Introduction to Internet Programming

Jim Fawcett

CSE686 – Internet Programming

Summer 2006

References

- Dr. Sapossnek, Boston Univ., has a series of presentations on various topics relating to internet programming with Microsoft .Net
  http://www.gotdotnet.com/team/student/academicreskit/
- Paul Amer, Univ. Del., Hyper Text Transfer Protocol (HTTP)
  http://www.cis.udel.edu/~amer/856/http.03f.ppt
- World Wide Web Consortium
  www.w3c.org
- Our website
  www.ecs.syr.edu/faculty/fawcett/handouts/webpages/webdev.htm

Internet History

- 1961 – First paper on packet-switching theory
  - Kleinrock, MIT
- 1969 – ARPANet goes on line
  - Four hosts, each connected to at least two others
- 1974 – TCP/IP, Berkley Sockets invented
- 1983 – TCP/IP becomes only official protocol
- 1983 – Name server developed at University of Wisconsin.
- 1984 – Work begins on NSFNET
- 1990 – ARPANET shutdown and dismantled
- 1990 – ANSNET takes over NSFNET
  - Non-profit organization – MERIT, MCI, IBM
  - Starts commercialization of the internet
- 1995 – NSFNET backbone retired

Web History

- 1990 – World Wide Web project
  - Tim Berners-Lee starts project at CERN
  - Demonstrates browser/editor accessing hypertext files
  - HTTP 0.9 defined, supports only hypertext, linked to port 80
- 1991 – first web server outside Europe
  - CERN releases WWW, installed at Stanford Linear Accelerator Center
- 1992 – HTTP 1.0, supports images, scripts as well as hypertext
- 1993 – Growth phase – exponential growth through 2000
- 1994 – CERN and MIT agree to set up WWW Consortium
- 1999 – HTTP 1.1, supports open ended extensions

Original Goals of the Web

- Universal readership
  - When content is available it should be accessible from any type of computer, anywhere.
- Interconnecting all things
  - Hypertext links everywhere.
  - Simple authoring

Web Design Principles

- Universal
- Decentralized
- Modular
- Extensible
- Scalable
- Accessible
- Forward/backwards compatibility

Basic Concepts

- ***Universal Addressing***
  - TCP/IP, DNS
- ***Universal Processing Protocols***
  - URLs, HTTP, HTML, FTP
- ***Format Negotiation through HTTP***
- ***Hypertext → Hypermedia via HTML → XHTML***
  - Support for text, images, sound, and scripting
- **Client/Server Model**

Servers on the Internet

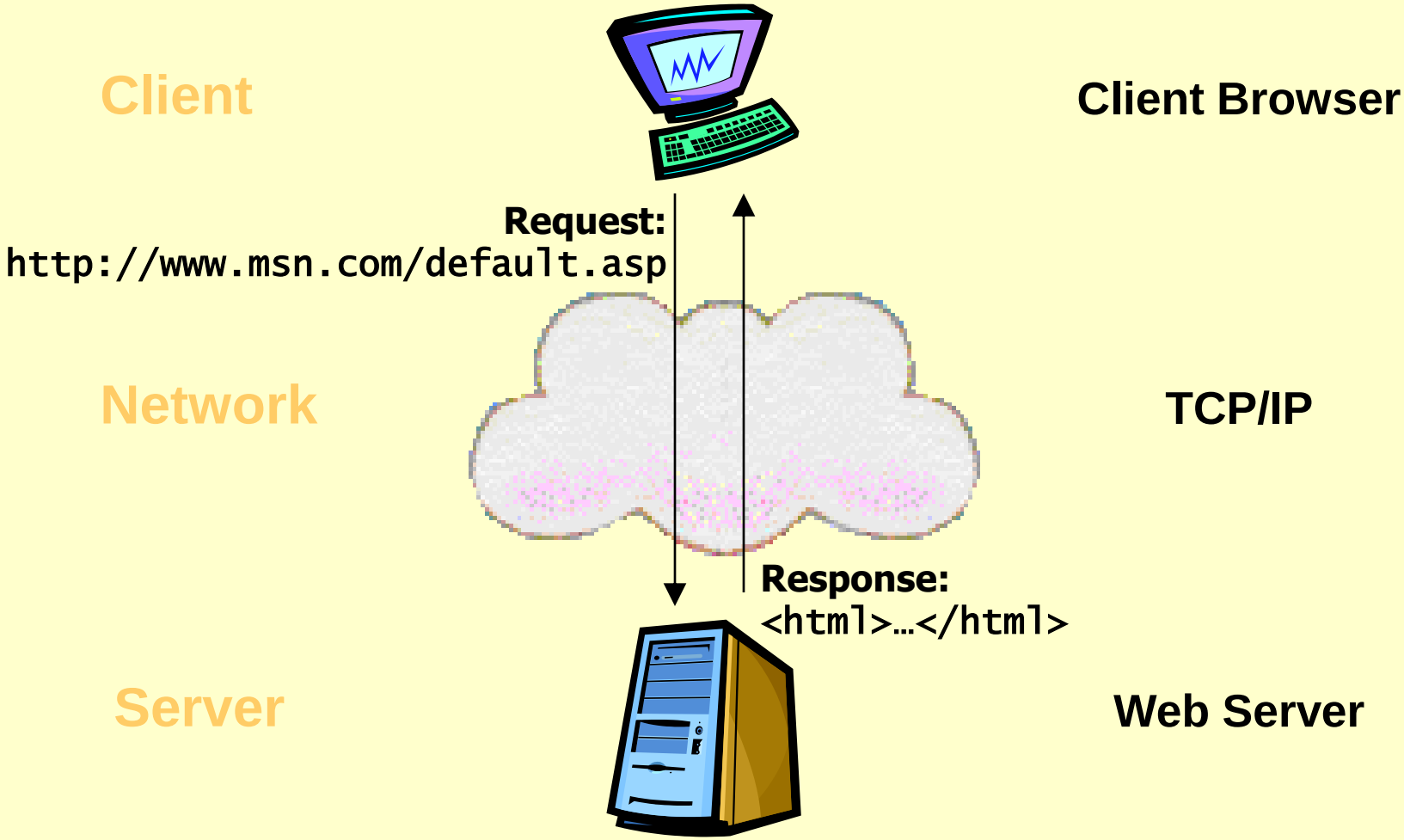
- HTTP - HyperText Transport Protocol
- FTP - File Transport Protocol
- Gopher - Text and Menus
- NNTP - Network News Transfer Protocol
- DNS - Distributed Name Service
- telnet - log into a remote computer
- Web services
  - coming soon to a web server near you

HyperText Markup Language (HTML)

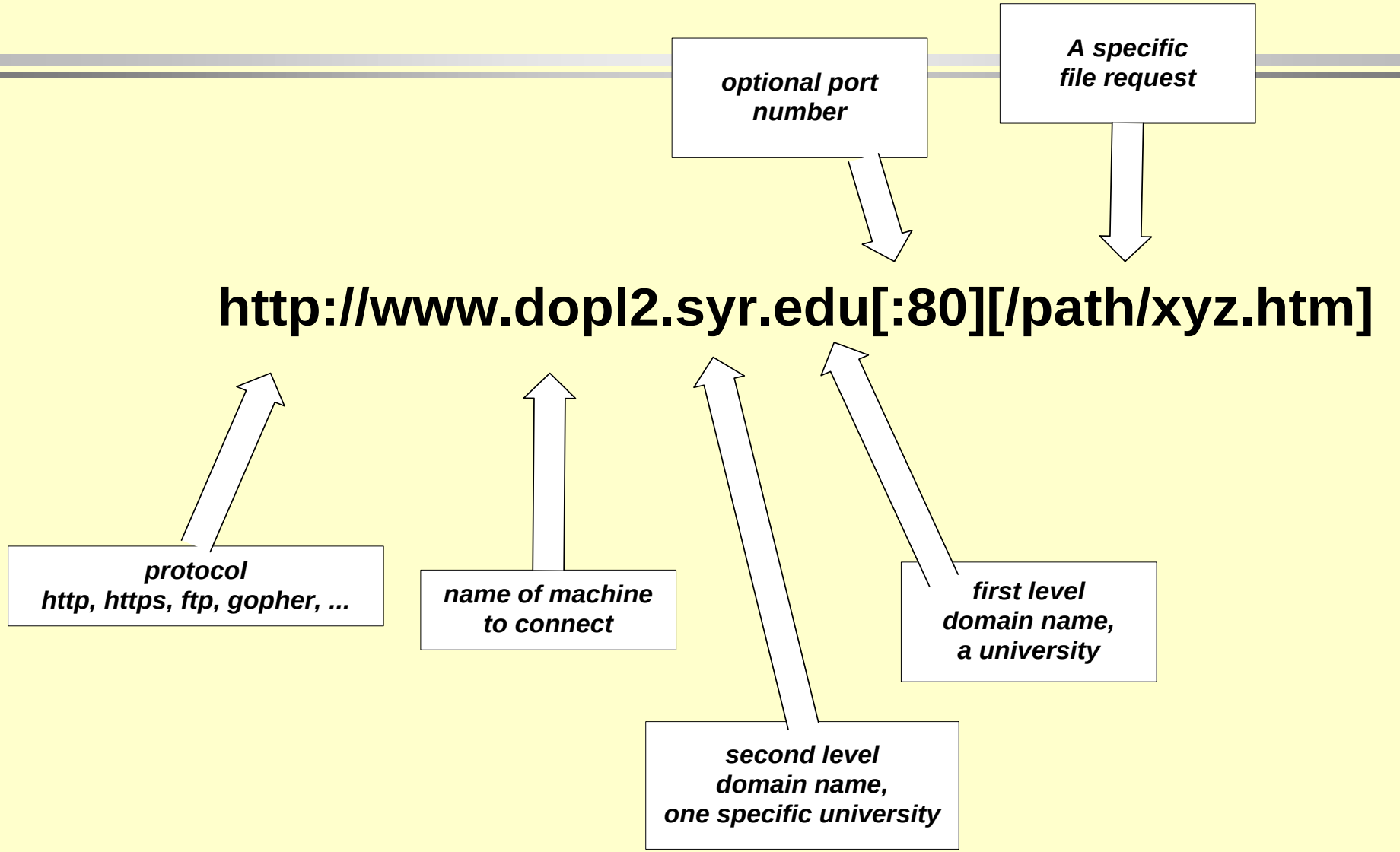
- The markup language used to represent Web pages for viewing by people
  - Designed to display data, not store/transfer data
- Rendered and viewed in a Web browser
- Can contain ***links*** to images, documents, and other pages
- Not extensible – uses only tags specified by the standard
- Derived from Standard Generalized Markup Language (SGML)
- HTML 3.2, 4.01, XHTML 1.0

