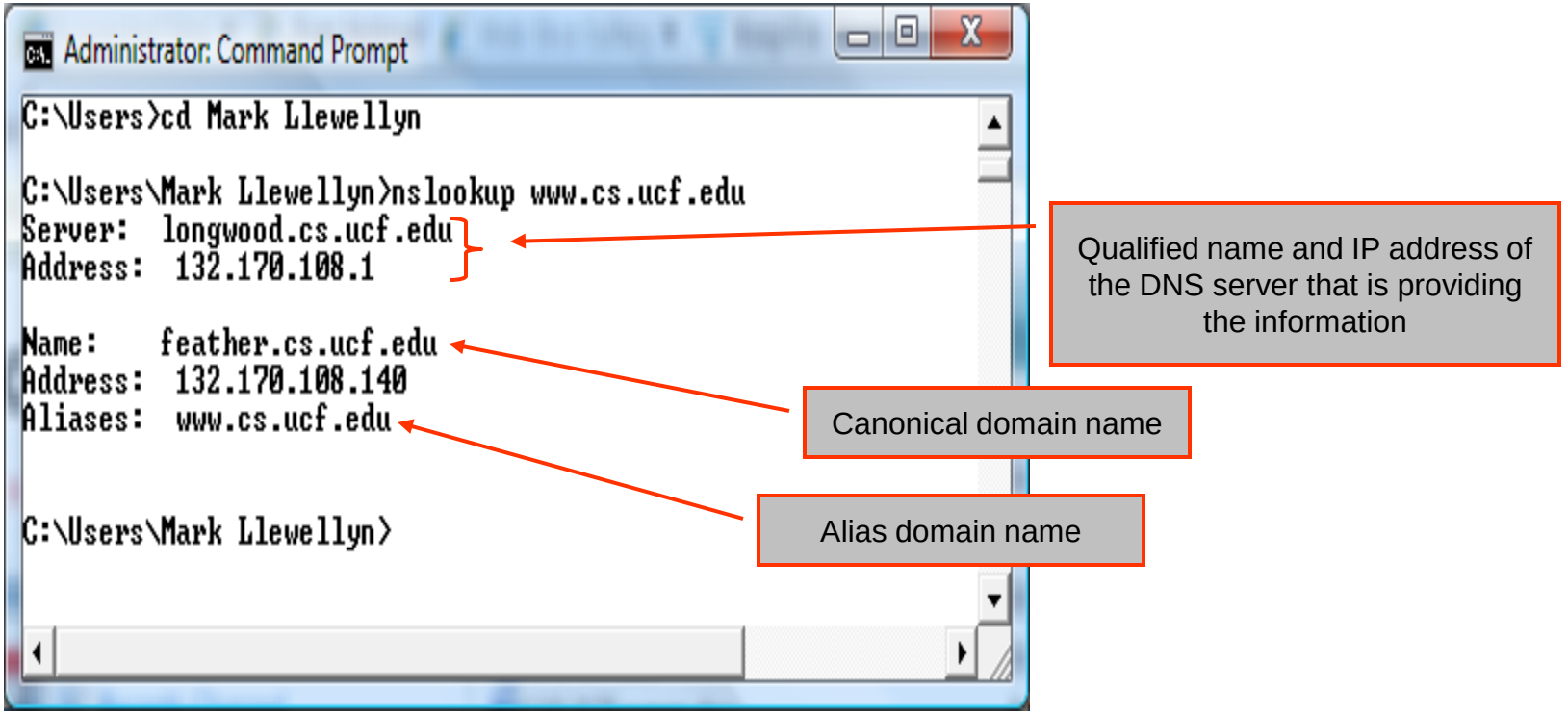
- Internet host names consist of a sequence of labels separated by dots.
- The final label (rightmost) in a host name is a top-level domain.
- There are two standard types of top-level domain:
  1. Generic: such as .com, .edu, .org, and .biz.
  2. Country-code: such as .de, .il, and .mx.
- The top-level domain names are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN), a private nonprofit organization formed to take over technical Internet functions that were originally funded by the U.S. government.

DNS

- Each top-level domain is divided into subdomains (second-level domains), which may in turn be further divided and so on.
- The assignment of second-level domains within each top-level domain is performed (for a fee) by a registry operator selected by ICANN.
- The owner of a second-level domain can then further divided that domain into subdomains, and so on. Ultimately, the subdomains of a domain are individual computers.
- Such a subdomain, consisting of a local host name followed by a domain name (typically consisting of at least two labels) is sometimes called a fully qualified domain name.
  - For example, `www.cs.ucf.edu` is a fully qualified domain name for a host with local name `www` that belongs to the `cs` third-level domain of the `ucf` second-level domain of the `edu` top-level domain.

DNS

- Some user-level tools are available that allow you to query the Inernet DNS.
- For example, on most machines the **`nslookup`** command can be typed at a command prompt in order to find the IP address of a fully qualified domain name or vice versa.



