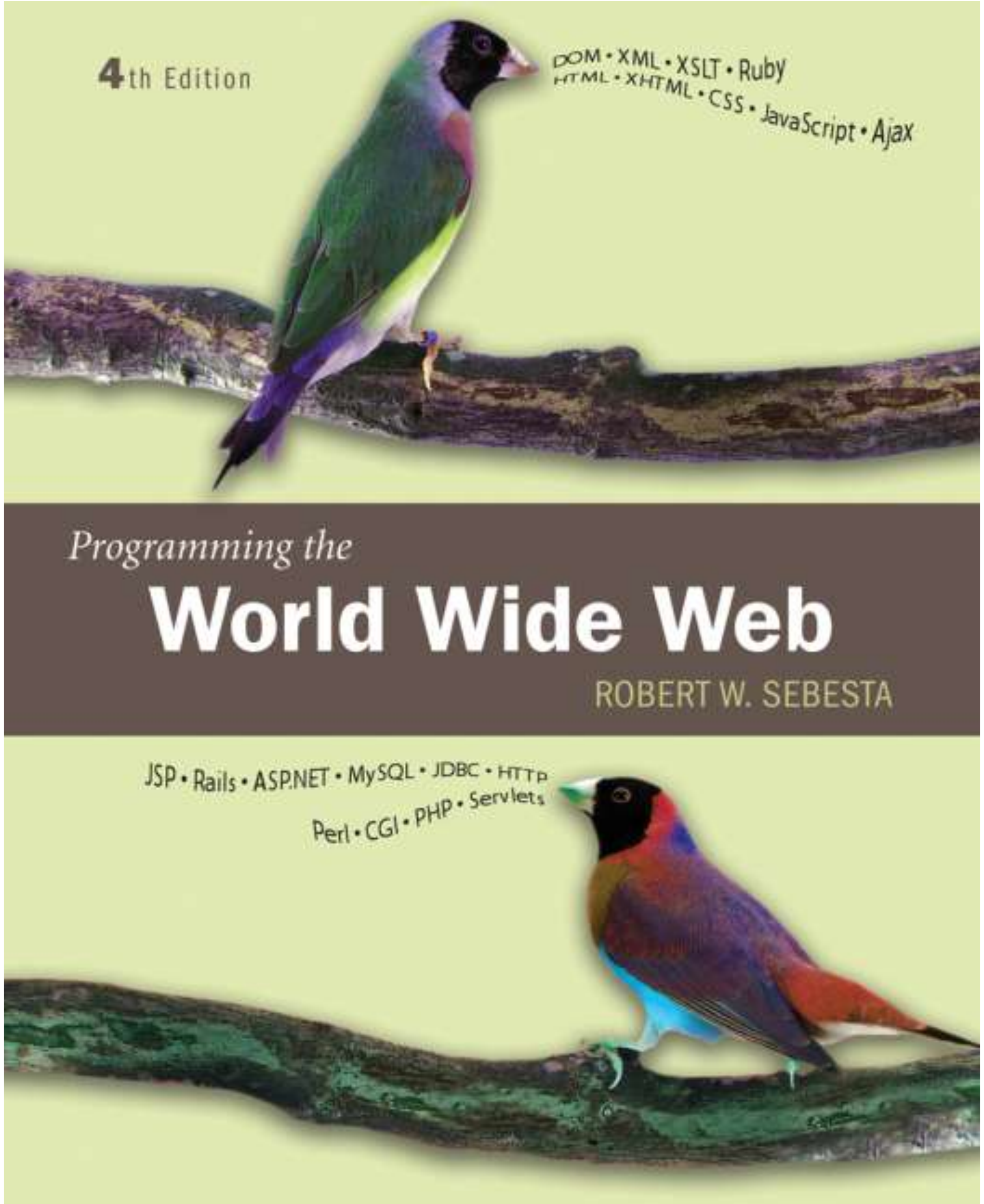

4th Edition
DOM • XML • XSLT • Ruby
HTML • XHTML • CSS • JavaScript • Ajax
Programming the
World Wide Web
ROBERT W. SEBESTA
JSP • Rails • ASP.NET • MySQL • JDBC • HTTP
Perl • CGI • PHP • Servlets