Internet Technologies

WWW Architecture



Address Resolution



Some Interesting Views of the Internet

The following plots are from the Cooperative Association for Internet Data Analysis

- http://www.caida.org
- http://www.caida.org/tools/visualization/walrus/gallery1/
- http://www.caida.org/tools/visualization/plankton/Images/

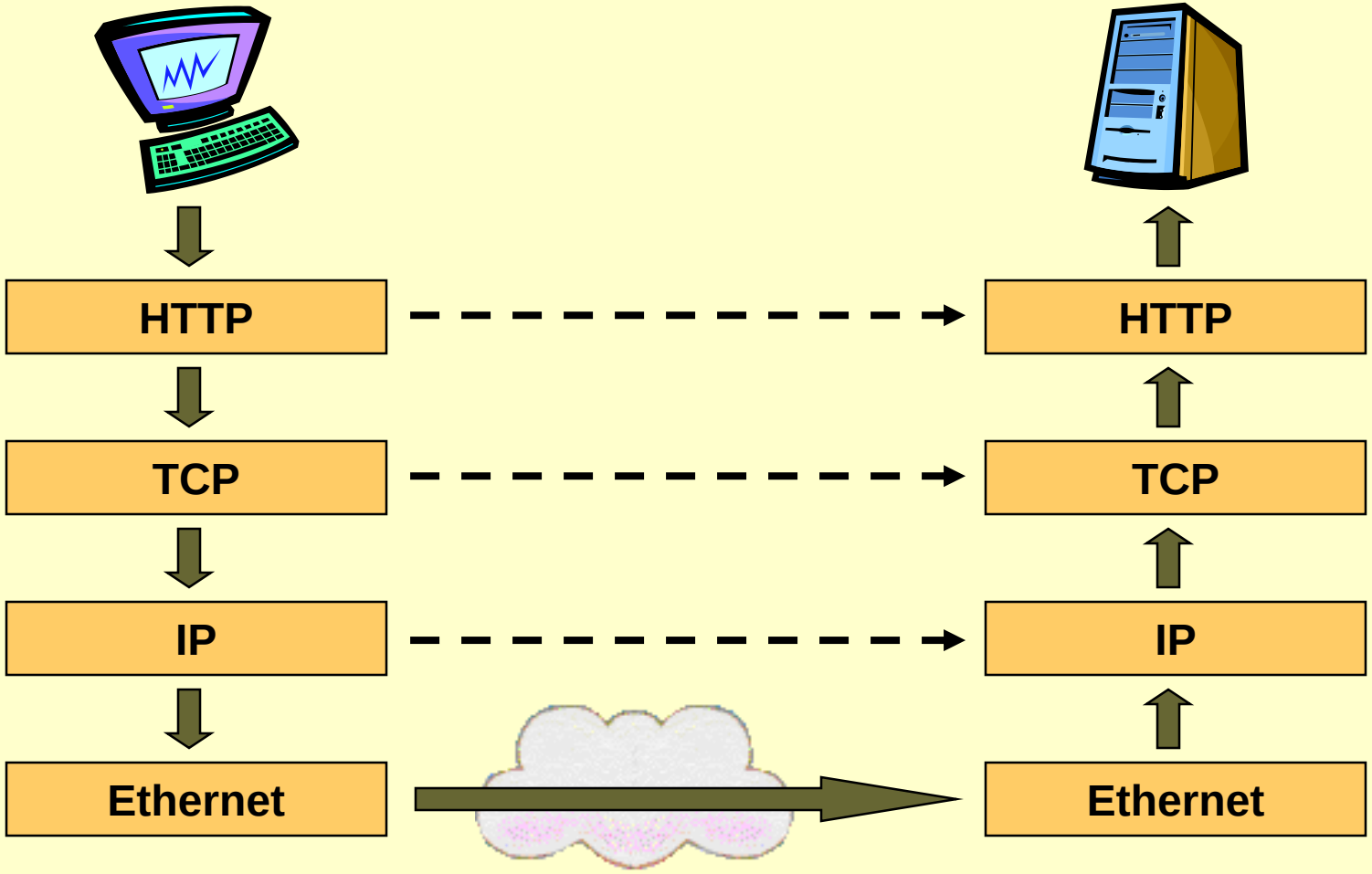
Networks

- Network = an interconnected collection of independent computers

- Why have networks?
  - Resource sharing
  - Reliability
  - Cost savings
  - Communication

- Web technologies add:
  - New business models: e-commerce, advertising
  - Entertainment
  - Applications without a client-side install

Network Protocol Stack



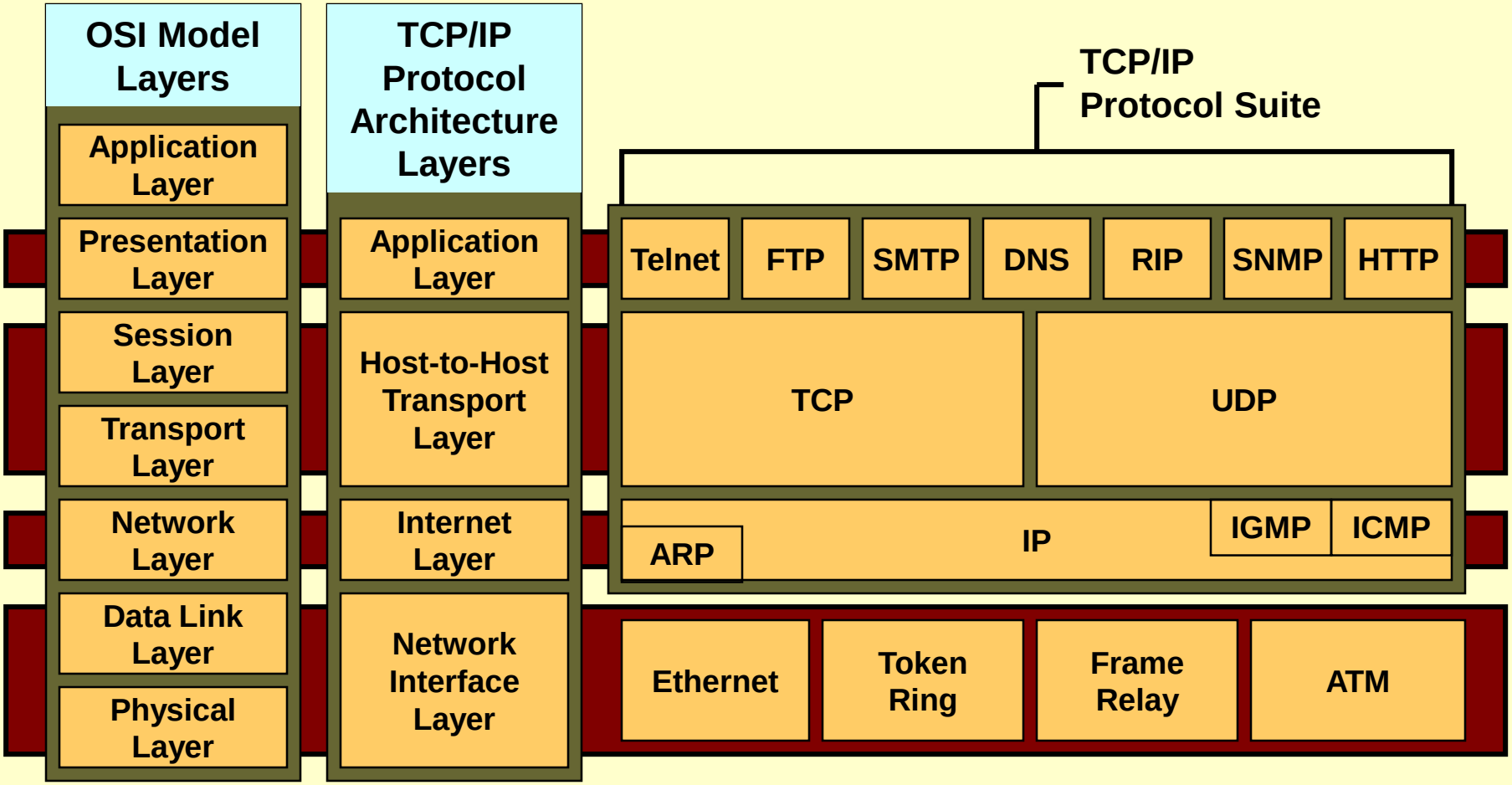
Networks - Transport Layer

- Provides efficient, reliable and cost-effective service
- Uses the Sockets programming model
- Ports identify application
  - Well-known ports identify standard services (e.g. HTTP uses port 80, SMTP uses port 25)
- Transmission Control Protocol (TCP)
  - Provides reliable, connection-oriented byte stream
- UDP
  - Connectionless, unreliable

Communication Between Networks

- Internet Protocol (IP)
  - Routable, connectionless datagram delivery
  - Specifies source and destination
  - Does not guarantee reliable delivery
  - Large message may be broken into many datagrams, not guaranteed to arrive in the order sent
- Transport Control Protocol (TCP)
  - Reliable stream transport service
  - Datagrams are delivered to the receiving application in the order sent
  - Error control is provided to improve reliability

Network Protocols



HTTP Protocol

- Client/Server, Request/Response architecture
  - You request a Web page
    - e.g. `http://www.msn.com/default.asp`
    - HTTP request
  - The Web server responds with data in the form of a Web page
    - HTTP response
    - Web page is expressed as HTML
  - Pages are identified as a Uniform Resource Locator (URL)
    - Protocol: `http`
    - Web server: `www.msn.com`
    - Web page: `default.asp`
    - Can also provide parameters: `?name=Leon`