Higher-Level Protocols

- The following analogy may help to relate the computer networking concepts we've just covered with something more familiar: the telephone network.
- The Internet is like the physical telephone network: it provides the basic communication infrastructure.
- UDP is like calling a number and leaving a message rather than actually speaking with the intended recipient.
- DNS is the Internet version of directory assistance, associating names with numbers.
- TCP is roughly equivalent to placing a phone call and having the other party answer: you now have a connection and are able to communicate back and forth.

Higher-Level Protocols

- In the cases of both TCP and a phone call, different protocols can be used to communicate once a connection has been established.
  - For example, when making a telephone call, the parties must agree on the language(s) that will be used to communicate. Beyond that, there are also conventions (protocols) that are followed to decide which party will speak first, how the parties will take turns speaking, and so on. Furthermore, different protocols may be used in different contexts: I answer the phone at home differently than I do at work, for example.
- Similarly, a variety of higher-level protocols are used to communicate once a TCP connection has been established (see figure on page 18 in the first set of notes). SMTP and FTP are two examples we mentioned earlier of widely used higher-level protocols used to communicate over TCP connections.
- The protocol that will be used to communicate over a TCP connection is normally determined by the port number used to establish the connection.

Higher-Level Protocols

- The primary TCP-based protocol used for communication between web servers and browsers is called the Hypertext Transport Protocol (HTTP).
- In some sense, just as IP is a key component in the definition of the Internet, HTTP is a key component in the definition of the World Wide Web.
- Before looking into HTTP – let's consider what the Web is, and how HTTP figures in its definition.

The World Wide Web

- Public sharing of information has been a part of the Internet since its early days.
  - For example, the Usenet newsgroup service began in 1979 and provided a means of "posting" information that could be read by users on other systems with the appropriate software.
  - If you would like to check out Usenet you can get to it through the Google Groups Usenet discussion forum at http://groups.google.com (See next page).

Google Groups - Windows Internet Explorer

http://groups.google.com/

File Edit View Favorites Tools Help

Google http Search Bookmarks Check AutoFill http Sign In

Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid- Download and s...

Google Groups

Page Safety Tools

Web Images Videos Maps News Shopping Gmail more ▼ Help | Sign in

Google groups Search Groups Advanced Groups Search Preferences

What can you do with groups? Take the tour »

Discuss online or over email

Create rich, custom pages

Customize your look and graphics

Create a group in 3 steps

1 Create an account

2 Setup your group

3 Invite people

Create a group...

Explore groups

Find out what people are doing with Google Groups

Search for a group

Arts & Entertainment Groups

Business Groups

Computer Groups

Health Groups

Home Groups

News Groups

People Groups

Recreation Groups

School & University Groups

Sci/Tech Groups

Society & Humanities Groups

Browse group categories...

Internet | Protected Mode: Off 100%

The World Wide Web

- Large files were (and still are) often shared by running an FTP server application that allowed any user to transfer the files from their origin machine to the user's machine.
- The first Internet chat software in widespread use, Internet Relay Chat (IRC), provided both public and private chat facilities.
- As the amount of information publicly available on the Internet grew, the need to locate information also grew.
- Various technologies for supporting information management and search on the Internet were developed.

The World Wide Web

- Some of the more popular information management technologies in the early 1990s were:
  - Gopher information servers, which provided a simple hierarchical view of documents.
  - Wide Area Information System (WAIS) system for indexing and retrieving information.
  - The ARCHIE tool for searching online information archives accessible by FTP.
- The World Wide Web also was developed in the early 1990s and for a while was just one among several Internet information management technologies.
- To understand why the Web supplanted the other technologies, its helpful to understand a bit of the mechanics of the Web and the other Internet information management technologies.

The World Wide Web

- All of these technologies consist of (at least) two types of software: client and server.
- An Internet-connected computer that wishes to provide information to other Internet systems must run server software, and a system that wishes to access the information provided by servers must run client software (for the Web, the client software is normally a web browser).
- The server and client applications communicate over the Internet by following a communication protocol built on top of TCP/IP.

The World Wide Web

- As we just mentioned (page 10), the protocol used by the Web is the Hypertext Transport Protocol (HTTP).
- As we will see, HTTP is a rather generic protocol that for the most part supports a client requesting a document from a server and the server returning the requested document.
  - It is this generic nature of HTTP that gives it the advantage of somewhat more flexibility than is present in the protocols used by WAIS and Gopher.
- A bigger advantage for the Web is the type of information communicated. Most web pages are written using the Hypertext Markup Language (HTML), which along with HTTP is a fundamental web technology.

The World Wide Web

- HTML pages can contain the familiar web links (technically called hyperlinks) to other documents on the Web.
  - While certain Gopher pages could also contain links, normal Gopher documents were just plain text.
  - WAIS and ARCHI provided no direct support for links.
- In addition to hyperlinks, modern versions of HTML also provide extensive page layout facilities, including support for inline graphics, which as you can imagine, has added significantly to the commercial appeal of the Web.

