

Lecture 13

Web Fundamentals

1

Today's Topic

- Networking and Internet
 - LAN
- Web Organization
 - Web pages
 - Web resources
 - Web client
 - Web Server
- HTTP Protocol
- HTML & HTML Forms

2

Networking and Internet

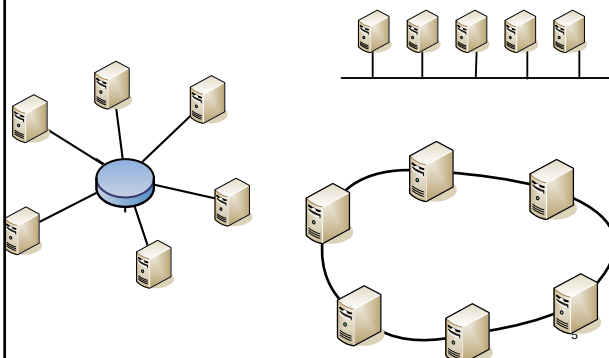
3

LAN (Local Area Network)

- Local area network connects "local" computing devices to share data or information
 - Computers, printers, main-frame, routers, ...
- **Topology**: defines physical connectivity between computing devices
 - Star, ring, bus, tree, ...
- Computing devices talk to each other using **LAN protocols**
 - Protocol: The 'language' spoken between devices to help them exchange information.
 - A formal description of message formats and the rules that two computers must follow in order to exchange those messages.
 - Apple Talk, NetBUI, MS-Network (Network Neighborhood), ...
 - Each computing device must have a unique identifier such as MAC (Media Access Control address)
 - Each modern NIC has a unique MAC address (e.g. 00:0D:60:B1:39:7B)
 - <http://standards.ieee.org/reqauth/oui/index.shtml>

4

Network Topology



Internet

- Inter-Net
 - Inter-connected LAN (local area network) through WAN (wide-area network)
 - Nearly no centralized control or management
 - Network segments are interconnected through **routers**
 - Routers are dedicated computers managing packets or data
 - TCP/IP is the universal **data transport protocol** on the Internet
 - Actual format or content is left to higher-level protocols (next), like the web

6

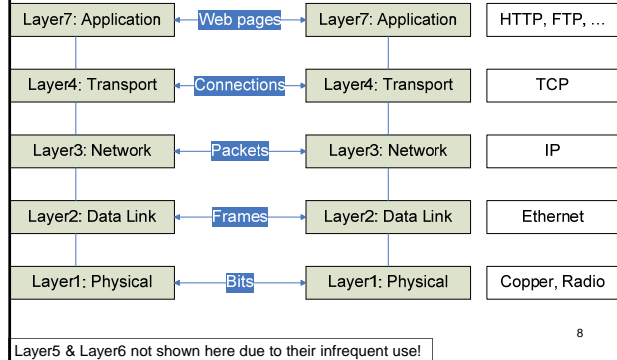
Source MIT 1.264 lecture notes

Network Communication

- Most applications are at layer-7 that uses high-level of abstraction to communicate with other devices
 - I want to talk to Mr. Smith
- Logically, Each layer talks directly to the same layer on the other side.
 - Smith says "Hi! There."
- Physically, data is moved down to lower levels and "wrapped up" with additional information (headers)
 - Message for Smith@140.118.105.12: "I want to talk to Mr. Smith"
- The layered model
 - Pro: each layer is 'unaware' of other layers → change of networking device requires little effort
 - Con: each layer introduces some "latency" or processing overhead