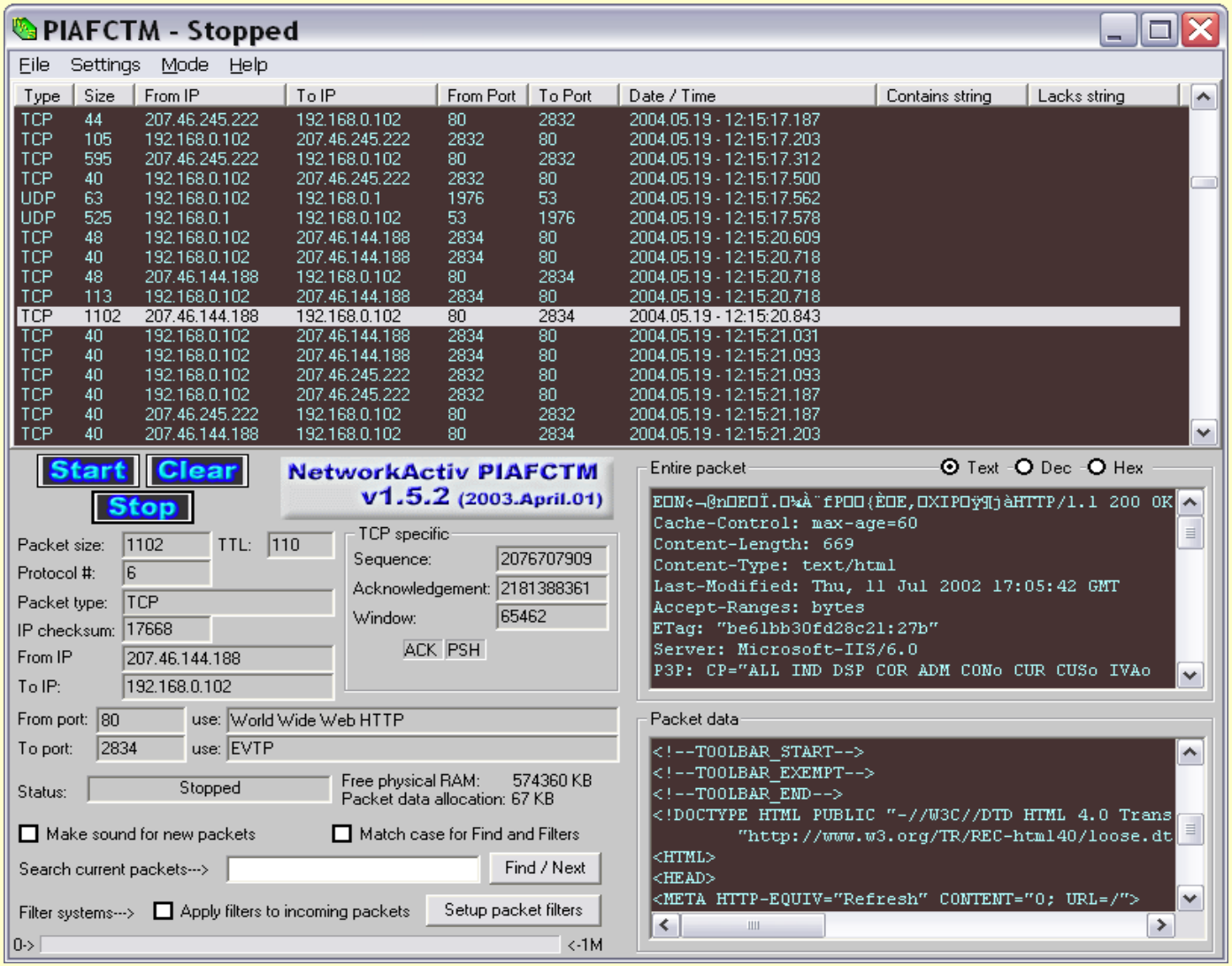
HTTP is Stateless

- HTTP is a stateless protocol
- Each HTTP request is independent of previous and subsequent requests
- HTTP 1.1 introduced `keep-alive` for efficiency
- Statelessness has a big impact on how scalable applications are designed

Cookies

- A mechanism to store a small amount of information (up to 4KB) on the client
- A cookie is associated with a specific web site
- Cookie is sent in HTTP header
- Cookie is sent with each HTTP request
- Can last for only one session (until browser is closed) or can persist across sessions
- Can expire some time in the future

Network Packet Sniffer



HTTP Messages

as seen by packet sniffer

```
 TCP  113   192.168.0.102  207.46.144.188   2834    80  [2004.05.19 - 12:15:20.718]

E  qSó@ €□…šÀ¨ fÏ.¼
□ P,□X {È
EP□DpÑ¼  GET /ms.htm HTTP/1.1
Connection: Keep-Alive
Host: www.microsoft.com


 TCP 1102  207.46.144.188   192.168.0.102     80  2834  [2004.05.19 - 12:15:20.843]

E □N¢¬@ n□E□Ï.¼À¨ f P
□{È
E,□XIP□ÿ¶jà  HTTP/1.1 200 OK
Cache-Control: max-age=60
Content-Length: 669
Content-Type: text/html
Last-Modified: Thu, 11 Jul 2002 17:05:42 GMT
Accept-Ranges: bytes
ETag: "be61bb30fd28c21:27b"
Server: Microsoft-IIS/6.0
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"
X-Powered-By: ASP.NET
Date: Wed, 19 May 2004 16:15:16 GMT

<!--TOOLBAR_START-->
<!--TOOLBAR_EXEMPT-->
<!--TOOLBAR_END-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
               "http://www.w3.org/TR/REC-html40/loose.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="Refresh" CONTENT="0; URL=/">
<TITLE>Microsoft Corporation -- Where Do You Want to Go Today?</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<FONT FACE="Verdana, Arial, Helvetica" SIZE=2>
If your browser can't handle redirect, please click <a href="/">here</a>
</FONT>
</BODY>
</HTML>
```

Request Message

method

Response Message

headers

message body

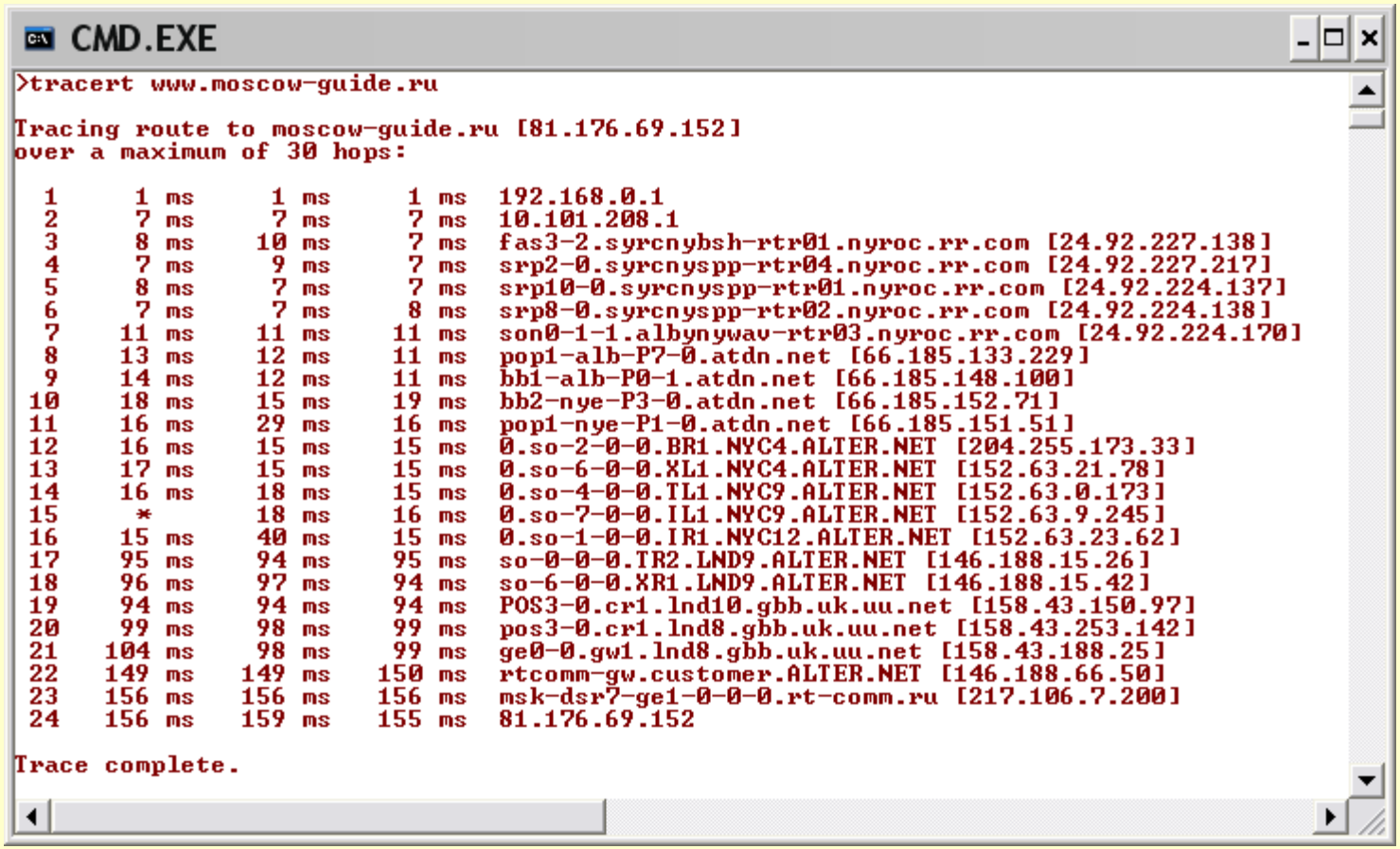
Typical HTTP Transaction

- Client browser finds a machine address from an internet Domain Name Server (DNS).
- Client and Server open TCP/IP socket connection.
- Server waits for a request.
- Browser sends a verb and an object:
  - GET XYZ.HTM or POST form
  - If there is an error server can send back an HTML-based explanation.
- Server applies headers to a returned HTML file and delivers to browser.
- Client and Server close connection.
  - It is possible for the client to request the connection stay open – requires design effort to do that.