The World Wide Web

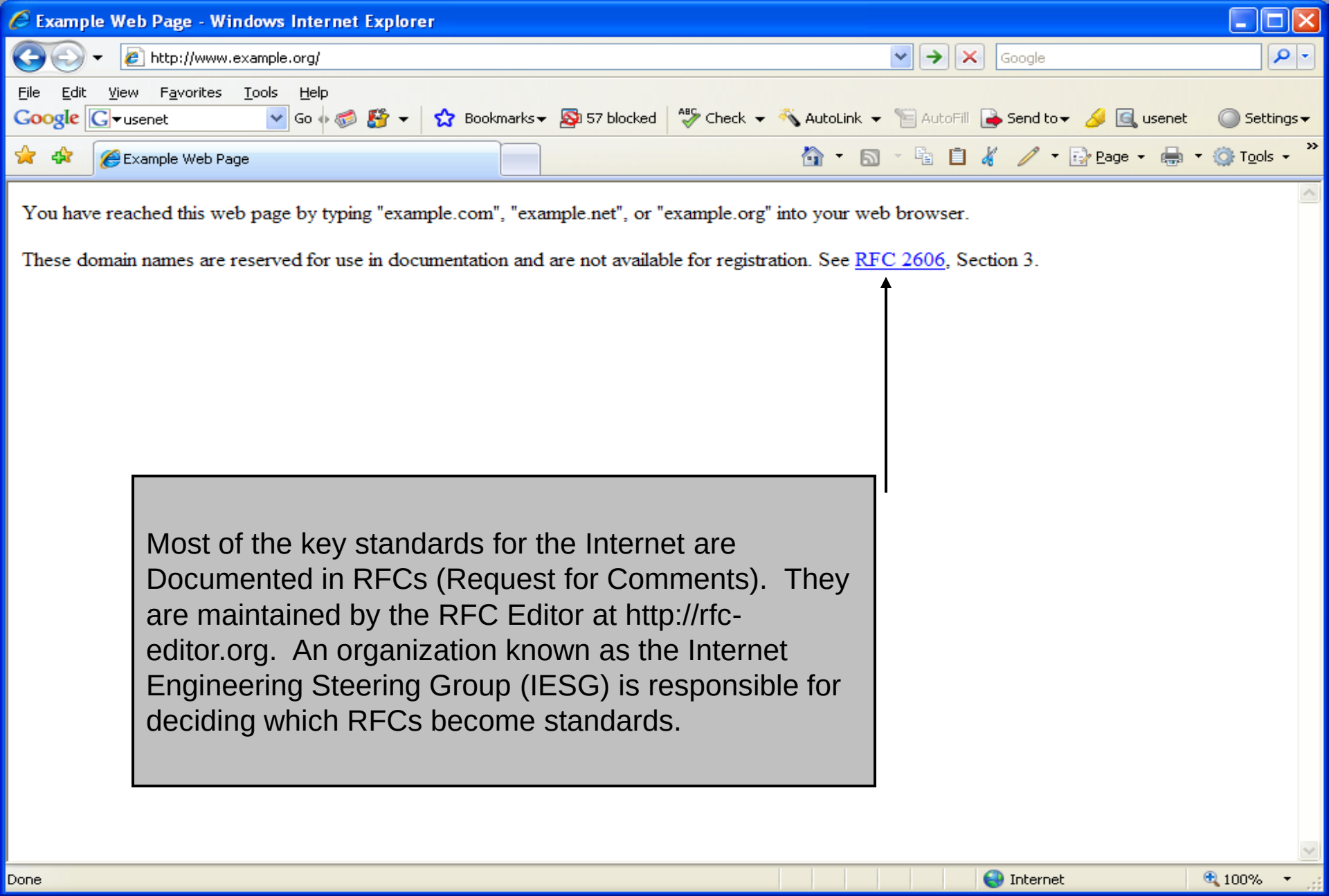
- The World Wide Web, then can be defined in much the same way as the Internet.
- While the Internet can be thought of as the collection of machines that are globally connected via IP, the World Wide Web can be informally defined as the collection of machines (web servers) on the Internet that provide information via HTTP, and in particularly those that provide HTML documents.
- Now, let's look more closely at HTTP.

Hypertext Transport Protocol – HTTP

- HTTP is a form of communication protocol, in particular a detailed specification of how web clients and servers should communicate.
- The basic structure of HTTP communication follows what is known as a request-response model.
- Specifically, the protocol dictates that an HTTP interaction is initiated by a client sending a request message to the server; the server is then expected to generate a response message.
- The format of the request and response messages is dictated by HTTP.
- HTTP does not dictate the network protocol to be used to send these messages, but does expect that the request and response are both send within a TCP-style connection between the client and the server. So most HTTP implementations send these messages using TCP.