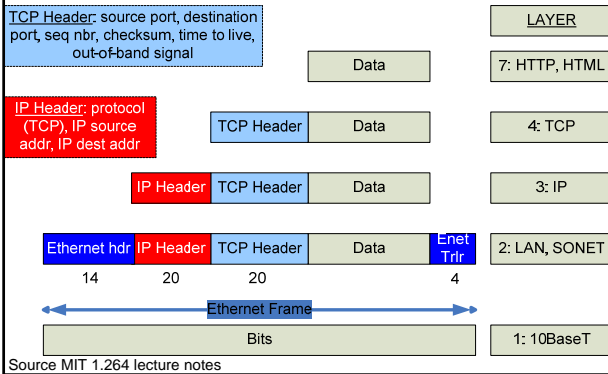
7

OSI Layered Network Model



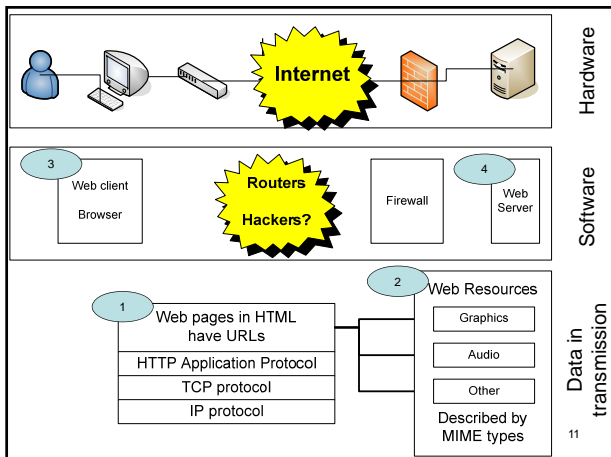
8

TCP/IP Layering



Web Organization

10



11

Web Organization (1) Web Pages

- Web page is the basic unit of organization
 - Simple pages retrieved in a single operation
 - Compound pages (typically text and graphics) take multiple passes (multiple TCP/IP connections unless 'Keep-Alive' is set)
 - Give me page, give me graph1, give me graph2, ...
 - Pages have hypertext
 - Links to other pages, implemented via embedded URLs
 - Pages are described using Hypertext Markup Language (**HTML**)
 - High level document description language
 - Specifies **structure** but not appearance of document
 - Defines sections, such as "level 1 header", "list", "emphasized text"
 - Browser handles rendering of page on client machines

12

Web Organization (2) Web Resources

- Each web resource (document) has a type
 - The type is described as "MIME" (Multi-purpose Internet Mail Extension) types
 - Some types (e.g. HTML) are rendered directly by web browsers
 - Some types (video, flash,...) are displayed via "plug-ins"
 - Web type system is extensible. New types are easy to accommodate.
 - Define new MIME type in server
 - Browser will prompt for helper application if MIME type unrecognized
 - Browser can suggest Web site from which to download helper app
 - Download and install helper app, and use the new MIME type
 - Web server extensibility: programs to generate documents
 - URLs can point at programs as well as pages
 - Programs can be simple (time of day) or complex (database, analysis)
 - Java, Perl, C++, Java Server Pages, Visual Basic, Active Server Pages, ...
 - Applications vary from simple (time of day) to complex database and analysis

13

MIME type examples

- | | |
|----------------------------|------------|
| • application/msword | Word |
| • application/pdf | Acrobat |
| • application/vnd.ms-excel | Excel |
| • application/zip | Zip file |
| • audio/basic | .au, .snd |
| • audio/x-wav | MS audio |
| • image/gif | GIF |
| • image/jpeg | JPEG |
| • text/plain | Plain text |
| • text/html | HTML |
| • video/mpeg | Video |

See <http://www.rfc-editor.org> for current list of HTTP, MIME, other Internet specifications

14

Web Organization (3) Web Clients

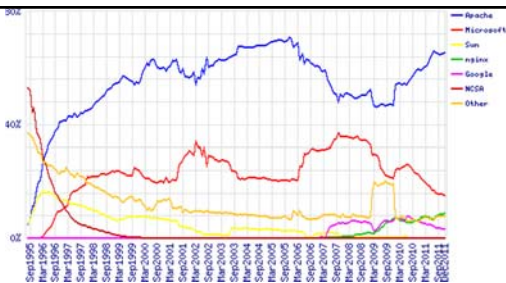
- Most web clients are web browsers
 - Internet Explorer, Mozilla Firefox, Opera, ...
 - Web browsers request web pages (through URL), render the gotten pages, and present it to the user
- Web clients can also be autonomous applications
 - Web spider / robots → search engines
- Standard (application) protocol
 - HTTP (HyperText Transfer Protocol), current version 1.1
 - Four phases
 - Open Connection (e.g. Hello there!)
 - Request (e.g. Please give me a big mac!)
 - Response (e.g. Here comes your big mac!)
 - Close Connection (e.g. Thank you and good bye!)

15

Web Organization (4) Web Servers

- Web servers "serve" web content and send out data using HTTP protocol
 - The content can be static content (HTML pages, images, video files, ...)
 - The content can also be generated dynamically based on some header info in the HTTP request
 - Two of the most popular web server software: Apache & Microsoft IIS
- A world-wide 'survey' can be found at
 - http://news.netcraft.com/archives/web_server_survey.html
 - Netcraft can also be used to make a site report, including the web server software, last reboot, network owner, ...

