

# Chapter 27

# Hypertext Transfer

# Protocol

Raj Jain

**Raj Jain is now at**  
**Washington University in Saint Louis**  
**Jain@cse.wustl.edu**  
<http://www.cse.wustl.edu/~jain/>



- ❑ Hypertext language and protocol
- ❑ HTTP messages
- ❑ Browser architecture
- ❑ CGI and Java

# Terminology

- ❑ Browser
- ❑ Hypertext: Documents contains pointers to other documents
- ❑ Media = Text, Graphics, Images, voice, animation, video
- ❑ Hypermedia: Documents contain pointers to other media
- ❑ Browser = User program

# Uniform Resource Locator

- ❑ URL = Uniform resource locator =  
protocol://site/directory/file or mailto:email\_address
- ❑ Protocol = http, ftp, gopher, telnet, nntp, smtp, wais
  - ❑ ftp://netlab.ohio-state.edu/pub/jain/courses/cis678/f27\_htt.html
  - ❑ http://www.cis.ohio-state.edu/~jain/cis678.htm
  - ❑ mailto:jain@cis.ohio-state.edu
- ❑ Hypertext transfer protocol (HTTP)
- ❑ Hypertext markup language (HTML)
- ❑ HTTP Daemon (httpd)

# HyperText Markup Language (HTML)

- Header, Body, Anchors, Hyper References

- Sample Code:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Hello</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<A HREF="http://www.cis.ohio-  
state.edu/~jain/greetings.html">How are you</A>
```

```
</BODY>
```

```
</HTML>
```

# Output

Hello

How are you?

# A Sample List of HTML Tags

<code>&lt;A&gt; &lt;/A&gt;</code>	Anchor (link or name)
<code>&lt;BODY&gt; &lt;/BODY&gt;</code>	Contents
<code>&lt;BR&gt;</code>	Break
<code>&lt;FORM&gt; &lt;/FORM&gt;</code>	Input form
<code>&lt;H1&gt; &lt;/H1&gt;</code>	Heading level 1
<code>&lt;HEAD&gt; &lt;/HEAD&gt;</code>	Header of a document
<code>&lt;HR&gt;</code>	Horizontal Rule
<code>&lt;HTML&gt; &lt;/HTML&gt;</code>	The doc type is HTML
<code>&lt;LI&gt;</code>	List Item
<code>&lt;OL&gt; &lt;/OL&gt;</code>	Ordered List
<code>&lt;P&gt;</code>	Paragraph break
<code>&lt;PRE&gt; &lt;/PRE&gt;</code>	Preformatted text
<code>&lt;TITLE&gt; &lt;/TITLE&gt;</code>	Document title
<code>&lt;UL&gt;</code>	Unnumbered list

# Sample HTTP Exchange

telnet www.cis.ohio-state.edu http

Connected to tholian.cis.ohio-state.edu.

Escape character is '^['.

GET /~jain/temp.html HTTP/1.0

Accept: text/plain, text/html

HTTP/1.0 200 OK

Server: Netscape-Enterprise/2.0a

Date: Tue, 25 Feb 1997 05:04:11 GMT

Accept-ranges: bytes

Last-modified: Tue, 25 Feb 1997 05:03:07 GMT

Content-length: 84

Content-type: text/html