Hypertext Transport Protocol – HTTP

- Let's relate this to what happens when you browse the Web.
- The next page shows a browser window in which I typed `http://www.example.org` in the location bar.
- When the Enter key is pressed after typing this address, the browser created a message conforming to the HTTP protocol, used DNS to obtain an IP address for ww.example.org, created a TCP connection with the machine at the IP address obtained, sent the HTTP message over this TCP connection, and received back a message containing the information that is shown in the client area of the browser (the portion of the browser containing the information received from the web server).
-



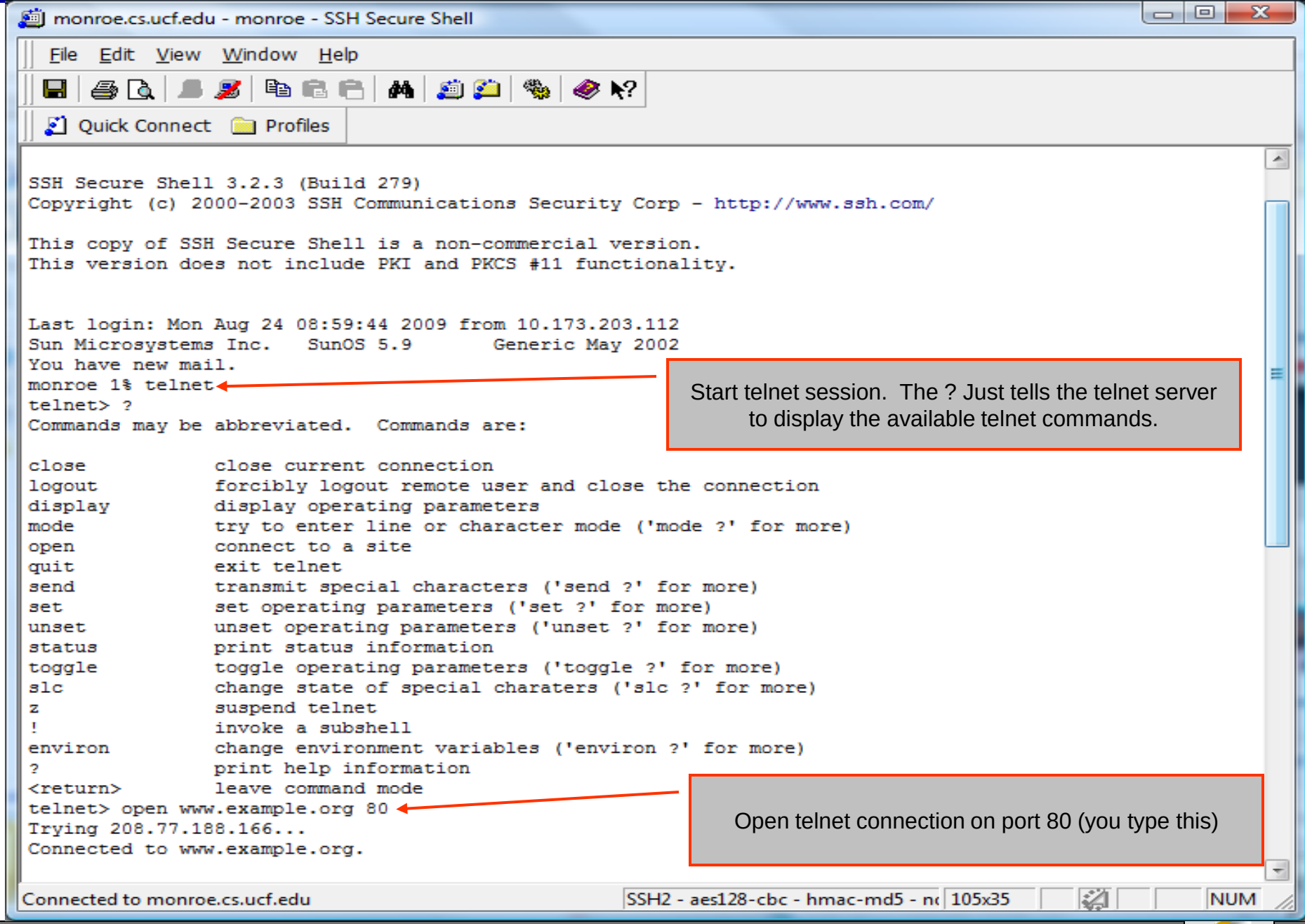
Most of the key standards for the Internet are Documented in RFCs (Request for Comments). They are maintained by the RFC Editor at http://rfc-editor.org. An organization known as the Internet Engineering Steering Group (IESG) is responsible for deciding which RFCs become standards.