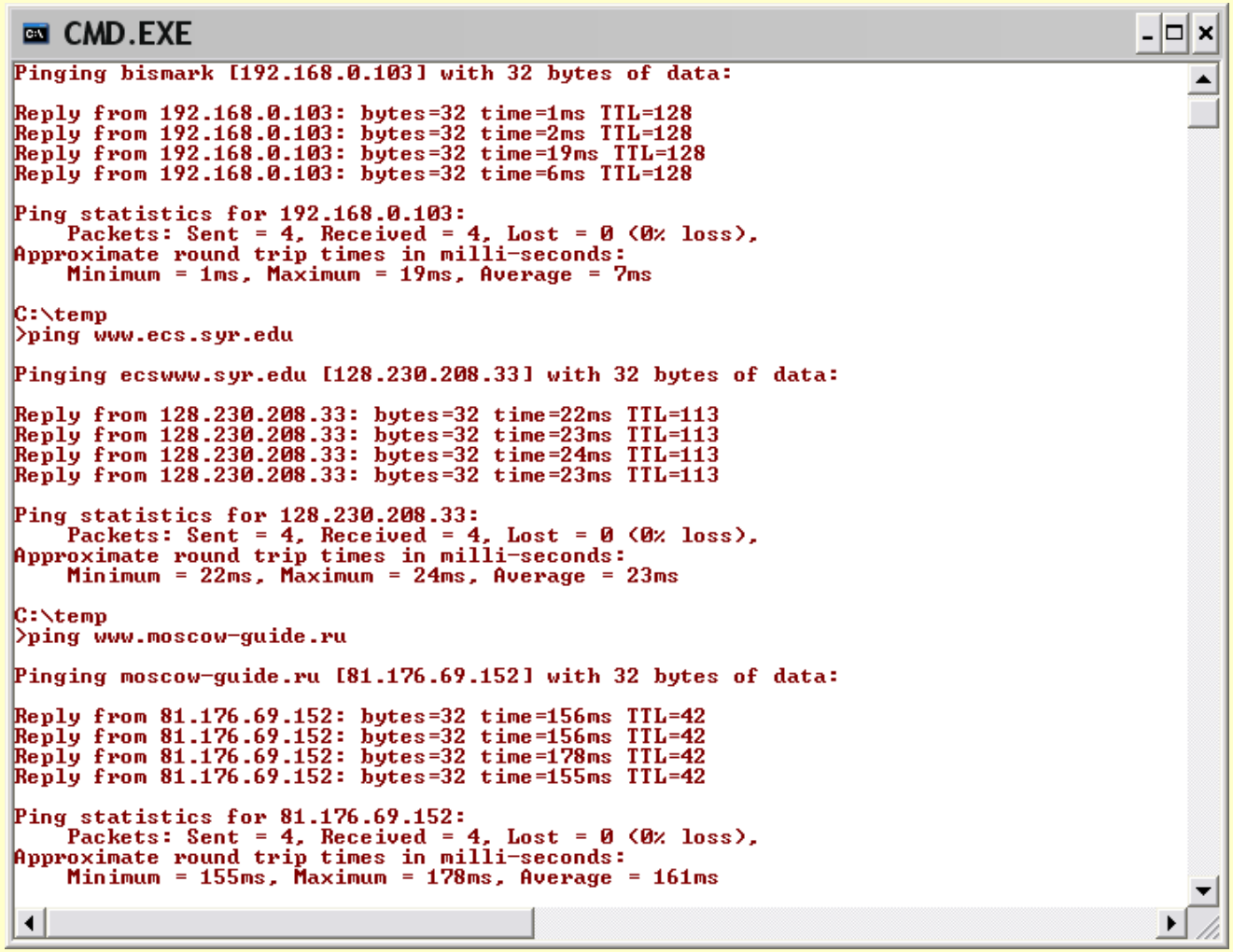
HTTP Methods

- GET request-URI HTTP/1.1
  - Retrieve entity specified in request-URI as body of response message
- POST request-URI HTTP/1.1
  - Sends data in message body to the entity specified in request-URI
- PUT request-URI HTTP/1.1
  - Sends entity in message body to become newly created entity specified by request-URI
- HEAD request-URI HTTP/1.1
  - Same as GET except the server does not send specified entity in response message
- DELETE request-URI HTTP/1.1
  - Request to delete entity specified in request-URI.
- TRACE request-URI HTTP/1.1
  - Request for each host node to report back

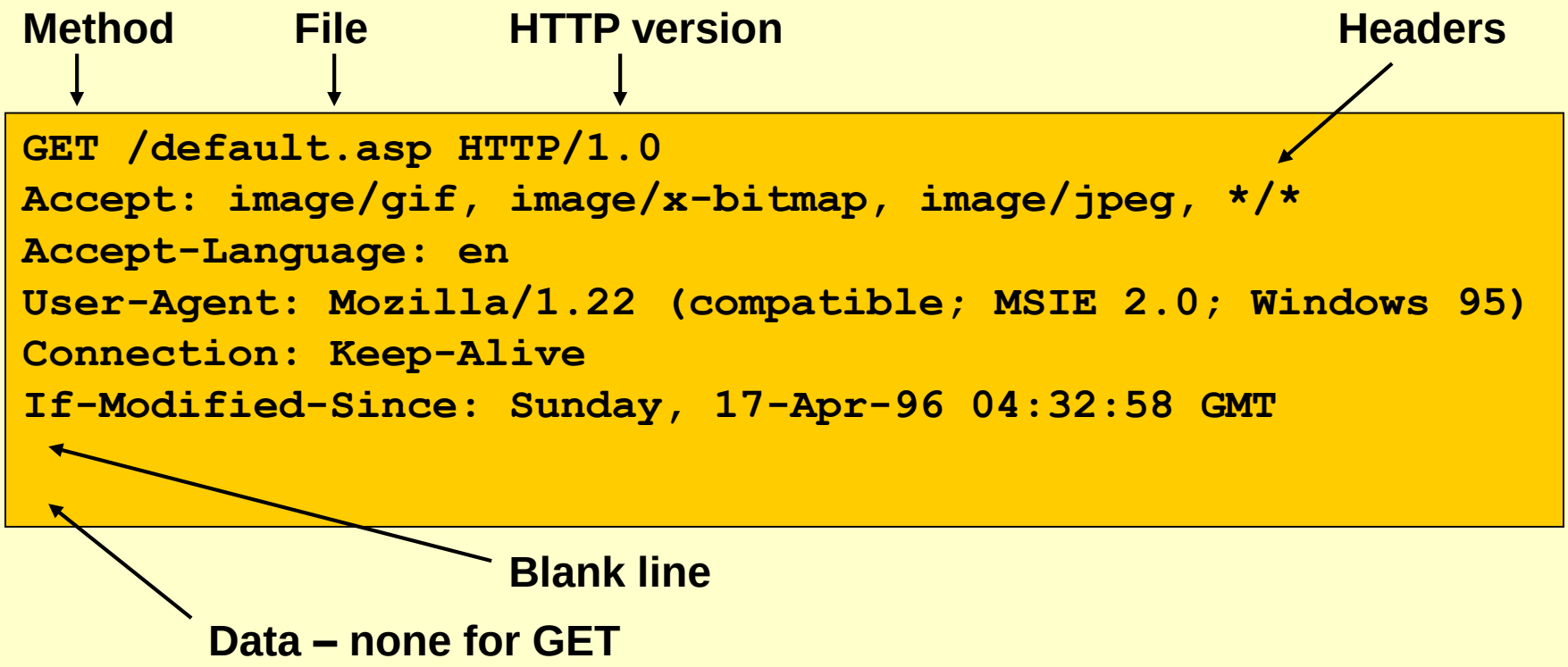
Tracing HTTP Message with Tracert



Pinging Various URLs



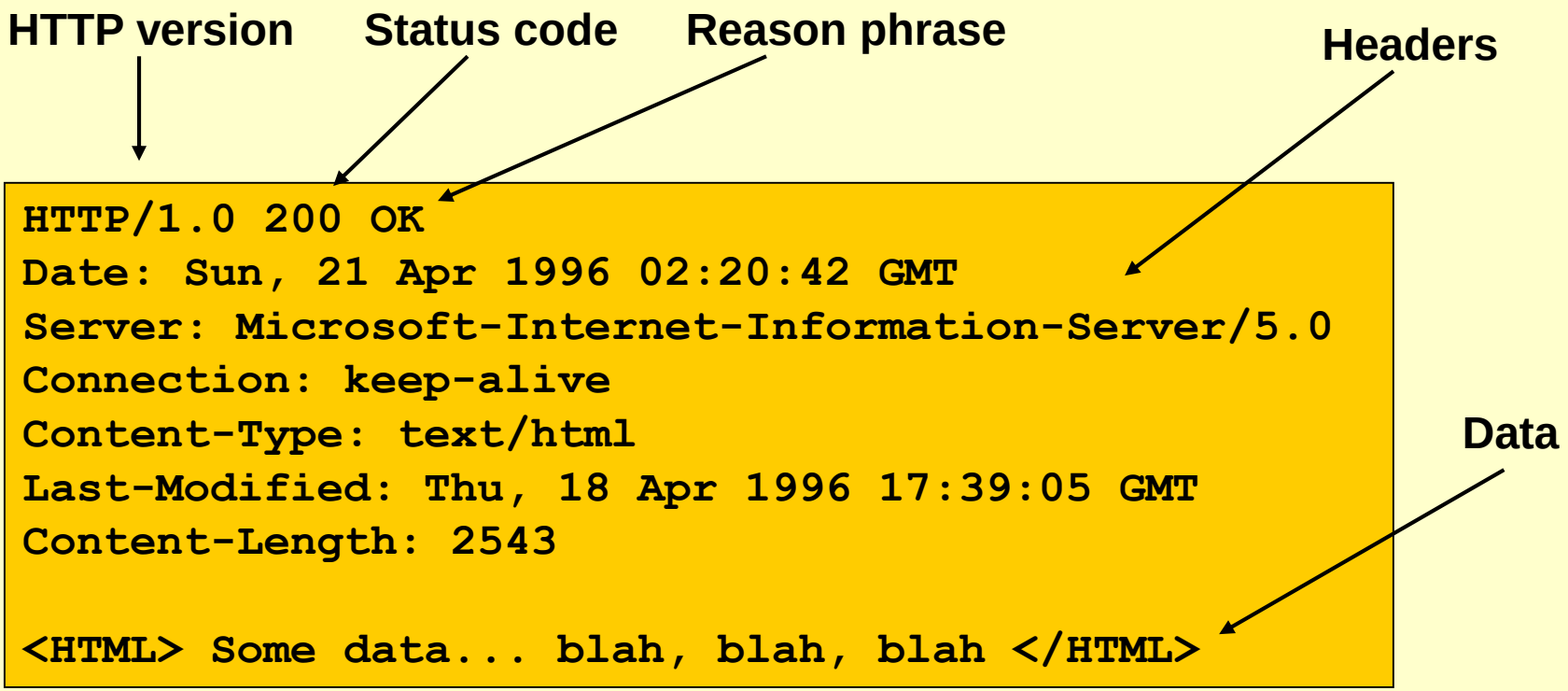
HTTP Request



Multipurpose Internet Mail Extensions (MIME)