Chapter 1

Fundamentals

1.1 A Brief Intro to the Internet

- **Internet History**
- **Internet Protocols**

1.1 Internet History

- Origins
  - ARPAnet - late 1960s and early 1970s
    - Network reliability
    - For ARPA-funded research organizations
- BITnet, CSnet - late 1970s & early 1980s
    - email and file transfer for other institutions
  - NSFnet - 1986
    - Originally for non-DOD funded places
    - Initially connected five supercomputer centers
    - By 1990, it had replaced ARPAnet for non-military uses
    - Soon became the network for all (by the early 1990s)
  - NSFnet eventually became known as the Internet

1.1 Internet History

- What the Internet is:
    - A world-wide network of computer networks
    - At the lowest level, since 1982, all connections use TCP/IP
    - TCP/IP hides the differences among devices connected to the Internet

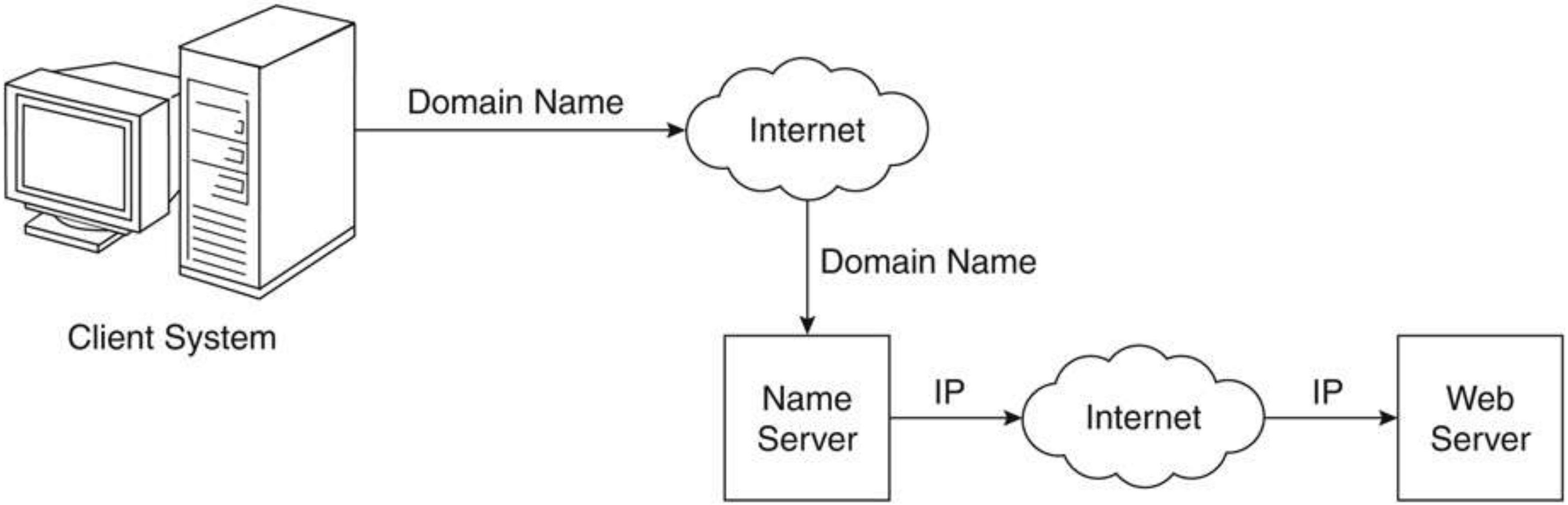
1.1 Internet Protocols

- Internet Protocol (IP) Addresses
  - Every node has a unique numeric address
  - Form: 32-bit binary number
    - New standard, IPv6, has 128 bits (1998)
- Organizations are assigned groups of IPs for their computers
- Problem: By the mid-1980s, several different protocols had been invented and were being used on the Internet, all with different user interfaces (Telnet, FTP, Usenet, mailto)