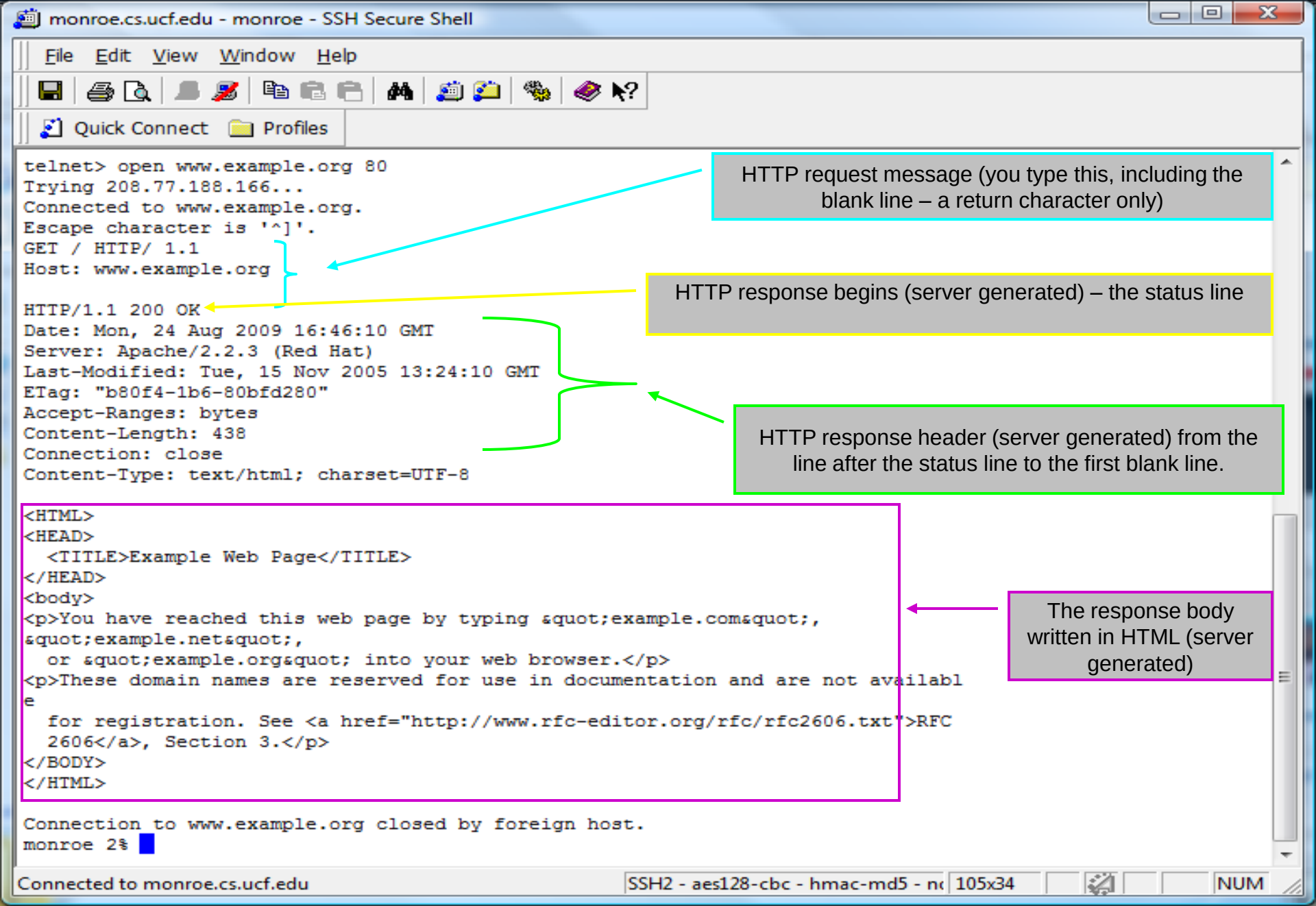
Hypertext Transport Protocol – HTTP

- A nice feature of HTTP is that these request and response messages often consist entirely of plain text in a fairly readable form.
- An HTTP request message consists of a start line followed by a message header and optionally a message body.
  - The start line always consists of printable ASCII characters, and the header normally does as well.
- What's more, the HTTP response (or at least most of it) is often also a stream of printable characters.

Hypertext Transport Protocol – HTTP

- Let's look at HTTP in action, by using Telnet to connect to the same site we used on page 22.
- This can be done on many systems by entering telnet from a command prompt.
- If you can't do this from your own system don't worry about it, just look at the example on the next page and follow what's happening.
- Specifically, we need to telnet to port 80, the IANA standard port for HTTP web servers, type in an HTTP request message corresponding to the Internet address we entered into the browser on page 22, and view the response.
- The request consists of three lines beginning with the GET and ending with a blank line.


monroe.cs.ucf.edu - monroe - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
SSH Secure Shell 3.2.3 (Build 279)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/
This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
Last login: Mon Aug 24 08:59:44 2009 from 10.173.203.112
Sun Microsystems Inc. SunOS 5.9 Generic May 2002
You have new mail.
monroe 1% telnet
telnet> ?
Commands may be abbreviated. Commands are:
close close current connection
logout forcibly logout remote user and close the connection
display display operating parameters
mode try to enter line or character mode ('mode ?' for more)
open connect to a site
quit exit telnet
send transmit special characters ('send ?' for more)
set set operating parameters ('set ?' for more)
unset unset operating parameters ('unset ?' for more)
status print status information
toggle toggle operating parameters ('toggle ?' for more)
slc change state of special charaters ('slc ?' for more)
z suspend telnet
! invoke a subshell
environ change environment variables ('environ ?' for more)
? print help information
<return> leave command mode
telnet> open www.example.org 80
Trying 208.77.188.166...
Connected to www.example.org.
Start telnet session. The ? Just tells the telnet server to display the available telnet commands.
Open telnet connection on port 80 (you type this)
Connected to monroe.cs.ucf.edu
SSH2 - aes128-cbc - hmac-md5 - n
105x35
NUM