- Defines types of data/documents
  - text/plain
  - text/html
  - image/gif
  - image/jpeg
  - audio/x-pn-realaudio
  - audio/x-ms-wma
  - video/x-ms-asf
  - application/octet-stream

HTTP Response



Status Codes

200 OK
201 Created
202 Accepted
204 No Content
301 Moved Permanently
302 Moved Temporarily
304 Not Modified
400 Bad Request
401 Unauthorized
403 Forbidden
404 Not Found
500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable

Classes:

1xx: Informational	- not used, reserved for future
2xx: Success	- action was successfully received, understood, and accepted
3xx: Redirection	- further action needed to complete request
4xx: Client Error	- request contains bad syntax or cannot be fulfilled
5xx: Server Error	- server failed to fulfill an apparently valid request

Programming the Web

- Client-Side Programming
  - JavaScript
  - Dynamic HTML
  - .Net controls

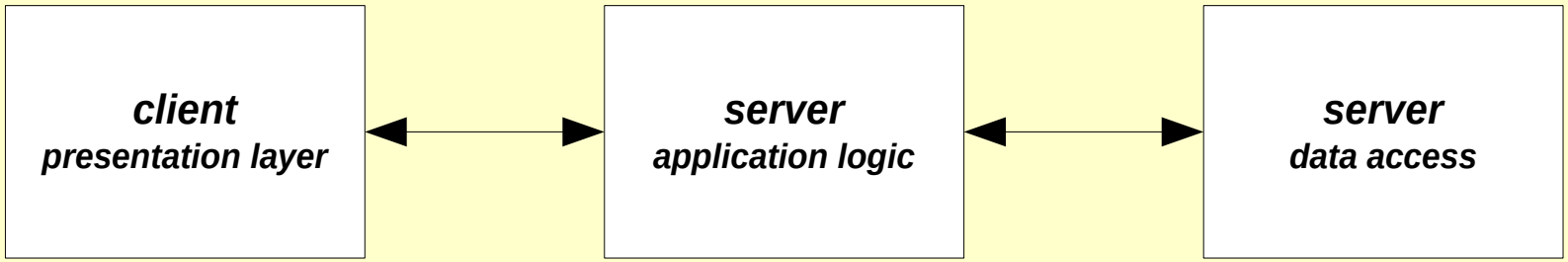
- Server-Side Programming
  - ASP script
  - Server components
  - C# code-behind
  - ADO
  - Web controls used on ASPX pages
  - Web services

Web Processing Models

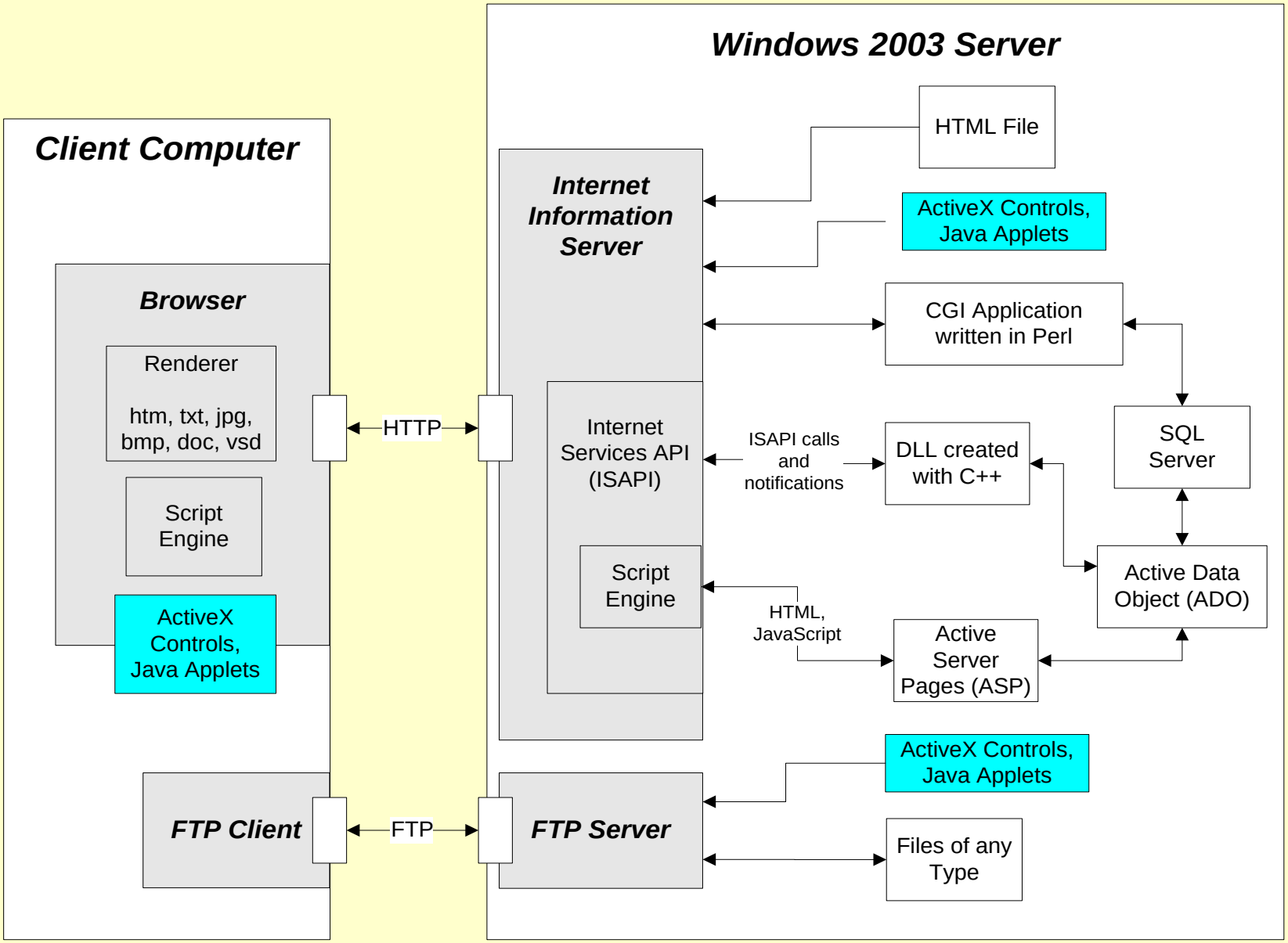
- ***HyperText Transfer Protocol (HTTP)***
  - Universal access
  - HTTP is a "request-response" protocol specifying that a client will open a connection to server then send request using a very specific format. Server will respond and then close connection.
- ***HyperText Markup Language (HTML)***
  - Web of linked documents
  - Unlimited scope of information content
- ***Graphical Browser Client***
  - Sophisticated rendering makes authoring simpler
- ***HTML File Server***
  - Using HTTP, Interprets request, provides appropriate response, usually a file in HTML format
- ***Three-Tier Model***
  - Presentation, application logic, data access

Three Tier Architecture

- Client Tier
  - Presentation layer
  - Client UI, client-side scripts, client specific application logic
- Server Tier
  - Application logic, server-side scripts, form handling, data requests
- Data Tier
  - Data storage and access



Client/Server - Current Web Model



Programming the Web
Client-Side Code

- What is client-side code?
  - Software that is downloaded from Web server to browser and then executes on the client
- Why client-side code?
  - Better scalability: less work done on server
  - Better performance/user experience
  - Create UI constructs not inherent in HTML
    - Drop-down and pull-out menus
    - Tabbed dialogs
  - Cool effects, e.g. animation
  - Data validation