16

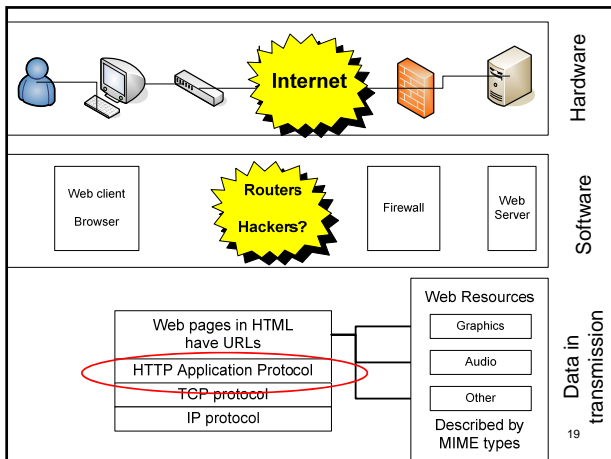


Developer	November 2011	Percent	December 2011	Percent	Change
Apache	341,880,662	65.00%	362,267,922	65.22%	0.22
Microsoft	81,261,099	15.45%	82,521,809	14.86%	-0.59
nginx	44,731,780	8.50%	49,143,289	8.85%	0.34
Google	17,749,748	3.37%	18,464,148	3.32%	-0.05

Source: www.netcraft.com

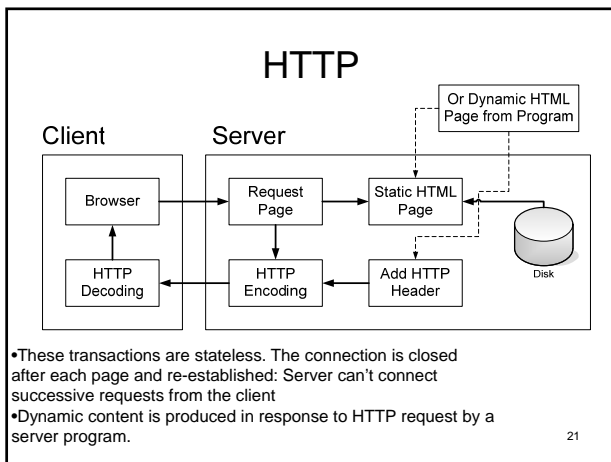
18

HTTP Protocol



HTTP Protocol

1. **Open connection**
 - Connection is opened by giving URL
 - URL: Uniform Resource Locator (e.g. http://www.ntust.edu.tw)
2. **Request**
 - Send request:
 1. Request method (GET, POST,..., may have additional data)
 2. URL
 3. HTTP version number
 4. Header Information, terminated with a blank line
3. **Response: Server processes the request and sends:**
 - HTTP protocol version and status code
 - Header information, terminated by blank line
 - Text (data)
4. **Close connection**



HTTP Request Example

- Telnet yo-1.ct.ntust.edu.tw 80
 - Here we use telnet to establish connection (instead of using web browsers)
 - The following is an example of HTTP request sent from browsers to servers.

1. Request	2. URL	3. Protocol	
	GET /index.html	HTTP/1.0	
4. Header	Host: yo-1.ct.ntust.edu.tw (required)		
	Accept: text/html, text/plain, image/jpeg, */*		
5. Data			

HTTP Requests (browser to server)

- **GET**: Requests specified document (used to 'post' too)
- **HEAD**: Requests only header of specified document
- **POST**: Requests that server accept data from browser and generate dynamic content
- **OPTIONS**: Get server and access options
- **TRACE**: Used in debugging
- **PUT**: Replace server document with data from browser
- **DELETE**: Delete specified document on server

- **GET** is the most frequent request from browsers
- **POST** used for processing fill-out forms
- **HEAD** is used by search engines to check for live pages
- No security or authentication in HTTP ('Basic' sends password in clear).
 - Use Secure Socket Layer (SSL) to encrypt your Web exchanges

HTTP Headers (browser to server)