monroe.cs.ucf.edu - monroe - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
telnet> open www.example.org 80
Trying 208.77.188.166...
Connected to www.example.org.
Escape character is '^]'.
GET / HTTP/ 1.1
Host: www.example.org
HTTP/1.1 200 OK
Date: Mon, 24 Aug 2009 16:46:10 GMT
Server: Apache/2.2.3 (Red Hat)
Last-Modified: Tue, 15 Nov 2005 13:24:10 GMT
ETag: "b80f4-1b6-80bfd280"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
<HTML>
<HEAD>
<TITLE>Example Web Page</TITLE>
</HEAD>
<body>
You have reached this web page by typing "example.com",
"example.net",
or "example.org" into your web browser.
These domain names are reserved for use in documentation and are not availabl
e
for registration. See RFC
2606, Section 3.
</BODY>
</HTML>
Connection to www.example.org closed by foreign host.
monroe 2%
Connected to monroe.cs.ucf.edu
SSH2 - aes128-cbc - hmac-md5 - n
105x34
NUM
HTTP request message (you type this, including the blank line – a return character only)
HTTP response begins (server generated) – the status line
HTTP response header (server generated) from the line after the status line to the first blank line.
The response body written in HTML (server generated)

HTTP Request Message

- Every HTTP request message has the same basic structure:

  Start line

  Header field(s) (one or more)

  Blank line

  Message body (optional)

- The start line in the previous example was:

  GET / HTTP/ 1.1

HTTP Request Message

- Every start line consists of three parts, with a single space used to separate adjacent parts:

1. Request method
2. Request-URI portion of the web address
3. HTTP version

- We'll look at each of these parts of the start line – in reverse order – in the next few pages, then move on to the header fields and body.

Start Line – HTTP Version

- The initial version of HTTP was referred to as HTTP/0.9, and the first Internet RFC (Request for Comments) described HTTP/1.0.
- In 1997, HTTP/1.1 was formally defined, and is currently an Internet Draft Standard (RFC-2616) [ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt].
- Essentially all operational browsers and servers support HTTP/1.1, including the server that generated the previous example (as indicated by the HTTP version portion of the status line).

Start Line – Request-URI

- The second part of the start line is known as the Request-URI.
- The concatenation of the string http://, the value of the Host header field (www.example.org, in this example), and the Request-URI (/ in this example) forms a string known as a Uniform Resource Identifier. (URI).
- A URI is an identifier that is intended to be associated with a particular resource (such as a web page or graphics image) on the World Wide Web.
- Every URI consists of two parts: the scheme, which appears before the colon (:), and another part that depends on the scheme.
- Web addresses, for the most part, use the `http` scheme (the scheme name in in URIs is case insensitive, but is generally written in lower case letters).

Start Line – Request-URI

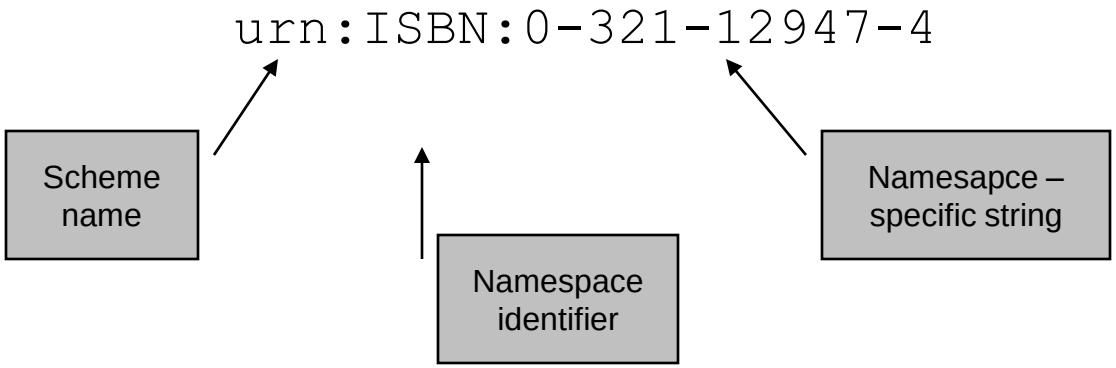
- In this scheme, the URI represents the location of a resource on the Web. A URI of this type is said to be a Uniform Resource Locator (URL).
- Therefore, URIs using the `http` scheme are both URIs and URLs.
- Some other URI schemes that mark the URI as a URL are shown in the table below. A complete list of the currently registered URI schemes along with references to details on each scheme can be found at http://iana.org/assignments/uri-schemes.html.

Scheme Name	Example URL	Type of Resource
ftp	ftp://ftp.example.org/pub/afile.txt	File located on an FTP server
telnet	telnet://host.example.org/	Telnet server
mailto	mailto: someone@example.org	Mailbox
https	https://secure.example.org/sec.txt	Resource on web server supporting encrypted communications
file	file:///C:/temp/localFile.txt	File accessible from machine processing this URI.