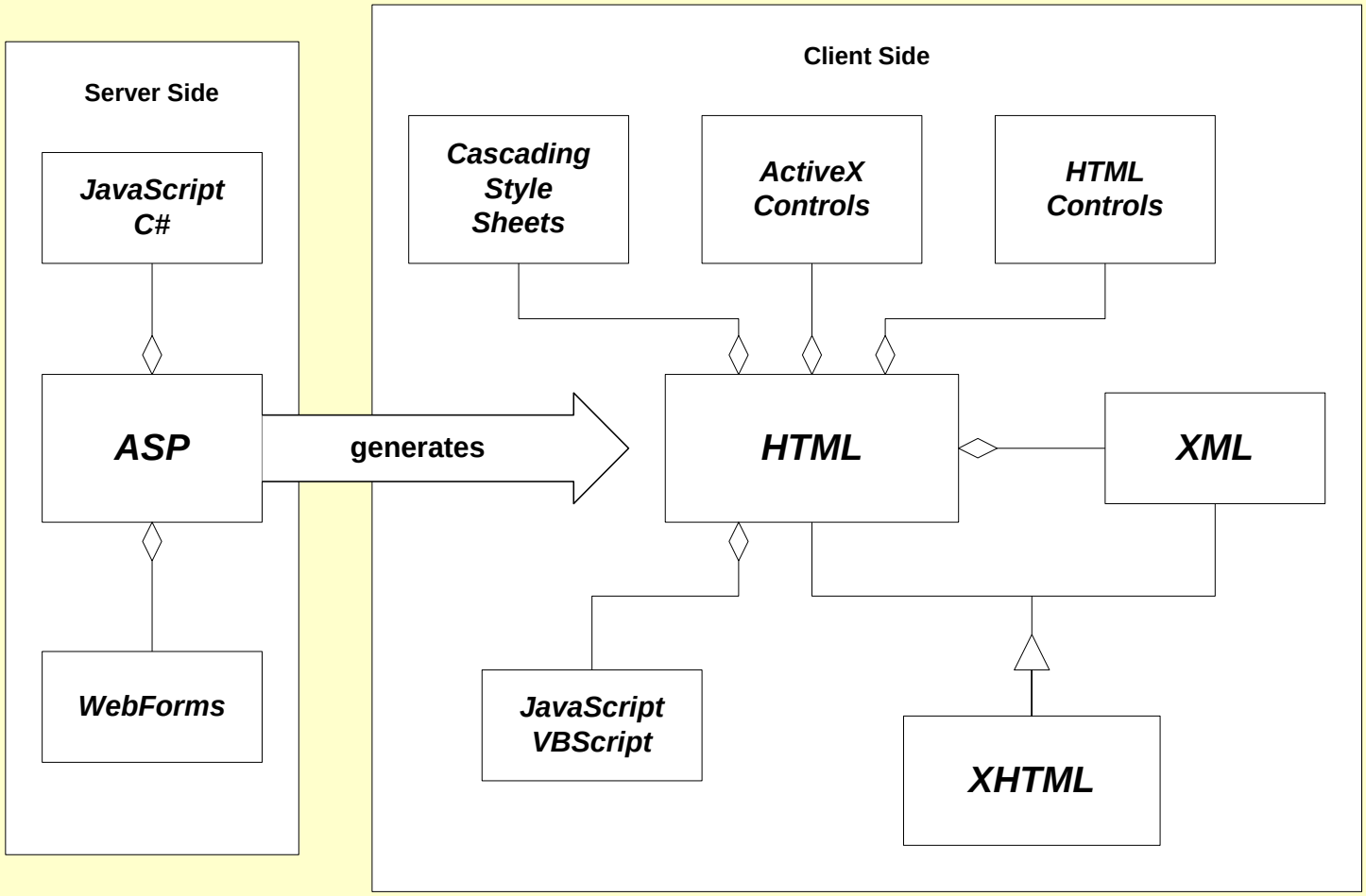
Programming the Web
Server-Side Code

- What is server-side code?
  - Software that runs on the server, not the client
  - Receives input from
    - URL parameters
    - HTML form data
    - Cookies
    - HTTP headers
  - Can access server-side databases, e-mail servers, files, mainframes, etc.
  - Dynamically builds a custom HTML response for a client

Programming the Web
Server-Side Code

- Why server-side code?
  - Accessibility
    - You can reach the Internet from any browser, any device, any time, anywhere
  - Manageability
    - Does not require distribution of application code
    - Easy to change code
  - Security
    - Source code is not exposed
    - Once user is authenticated, can only allow certain actions
  - Scalability
    - Web-based 3-tier architecture can scale out

Web Programming – Language Model



Programming Paradigms
Event-Based Programming

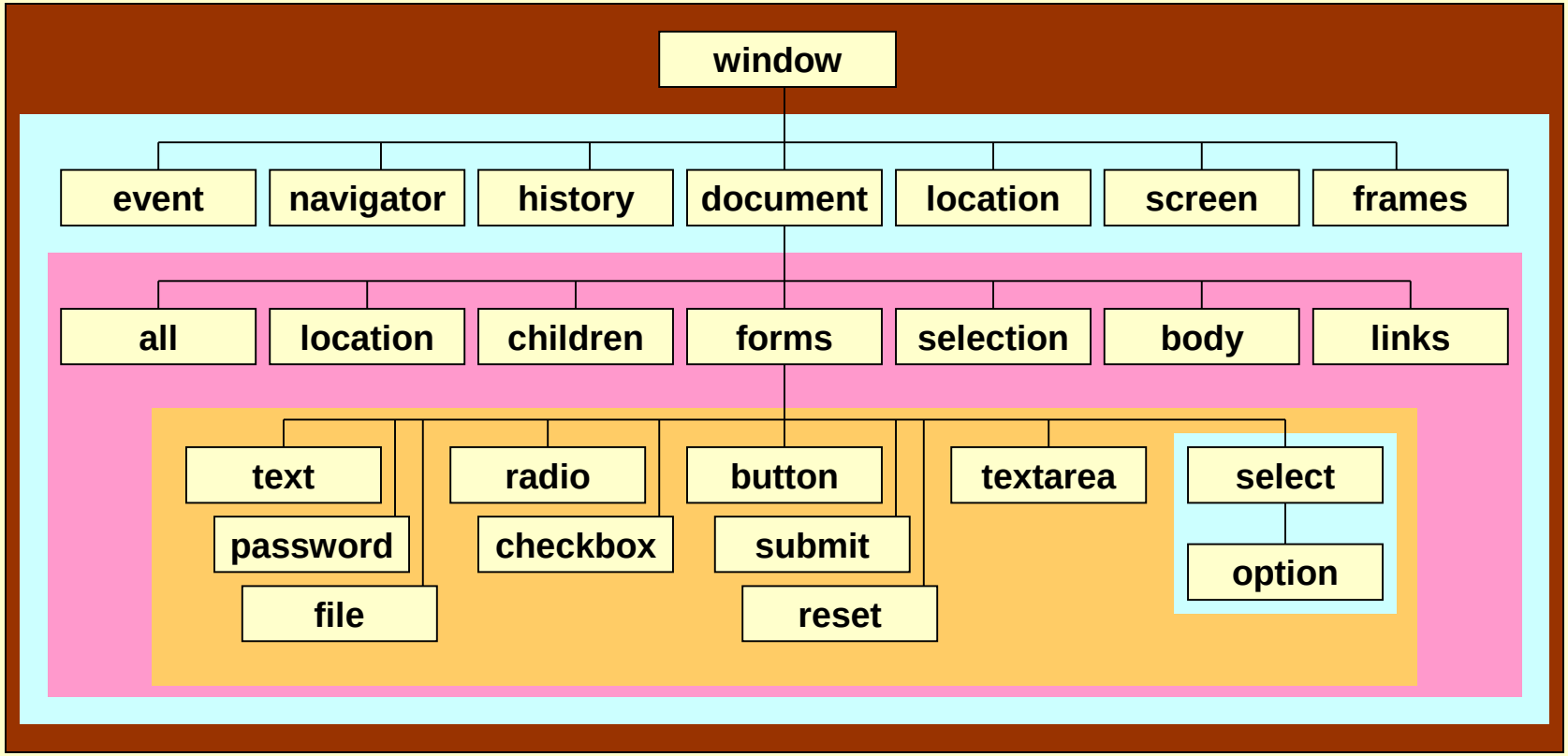
- When something of interest occurs, an event is raised and application-specific code is executed
- Events provide a way for you to hook in your own code into the operation of another system
- Event = callback
- User interfaces are all about events
  - `onClick`, `onMouseOver`, `onMouseMove`…|
- Events can also be based upon time or interactions with the network, operating system, other applications, etc.

Event-Based Programming on Client
Dynamic HTML (DHTML)

- Script is embedded within, or attached to, an HTML page
- Usually written in JavaScript (ECMAScript, JScript) for portability
  - Internet Explorer also supports VBScript and other scripting languages
- Each HTML element becomes an object that has associated events (e.g. `onClick`)
- Script provides code to respond to browser events

Programming the Web
DHTML

- DHTML Document Object Model (DOM)



Server Object Model

- ***Application Object***
  - Data sharing and locking across clients
- ***Request Object***
  - Extracts client data and cookies from HTTP request
- ***Reponse Object***
  - Send cookies or call Write method to place string in HTML output
- ***Server Object***
  - Provides utility methods
- ***Session Object***
  - If browser supports cookies, will maintain data between page loads, as long as session lasts.

Server Side Programming with ASP

- An Active Server Page (ASP) consists of HTML and script.
  - HTML is sent to the client "as-is"
  - Script is executed on a server to dynamically generate more HTML to send to the client.
  - Since it is generated dynamically, ASP can tailor the HTML to the context in which it executes, e.g., based on time, data from client, current server state, etc.

Programming the Web
Active Server Pages (ASP)

- Technology to easily create server-side applications
- ASP pages are written in a scripting language, usually VBScript or Jscript
- An ASP page contains a sequence of static HTML interspersed with server-side code
- ASP script commonly accesses and updates data in a database

Event-Based Programming on Server
ASP.Net

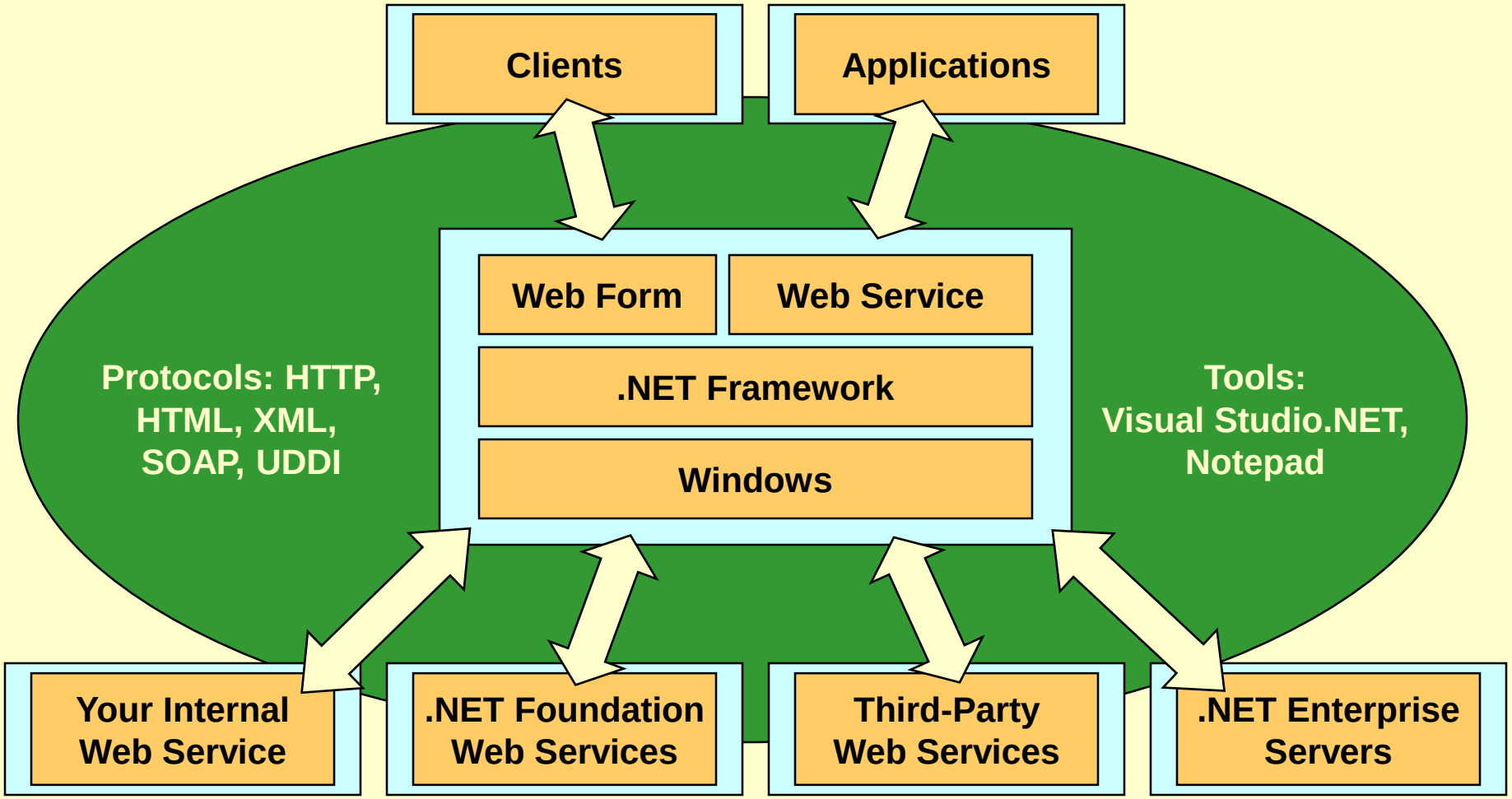
- Pages are constructed from HTML, Web Controls, and C# event handlers.
- The ASP.Net Page processing renders Web Controls on a page into HTML constructs with attached Javascript event handlers.
  - The Javascript handlers post messages back to the server describing the event, which is then handled by C# code on the server.
- The result of the handled event is usually another page sent back to the browser client.