1.1 Internet Protocols

- Domain names
  - Form: host-name.domain-names
  - First domain is the smallest; last is the largest
  - Last domain specifies the type of organization
  - Fully qualified domain name - the host name and all of the domain names
  - DNS servers - convert fully qualified domain names to IPs

Figure 1.1 Domain name conversion



Client and Server

- Clients and Servers are programs that communicate with each other over the Internet
- A Server runs continuously, waiting to be contacted by a Client
  - Each Server provides certain services
  - Services include providing web pages
- A Client will send a message to a Server requesting the service provided by that server
  - The client will usually provide some information, parameters, with the request

1.2 The World-Wide Web

- A possible solution to the large number of different protocols being used on the Internet
- Origins
    - Tim Berners-Lee at CERN proposed the Web in 1989
        - Purpose: to allow scientists to have access to many databases of scientific work through their own computers
    - Document form: hypertext
    - Pages? Documents? Resources?
        - We’ll call them documents
    - Hypermedia – more than just text – images, sound, etc.

1.2 The World-Wide Web

- **Web or Internet?**
  - **The Web uses one of the protocols, http, that runs on the Internet--there are several others (telnet, mailto, etc.)**

1.3 Web Browsers

- Browsers are clients - always initiate, servers react (although sometimes servers require responses)
- Mosaic - NCSA (Univ. of Illinois), in early 1993
  - First to use a GUI, led to explosion of Web use
  - Initially for X-Windows, under UNIX, but was ported to other platforms by late 1993
- Most requests are for existing documents, using HyperText Transfer Protocol (HTTP)
  - But some requests are for program execution, with the output being returned as a document
- Name some of the Web Browsers?

1.4 Web Servers

- Provide responses to browser requests, either existing documents or dynamically built documents
- Browser-server connection is now maintained through more than one request-response cycle
- All communications between browsers and servers use Hypertext Transfer Protocol (HTTP)
- Port number for HTTP? and HTTPS?

1.4 Web Server Operation

- Web servers run as background processes in the operating system
  - Monitor a communications port on the host, accepting HTTP messages when they appear
- All current Web servers came from either
  1. The original from CERN
  2. The second one, from NCSA
- Most common server now is Apache

1.4 Web Server Operation Details

- The file structure of Web servers have two main directories:

    1. **Document root** (servable documents)
    2. **Server root** (server system software)

- Document root is accessed indirectly by clients
    - Its actual location is set by the server configuration file
    - Requests are mapped to the actual location
- **Virtual document trees** (servable documents from the secondary areas)
- **Virtual hosts (**servers support more than one site on a computer
- **Proxy servers** (Servers serving documents from the document root of other machine)
- Web servers now support other Internet protocols

1.4 Web Server Operation : Apache

- Apache (open source, fast, reliable)
- 3 config files of Apache server:
  - Httpd.conf
  - Srm.conf
  - Access.conf
- Different directives are available in
  http://httpd.apache.org/docs/2.2/mod/quickreference.html
- Basically designed for UNIX systems

1.4 Web Server Operation : IIS

- IIS for windows systems
  - Operation is maintained through a program (IIS snap-in) with a GUI interface
  - Operations of Apache is controlled by configuration files

1.5 URLs

- General form:

  Format: object-address

  - The scheme is often a communications protocol, such as telnet or ftp

- For the http protocol, the object-address is:

  //fully-qualified-domain-name/path-to-document

- For the file protocol, only the doc path is needed

  file://path-to-document

1.5 URLs

- Host name may include a port number, as in zeppo:80 (80 is the default, so this is silly)
- URLs cannot include spaces or any of a collection of other special characters (semicolons, colons, ...)
- The doc path may be abbreviated as a partial path
  - The rest is furnished by the server configuration
- If a special character has to be entered then % with its hexadecimal ASCII code should be included

Eg: if domain name '**RV CE**' has to be specfied then it has to be written as '**RV%20CE' (20 is ASCII code for space)**

1.6 Multipurpose Internet Mail Extensions (MIME)

- Used by the browsers for deciding the format of doc received by the web server
- Originally developed for email
- Attached by the server to the beginning of the document
- Type specifications
  - Form:

    type/subtype
  - Common types are: text, image and video
  - Common subtypes are: plain, html, gif, jpeg, mpeg, quicktime
  - Examples: text/plain, text/html, image/gif, image/jpeg

1.6 Multipurpose Internet Mail Extensions (MIME)

- Server gets type from the requested file name’s suffix (.html implies text/html)
- Browser gets the type explicitly from the server
- Experimental sub types
- Subtype begins with x-
- e.g., video/x-msvideo
- Experimental types require the server to send a helper application or plug-in so the browser can deal with the file

1.7 The HyperText Transfer Protocol

- The protocol used by ALL Web communications
- Request Phase
  - Form:

    HTTP method domain part of URL HTTP ver.

    Header fields

    blank line

    Message body
  - An example of the first line of a request:

    GET /degrees.html HTTP/1.1

1.7 The HyperText Transfer Protocol: Methods

- GET - Fetch a document
- POST - Execute the document, using the data in body
- HEAD - Fetch just the header of the document
- PUT - Store a new document on the server
- DELETE - Remove a document from the server

1.7 HTTP Headers

- Four categories of header fields:

  General, request, response, & entity

- Common request fields:

  Accept: text/plain

  Accept: text/*

  If-Modified_since: date

- Common response fields:

  Content-length: 488

  Content-type: text/html

- Can communicate with HTTP without a browser

  > telnet blanca.uccs.edu http

  GET /respond.html HTTP/1.1

  Host: blanca.uccs.edu connection:keep-alive ---- ------ ------- hello world

1.7 HTTP Response

- Form:

  Status line

  Response header fields

  blank line

  Response body
- Status line format:

  HTTP version   status code   explanation
- Example: HTTP/1.1 200 OK

  (Current version is 1.1)
- Status code is a three-digit number; first digit specifies the general status

  1 => Informational

  2 => Success

  3 => Redirection

  4 => Client error

  5 => Server error
- The header field, Content-type, is required

1.7 HTTP Response Example

```
HTTP/1.1  200  OK
Date: Tues, 18 May 2004 16:45:13 GMT
Server: Apache (Red-Hat/Linux)
Last-modified: Tues, 18 May 2004 16:38:38 GMT
Accept-ranges: bytes
Content-length: 364
Connection: close
Content-type: text/html, charset=ISO-8859-1
```

- Both request headers and response headers must be followed by a blank line

1.7 Status code Example

- Example for status code are
  - ❖404 --- file not found
  - ❖200 --- OK(Successful Request)
  - ❖500 --- Server encountered a problem
  - ❖202- Accepted for processing

1.8 Security

- Both Web **Browsers** and **Servers** are unsafe !!!??
- Security issues are
  1. **Privacy:** it shouldn't be possible to steal the data while transmitting
  2. **Integrity:** it shouldn't be possible to modify the data while transmitting
  3. **Authentication:** it should be possible for both ends to be certain of each others identity
  4. **Nonrepudiation:** it should be possible to legally prove that the message was actually sent and received
- **Encryption** is the tool for resolving security issues
  1. Symmetric key encryption
  2. Public key encryption

1.8 Security

- Some more security problems are
  1. DoS
  2. Viruses
  3. Worms