Uniform Resource Name (URN)

- In addition to the URL type of URI, there is one other type, called a Uniform Resource Name (URN).
- Although not as common as URLs, URNs are sometimes used in web development (when using tag libraries the tag namespace must be a URN – we'll see this later in the semester).
- A URN is designed to be a unique name for a resource rather than specifying a location at which to find the resource.
  - For example, the textbook for our course has an ISBN (International Standard Book Number) of 0-321-12947-4 associated with it, and this is the only book worldwide with this number. So it makes sense to associate information regarding this book, such as bibliographic data, with its ISBN. In fact, this book has an associated URN, which can be written as follows: `urn:ISBN:0-321-12947-4`

Uniform Resource Name (URN)



- The URI for a URN always consists of three colon-separated parts, as illustrated here. The first part is the scheme name, which is always urn for a URN-type URI.
- The second part is the namespace identifier, which is this example is ISBN. Other currently registered URN namespace identifiers are listed at: http://iana.org/assignments/urn-namespaces.
- The third part is the namespace-specific string. The exact format and meaning of this string varies with the namespace, in this example, it represents the ISBN of a book and has format defined by the IANA.

Start Line – Request Method

- The last (first) part of the start line is known as the request method.
- The standard HTTP methods and a brief description of each are shown in the table on the next page.
- The method part of the start line of an HTTP request must be written entirely in uppercase letters, as shown in the table.
- The primary HTTP method is GET. This is the method used when you type a URL into the Location bar of your browser.
  - It is also the method that is used by default when you click on a link in a document displayed in your browser and when the browser downloads images for display within an HTNL document.
- The POST method is typically used to send information collected from a form displayed within a browser, such as an order-entry form, back to the web server.

Standard HTTP/1.1 Methods

Method	Requests server to…
GET	Return the resource specified by the Request-URI as the body of a response message.
POST	Pass the body of this request message on as data to be processed by the resource specified by the Request-URI.
HEAD	Return the same HTTP header fields that would be returned if a GET method were used, but not return the message body that would be returned to a GET. Provides information about a resource without the communication overhead of transmitting the body of the response.
PUT	Store the body of this message on the server and assign the specified Request-URI to the data stored so that future GET requests messages containing this Request-URI will receive this data in their response messages.
DELETE	Respond to future HTTP request messages that contain the specified Request-URI with a response indicating that there is no resource associated with this Request-URI.
TRACE	Return a copy of the complete HTTP request message, including start line, header fields, and body received by the server. Used for testing purposes.

Header Fields and MIME Types

- We've already seen that the Host header field is used when forming the URI associated with an HTTP request (see page 24). The Hose header field is required in every HTTP/1.1 request message.
- HTTP/1.1 also defines a number of other header fields, several of which are commonly used by modern browsers.
- Each header field begins with a field name, such as Host, followed by a colon and then a field value. White space is allowed to precede or follow the field value, but such white space is not considered part of the value itself.
- The example on the next page (slightly modified), represents an actual HTTP request sent by a browser consisting of a start line, 10 header fields, and a short message body.

Example HTTP Request Message

```
POST /servlet/EchoHttpRequest  HTTP/1.1
host: www.example.org:56789
user-agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.4)
   Gecko/20030624
accept: text/xml,application/xml,application/xhtml+xml,
   text/html;q=0.0,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,
   image.gif;q=0.2,*/*;q=0.1
accept-language:en-us,en;q=0.5
accept-encoding:gzip,deflate
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
connection:  keep-alive
keep-alive: 300
content-type: application/x-www-form-urlencoded
content-length: 13

doit=Click+me
```


Header Fields and MIME Types

- Before we look at each of the header fields, it will be helpful to understand some common header field features:

1. Header names are not case sensitive, although they often appear as defined in the HTTP/1.1 reference [RFC-2616].

2. A header field value may wrap onto several lines by preceding each continuation line with one or more spaces or tabs (see the lines on page 35 for User-Agent and Accept fields)

3. The header field name must begin in the first character in a line, with no preceding white space.

4. MIME types are used in several header field values. MIME is an acronym for Multipurpose Internet Mail Extensions, and refers to a standard that can be used to pass a variety of types of information, including graphics and applications, through e-mail as well as through other Internet message protocols.

Header Fields and MIME Types

5. Header field values may use so-called quality values to indicate preferences. A quality value is specified by a string of the form `:q=num` where `num` is a decimal number between 0 and 1, with a higher number representing greater preference. Each quality value applies to all of the comma separated field values preceding it back to the next earlier quantity values.

   – In the example on page 35, according to the Accept header field, the browser in this example prefers text/xml (quality value 0.9) over image/jpeg (quality value 0.2).

6. The * character in a header field is a wildcard character. For instance, the string */* in the Accept header field value represents all possible MIME types.

- Each of the header fields shown in the example on page 35 are briefly described in the table on the following two pages.