Introduction to .NET
What is .NET?

- A vision
  - Web sites will be joined by Web services
  - New smart devices will join the PC
  - User interfaces will become more adaptable and customizable
  - Enabled by Web standards

Introduction to .NET

The .NET Platform



Common Language Runtime
Assemblies

- Assembly
  - Logical unit of deployment
  - Contains Manifest, Metadata, MSIL and resources
- Manifest
  - Metadata about the components in an assembly (version, types, dependencies, etc.)
- Type Metadata
  - Completely describes all types defined in an assembly: properties, methods, arguments, return values, attributes, base classes, …

Common Language Runtime
Services

- Code management
- Conversion of MSIL to native code
- Loading and execution of managed code
- Creation and management of metadata
- Verification of type safety
- Insertion and execution of security checks
- Memory management and isolation
- Handling exceptions across languages
- Interoperation between .NET Framework objects and COM objects and Win32 DLLs
- Automation of object layout for late binding
- Developer services (profiling, debugging, etc.)

Common Language Runtime
Security

- Evidence-based security (authentication)
- Based on user identity and code identity
- Configurable policies
- Imperative and declarative interfaces

Windows Forms

- Framework for building rich clients
- Built upon .NET Framework, languages
- Rapid Application Development (RAD)
- Visual inheritance
- Anchoring and docking
- Rich set of controls
- Extensible controls
- Data-aware
- Easily hooked into Web Services
- ActiveX support
- Licensing support
- Printing support
- Advanced graphics

Web Forms

- Built with ASP.NET
  - Logical evolution of ASP
  - Similar development model: edit the page and go
- Requires less code
- New programming model
  - Event-driven/server-side controls
  - Rich controls (e.g. data grid, validation)
  - Data binding
  - Controls generate browser-specific code
  - Simplified handling of page state

Web Forms

- Allows separation of UI and business logic
- Uses .NET languages
  - Not just scripting
- Easy to use components
- XCOPY/FTP deployment
- Simple configuration (XML-based)

ADO.NET

- Similar to ADO, but better factored
- Language-neutral data access
- Supports two styles of data access
  - Disconnected
  - Forward-only, read-only access
- Supports data binding
- DataSet: a collection of tables
- Can view and process data relationally (tables) or hierarchically (XML)

Security Issues

- Threats
  - Data integrity
    - code that deletes or modifies data
  - Privacy
    - code that copies confidential data and makes it available to others
  - Denial of service
    - code that consumes all of CPU time or disk memory.
  - Elevation of privilege
    - Code that attempts to gain administrative access

Protections

- Least privilege rule:
  - Use the technology with the fewest capabilities that gets the job done.
- Digital signing
  - Who are you?
- Security zones
  - Trusted and untrusted sites
- Secure sockets layer (SSL)
- Transport layer security (TLS)
- Encryption

Areas of Exploration

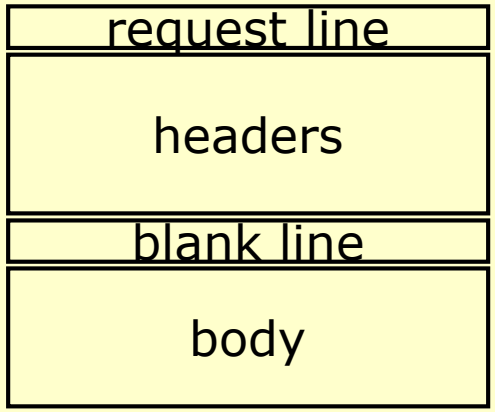
- XML - Universal Data Services
- TVWeb - merger of features
- MathML - Mathematical Markup Language
- RDF - Resouce Description Framework
- Accessibility - for the handicapped
- SMIL - Synchronized Multimedia Integration Language
- Internationalization
- Speech

References

- Introduction to the Web and .Net, Mark Sapossnek, Computer Science, Boston Univ.
  - slides available on www.gotdotnet.com
- World Wide Web Consortium
  - Excellent Tutorial Papers, standards
- XHTML Black Book, Steven Holzner, Coriolis, 2000
  - Very comprehensive treatment of HTML, XHTML, JavaScript
- Inside Dynamic HTML, Scott Issacs, Microsoft Press, 1997
- C# .Net Web Developer's Guide, Turtschi et. al., Syngress, 2002
  - Class text
- Web Developers Virtual Library
  - Excellent set of tutorials
- Class Web Links
  - Web links.htm

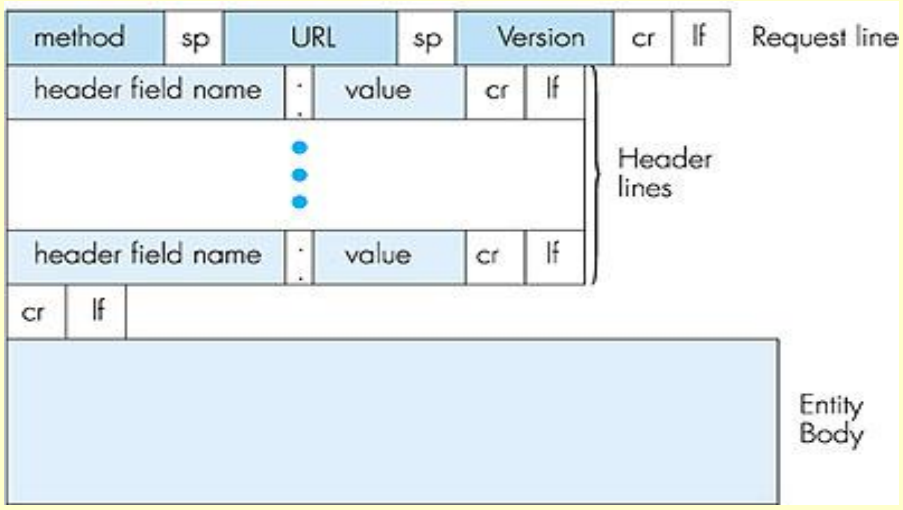
Appendix A
HTTP Message Headers

Request Message



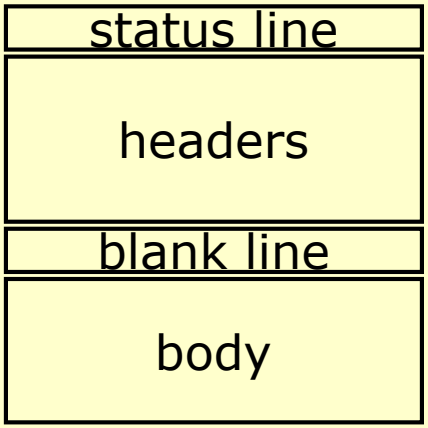
request methods:

DELETE, GET, HEAD, POST, PUT, TRACE

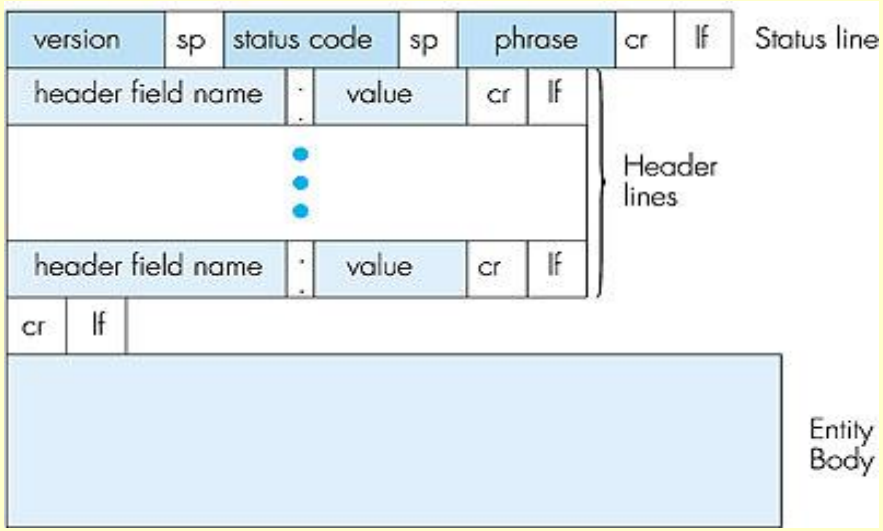



```
GET /pub/index.html HTTP/1.0
Date: Wed, 20 Mar 2002 10:00:02 GMT
Pragma: no-cache
From: amer@udel.edu
User-Agent: Mozilla/4.03
```