```
<HTML>
<HEAD>
<TITLE>Hello</TITLE>
</HEAD>
<BODY>
Hello! How are you?
</BODY>
</HTML>
```

Connection closed by foreign host.

# HTTP Requests

- ❑ **GET** Return the contents
- ❑ **HEAD** Return the header
- ❑ **POST** Treat the document as a script and send some data to it
- ❑ **PUT** Replace the contents with some data
- ❑ **DELETE** Delete the indicated document

# HTTP Request Headers

Header	Description
From	Email address of user
User-Agent	Client s/w
Accept File	File types that client will accept
Accept-encoding	Compression methods
Accept-Language	Languages
Referrer	URL of the last document the client displayed
If-Modified-Since	Return document only if modified since specified
Content-length	Length (in bytes) of data to follow

# HTTP Status Codes

Code	Text
2xx	Success
3xx	Redirection
301	Moved
302	Found
4xx	Client Errors
400	Bad Request
401	Unauthorized
404	Not found
5xx	Server Errors
500	Internal Error
502	Service overloaded

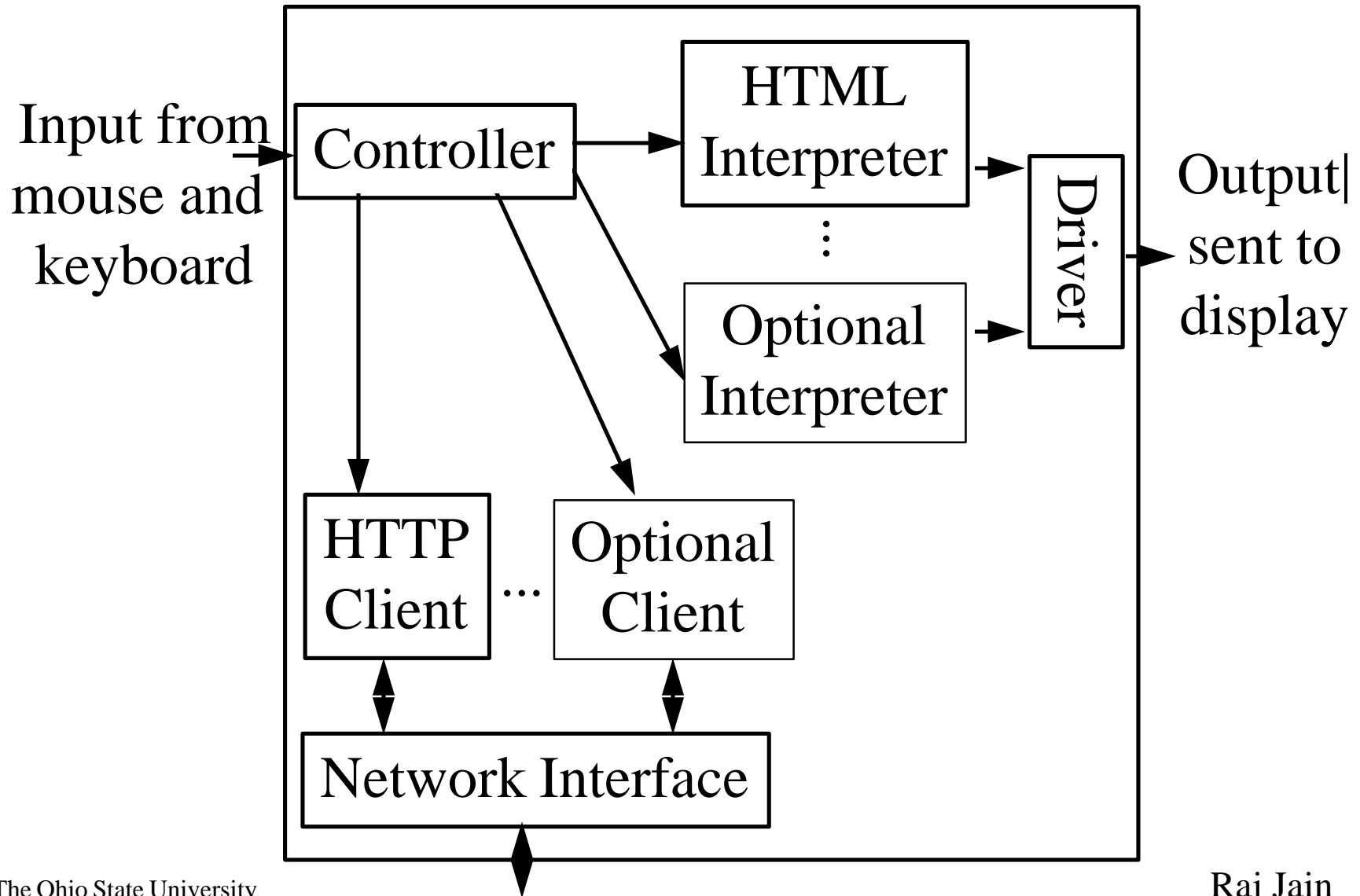
# HTTP Response Headers

Header	Description
Server	Server software
Date	Current Date
Last-Modified	Modification date of document
Expires	Date at which document expires
Location	The location of the document in redirection responses
Pragma	A hint, e.g., no cache
MIME-version	
Link	URL of document's parent
Content-Length	Length in bytes
Allowed	Requests that user can issue, e.g., GET

# HTTP 1.1 Features

- ❑ Persistent TCP Connections: Remain open for multiple requests
- ❑ Partial Document Transfers: Clients can specify start and stop positions
- ❑ Conditional Fetch: Several additional conditions
- ❑ Better content negotiation
- ❑ More flexible authentication

# Browser Architecture



# Key Decisions

- ❑ Presentation separate from content
  - ❑ Presentation controlled by the browser
  - ❑ Content provided by the server
- ❑ Short term interaction  $\Rightarrow$  Open connection, exchange, close connection



# Common Gateway Interface (CGI)

- ❑ Dynamic documents: Content changes with time, e.g., counter
- ❑ Initial solution: Server-side includes  $\Rightarrow$  Special commands to servers
- ❑ Later, a general interface developed at National Center for Supercomputing Applications (NCSA)
- ❑ CGI programs output hypertext and commands to the server
- ❑ Can be in any language, e.g., C, C++, or even be a shell script

# CGI Example

Content/type: text/plain

Location: /new/bar.txt

<blank line>

⇒ send <http://site/new/bar.txt>

# CGI Parameters

- ❑ `http://www.cis.ohio-state.edu/cgi-bin/register.sh?fname=raj&lname=jain`
- ❑ A question mark separates prefix and suffix (parameters)
- ❑ The parameters are passed to the CGI program using environment variables

# CGI Environmental Variables

Variable	Meaning
Server_name	Domain name of the server computer
Gateway_Interface	Version of CGI s/w
Script_name	Path in URL after server name
Query_string	Information following ?
Remote_addr	Browser's address

# State Information

- ❑ Passed in parameters
- ❑ `http://www.cis.ohio-state.edu/cgi-bin/counter.sh?$N`
- ❑ Example:

```
#!/bin/sh
```

```
echo content/type: text/plain
```

```
echo
```

```
N=$QUERY_STRING
```

```
echo "<HTML>"
```

```
echo You have clicked this $N times
```

```
echo "<A HREF=\"http://cis.ohio-state.edu/cgi-bin/counter.sh?$N\">"
```

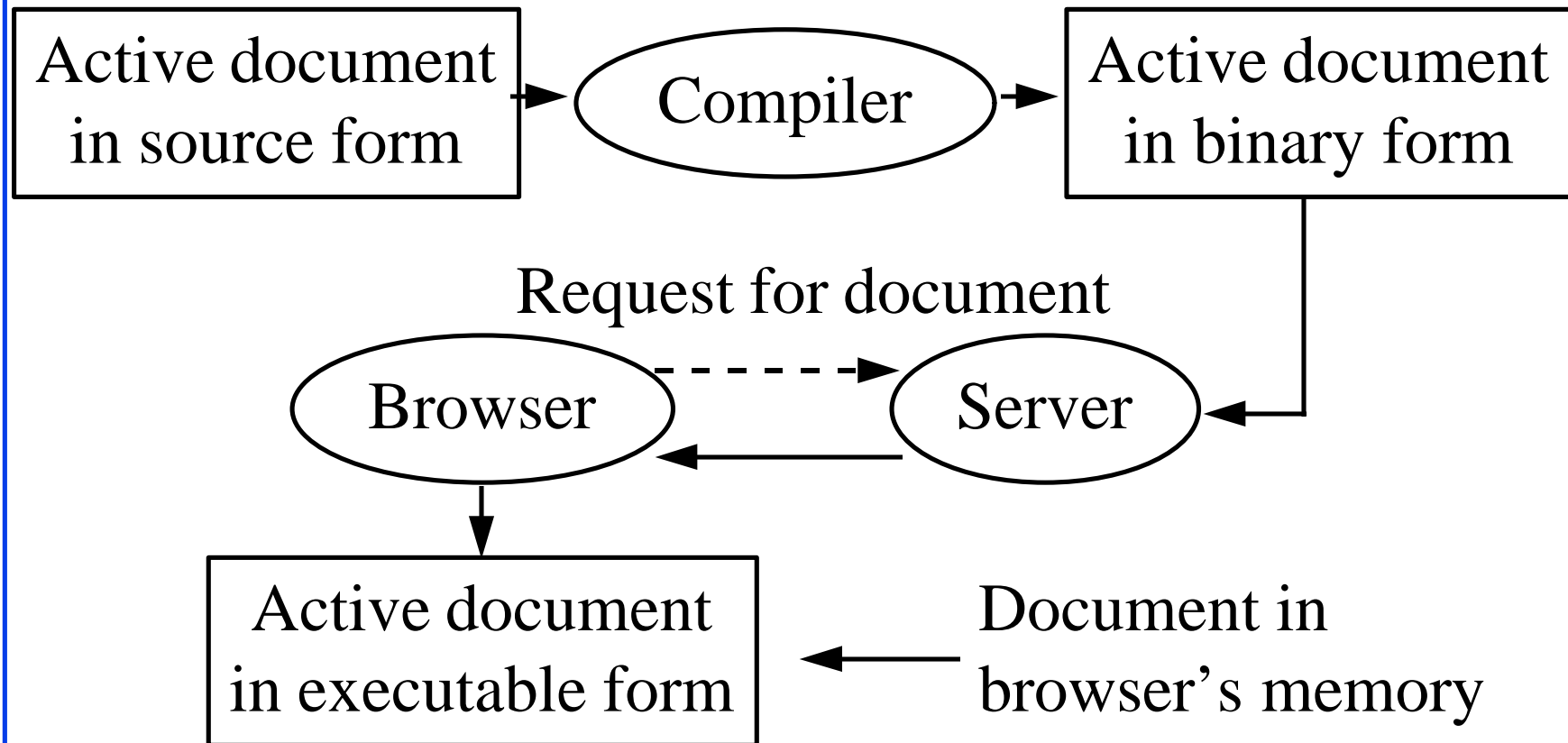
```
echo "Click Here to refresh the page.</A></HTML>"
```

You have clicked this 4 times [Click Here to refresh the page.](http://cis.ohio-state.edu/cgi-bin/counter.sh?4)

# Active Documents

- ❑ Active = Program
- ❑ The output of the program is displayed continuously
- ❑ Useful for animation, Continuous update (clock)
- ❑ Need a program that can run on any computer
- ❑ Java = Just Another Vague Acronym?  
Not really, the name invented after at a coffee shop.
- ❑ Invented by James Gosling at Sun Microsystems
- ❑ Java programs consist of byte codes that can be interpreted on any computer/operating system.
- ❑ Java provides programming language, runtime environment, class library

# Active Documents (Cont)



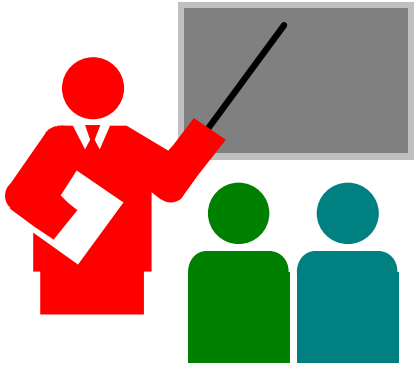
# Characteristics of Java

- ❑ High Level  $\Rightarrow$  Hides hardware details
- ❑ General Purpose
- ❑ Object Oriented. Similar to C++
- ❑ Dynamic  $\Rightarrow$  Object instances are created at run time
- ❑ Strongly typed  $\Rightarrow$  Each data item has a type.  
Operations can be applied only to particular data types.
- ❑ Statically Type Checked  $\Rightarrow$  Checked at compile time.  