Some Common HTTP/1.1 Request Header Fields

Field Name	Use
Host	Specify authority portion of URL (plus host port number). Used to support virtual hosting (running separate web servers for multiple fully qualified domain names sharing a single IP address.
User-Agent	A string identifying the browser or other software that is sending the request.
Accept	MIME types of documents that are acceptable as the body of the response, possibly with indication of preference ranking (quality value). If the server can return a document according to one of several formats, it should use a format that has the highest possible preference rating in this header.
Accept-Language	Specifies the preferred language(s) for the response body. A sever may have several translations of a document, and among these should return the one that has the highest preference rating in this header field.
Accept-Encoding	Specifies the preferred encoding(s) for the response body. For example, if a server wishes to send a compressed document (to reduce transmission time), it may one use one of the types of compression specified in this header.
Accept-Charset	Allows the client to express preferences to a server than can return a document using various character sets.

Some Common HTTP/1.1 Request Header Fields (cont.)

Field Name	Use
Connection	Indicates whether or not the client would like the TCP connection kept open after the response is sent. Typical values are `keep-alive` if the connection should be kept open (the default behavior for servers/clients compatible with HTTP/1.1), and `close` if not.
Keep-Alive	Number of seconds TCP connection should be kept open.
Content-Type	The MIME type of the document contained in the message body, if one is present. If this field is present in a request message, it normally has the value shown in the example, `application/x-www-form-urlencoded`.
Content-Length	Number of bytes of data in the message body, if one is present. In the example on page 36, this number is 13 since `doit=Click+me` contains 13 characters (bytes).
Referer	(Yes, it is spelled correctly!) The URI of the resource from which the browser obtained the Request-URI value for this HTTP request. For example, if the user clicks on a hyperlink in a web-page, causing an HTTP request to be sent to a server, the URI of the web page containing the hyperlink will be sent in the Referer field of the request. This field is not present if the HTTP request was generated by the user entering a URI in the browser's Location bar.

HTTP Response Message

- Every HTTP response message has the same basic structure:

  Status line

  Header field(s) (one or more)

  Blank line

  Message body (optional)

- The status line in the example on page 25 was:

  HTTP/1.1 200 OK

HTTP Response Message

- Like the start line of a request message, the status line of a response message consists of three parts, with a single space used to separate adjacent parts:

  1. The HTTP version used by the server software when formatting the response.
  2. A numeric status code indicating the type of the response.
  3. A text string (the reason phrase) that presents the information represented by the number status code in human-readable form.

- In the example on page 25, the status code is 200 and the reason phrase is OK. This particular status code indicates that no errors where detected by the server. The body of the response containing this status code should contain the resource requested by the client.

HTTP Response Message – Status Codes

- All status codes are three-digit decimal numbers.
- The first digit represents the general class of status code. There are five classes of HTTP/1.1 status codes and these are shown in the table on page 45.
- The last two digits of a status code define the specific status within the specified class. A few of the more common status codes are shown in the table on page 46.
- The HTTP standard recommends reason phrases for all status codes, but a server may use alternative but equivalent phrases. All status codes and recommended phrases are contained in [RFC-2616].

HTTP/1.1 Status Code Classes

(First Digit of Status Code)

Digit	Class	Standard Use
1	Informational	Provide information to client before request processing has been completed.
2	Success	Request has been successfully processed.
3	Redirection	Client needs to use a different resource to fulfill request.
4	Client Error	Client's request is not valid.
5	Server Error	An error occurred during server processing of a valid client request.

Some Common HTTP/1.1 Status Codes

Status Code	Recommended Reason Phrase	Usual Meaning
200	OK	Request processed normally.
301	Moved Permanently	URI for the requested resource has been changed. All future requests should be made to URI contained in the Location header field of the response. Most browsers will automatically send a second request to the new URI and display the second response.
307	Temporary Redirect	URI for the requested resource ahs been changed at least temporarily. This request should be fulfilled by making a second request to the URI contained in the Location header field of the response. Most browsers will automatically send a second request to the new URI and display the second response.
401	Unauthorized	The resource is password protected, and the user has not yet supplied a valid password.
403	Forbidden	The resource is present on the server but is read protected.
404	Not Found	No resource corresponding to the given Request-URI was found at this server.
500	Internal Server Error	Server software detected an internal failure.

HTTP Response Message – Header Fields

- Some of the header fields used in HTTP request messages, including Connection, Content-Type, and Content-Length, are also valid in response messages.
- The Content-Type of a response can be any one of the MIME type values specified by the Accept header field of the corresponding request.
- Some other common response header fields are shown in the table on page 48.

Some Common HTTP/1.1 Response Header Fields

Field Name	Use
Date	Time at which response was generated. Used for cache control. This field must be supplied by the server.
Server	Information identifying the server software generating this response.
Last-Modified	Time at which the resource returned by this request was last modified. Can be used to determine whether a cached copy of a resource is valid or not.
Expires	Time after which the client should check with the server before retrieving the returned resource from the client's cache.
Accept-Ranges	Clients can request that only a portion (range) of a resource be returned by using the Range header field. This might be used if the resource is, say, a large PDF file and only a single page is currently needed. Accept-Ranges specifies the units that may be used by the client in a range request, or none, if range requests are not accepted by this server for this resource.
Location	Used in responses with redirect status code to specify new URI for the requested resource.