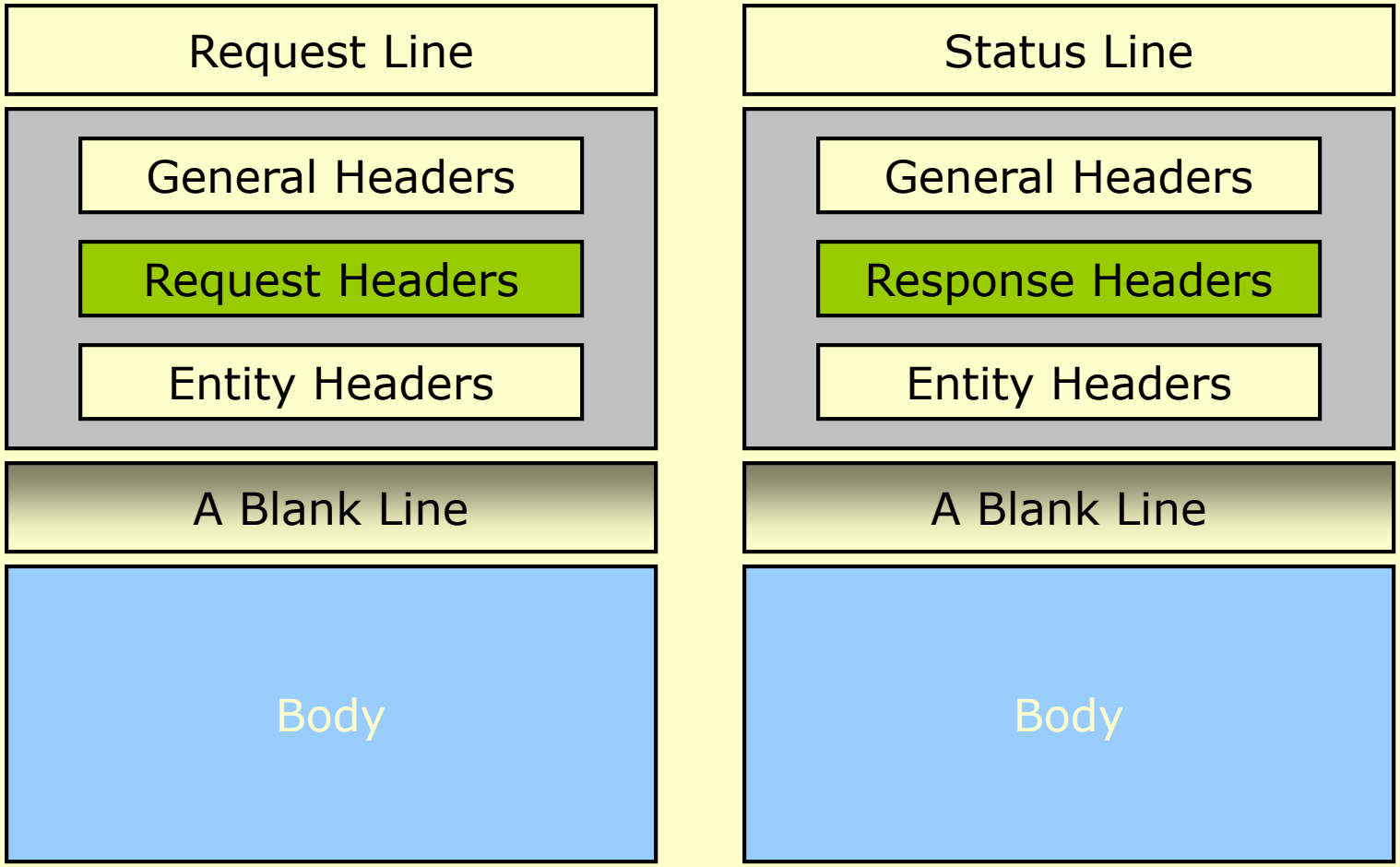
Response Message



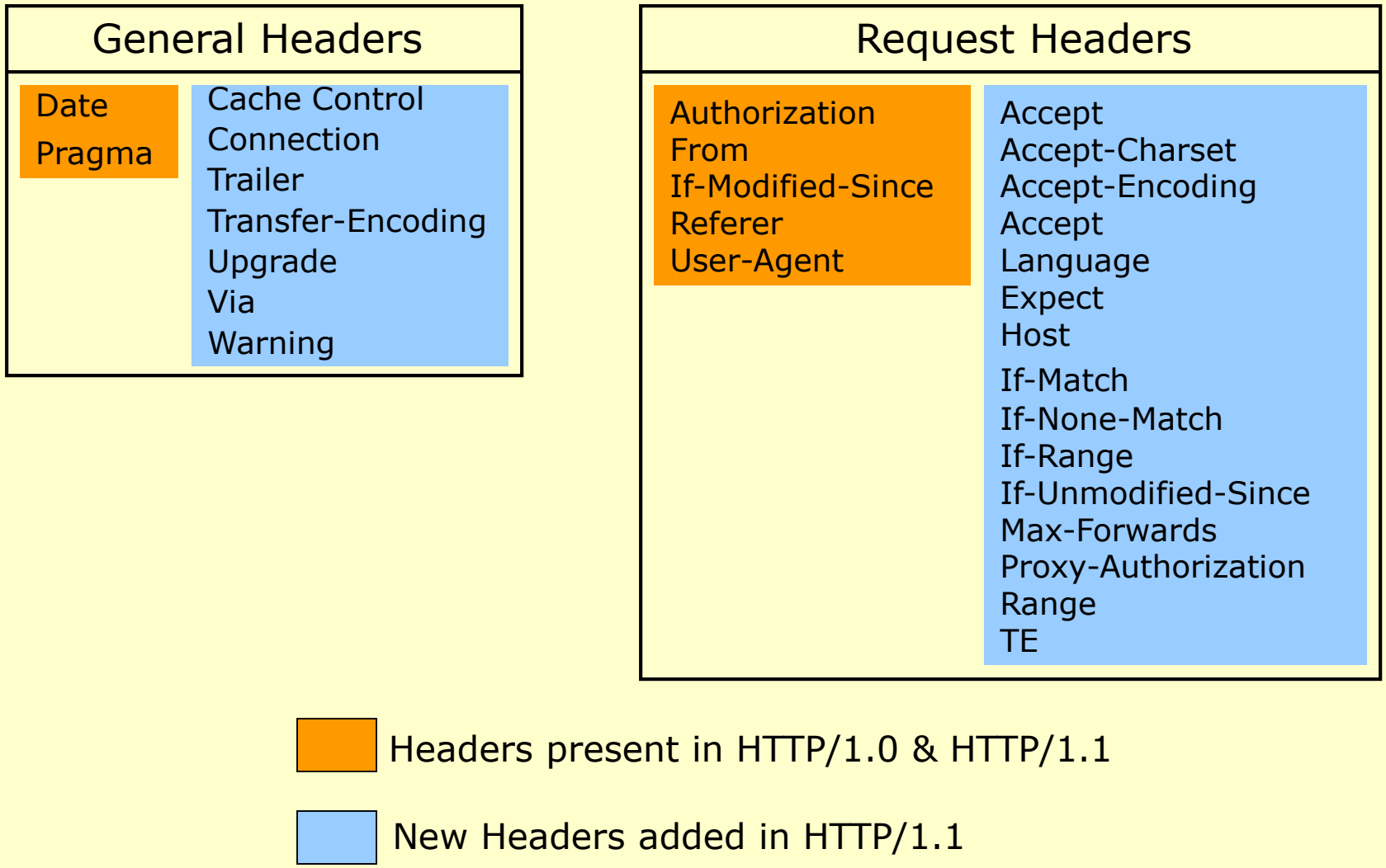

```
HTTP/1.1 200 OK
Date: Tue, 08 Oct 2002 00:31:35 GMT
Server: Apache/1.3.27 tomcat/1.0
Last-Modified: 7Oct2002 23:40:01 GMT
ETag: "20f-6c4b-3da21b51"
Accept-Ranges: bytes
Content-Length: 27723
Keep-Alive: timeout=5, max=300
Connection: Keep-Alive
Content-Type: text/html
```



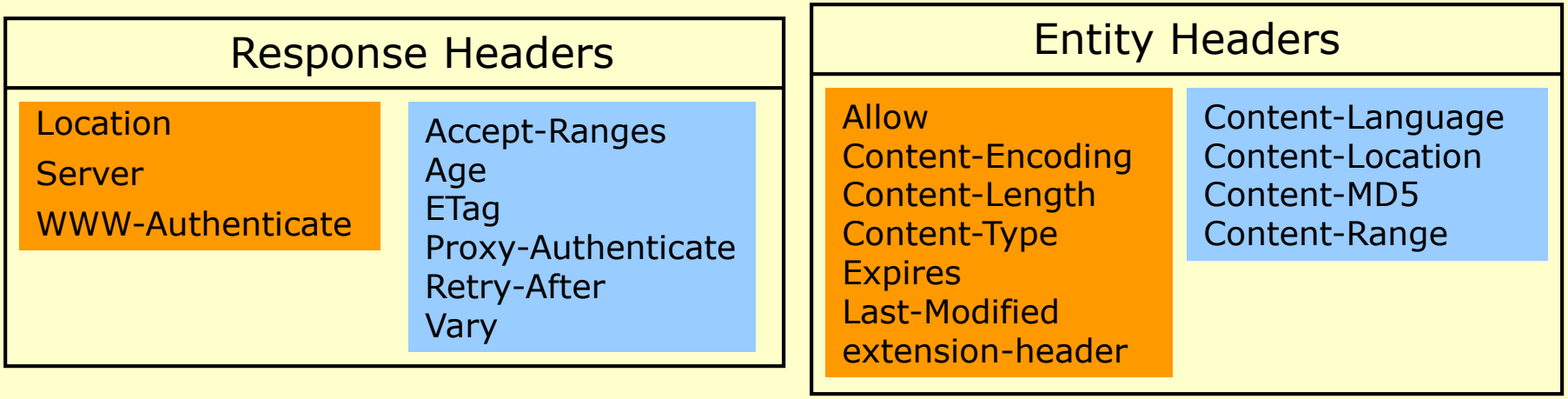
Headers


Request Line
General Headers
Request Headers
Entity Headers
A Blank Line
Body
Status Line
General Headers
Response Headers
Entity Headers
A Blank Line
Body

Headers


General Headers
Date
Pragma
Cache Control
Connection
Trailer
Transfer-Encoding
Upgrade
Via
Warning
Request Headers
Authorization
From
If-Modified-Since
Referer
User-Agent
Accept
Accept-Charset
Accept-Encoding
Accept
Language
Expect
Host
If-Match
If-None-Match
If-Range
If-Unmodified-Since
Max-Forwards
Proxy-Authorization
Range
TE
Headers present in HTTP/1.0 & HTTP/1.1
New Headers added in HTTP/1.1

Headers



Headers present in HTTP/1.0 & HTTP/1.1

New Headers added in HTTP/1.1

End of Presentation