Not at run-time.
- ❑ Concurrent  $\Rightarrow$  Allows multiple threads of control



# The Java Run-Time Environment

- ❑ Interpretive execution
- ❑ Automatic garbage collection
- ❑ Includes a socket library for internet access
- ❑ Graphics support



## Summary

- ❑ HTML and HTTP
- ❑ Servers use environment variables to pass parameters to CGI programs
- ❑ CGI programs can be written in any language
- ❑ Static, Dynamic, and Active documents
- ❑ Java provides applets that can be interpreted at any computer
- ❑ Dynamics is server's (CGI) or client's (Java) responsibility

# Homework

- ❑ Read Chapters 27, 28, 29
- ❑ Submit Answers to Exercises 27.6, 28.3, 29.2

# HTTP: RFCs

- ❑ [RFC1866] T. Berners-Lee, D. Connolly, "Hypertext Markup Language - 2.0", 11/03/1995, 77 pages.
- ❑ [RFC1945] T. Berners-Lee, R. Fielding, H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", 05/17/1996, 60 pages.
- ❑ [RFC2068] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, , "**Hypertext Transfer Protocol -- HTTP/1.1**", 01/03/1997, 162 pages.
- ❑ [RFC2069] J. Franks, P. Hallam-Baker, J. Hostetler, P. A. Luotonen, E. L. Stewart, "An Extension to HTTP: Digest Access Authentication", 01/03/1997, 18 pages.
- ❑ [RFC2070] F. Yergeau, G. Nicol, G. Adams, M. Duerst, "Internationalization of the Hypertext Markup Language", 01/06/1997, 43 pages.
- ❑ [RFC2109] D. Kristol, L. Montulli, "HTTP State Management Mechanism", 02/18/1997, 21 pages.