- **Accept**: MIME types accepted by client (multiple allowed)
- **Connection**: Connection type for client (keep-alive, close)
- **Cookie**: Return previous cookie to server (data exchange)
- **From**: Email address of user (sent only by crawlers)
- **Host**: Original host requested (forward, multiple names)
- **If-Modified-Since**: Used to reduce fetching of docs browser already has
- **Referer**: URL of last document client displayed
- **User-Agent**: Name and version of client software (browser)

HTTP Response Example

1: Protocol	2: Status code
HTTP/1.1 200 OK	
3: Header	Date: Wed, 04 May 2005 14:33:29 GMT Server: Apache Last-Modified: Mon, 31 Jan 2005 09:53:17 GMT ETag: "16ec7-a9-fc4c929b" Accept-Ranges: bytes Content-Length: 169 Content-Type: text/html; charset=null
4: Data	<HTML> . . . </HTML>

25

HTTP Response Status Code

- **100-101**: Informational response; client should respond with some other action (continue, new protocol)
- **200-206**: Request was successful
- **300-307**: Document has moved; indicate new address
- **400-417**: Client error, such as unauthorized request
- **500-505**: Server error
- Examples:
 - 200 OK
 - 404 Not found
 - 500 Internal server error

26

HTTP Response Header

- **Allow**: Requests allowed, such as GET (400s)
- **Content-Length**: Length in bytes of data to be returned
 - (used for 'keep alive' connections)
- **Content-Type**: MIME type of returned data
- **Expires**: Date at which document expires
- **Last-Modified**: Date at which document was last modified
- **Location**: New document address (with 300 status)
- **Set-Cookie**: Gives browser a 'cookie'
 - http://en.wikipedia.org/wiki/HTTP_cookie

27

HTML

Good tutorial:

<http://www.w3schools.com/html/>

28

HTML Example

```
<tag> ... </tag>: describes structures
This page has:
  <html> ... </html>: A HTML document
  <head> ... </head>: Document header
  <body> ... </body>: Document body
```

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=big5">
  <title>My First WebPage</title>
</head>
<body>
  <h1> Hello World! </h1>
</body>
</html>
```

[Link](#)

29

HTML Document

- Tags (<...>) are hints to web browsers and are usually hidden from users, and are often in pairs
 - In XHTML, tags are required to be in pairs.
- Some tags have attributes
 - E.g. ` ABC `
- HTML documents are enclosed by: `<html>` and `</html>`
 - Two sections: **head** and **body**
 - **head** has identifying information not displayed.
 - **body** is displayed, with formatting:
 - `<p>`: Paragraph
 - `<h1>` through `<h6>`: Header level 1 through 6.
 - `
`: line break or carriage return
 - `<a>`: Anchor, placed around text or images; used for hyperlinks

30

Tag Examples (1/2)

- Control appearance of page
 - Not as precise as MS Word or other editors
 - Intended for pages that can be viewed on machines with very different graphics capabilities. Tags discourage specific assumptions.

Opening tag	Closing tag	Definition
<code></code>	<code></code>	Emphasis (often <i>italic</i>)
<code></code>	<code></code>	Strong emphasis (often bold)
<code><var></code>	<code></var></code>	Variable
<code><cite></code>	<code></cite></code>	Citation (of an article or book)
<code><code></code>	<code></code></code>	Computer code (tags ignored)

31

Tag Examples (2/2)

- Paragraph format: `<h2 align="center"> text </h2>`
- Preformatted text: `<pre> (lines up columns exactly, etc.) </pre>`
- Horizontal lines: `<hr width="80%" align="left">`
- Lists: can be multi-level, etc.
 - Ordered (numbered) ``
 - Unordered (bulleted) ``
 - Definition (indented) `<dl>`
- Tables: `<table>`, `<caption>`, `<tr>` (row), etc.
- Font size and color: ``, ``
- Special characters: `>`; is >

Note: The blue text shown are related to the appearance, and these are no longer recommended. Use [CSS \(Cascading Style Sheet\)](#) instead.

32

HTML Forms

- Used as front ends to server programs
 - Active Server Pages, Java Server Pages, PHP, servlets, ...
- Forms are user interface [controls](#) to collect data from user and transmit it to the server application program
- Forms are most common user interface method
 - Forms are placed on Web pages that can also have other elements
 - JavaScript (totally different than Java language!) can be used with HTML forms to enhance their operation
 - Java applets provide a richer user interface, are more complex and have security features. Not used much.
 - All of these run on the [browser](#) and are user interface components

33

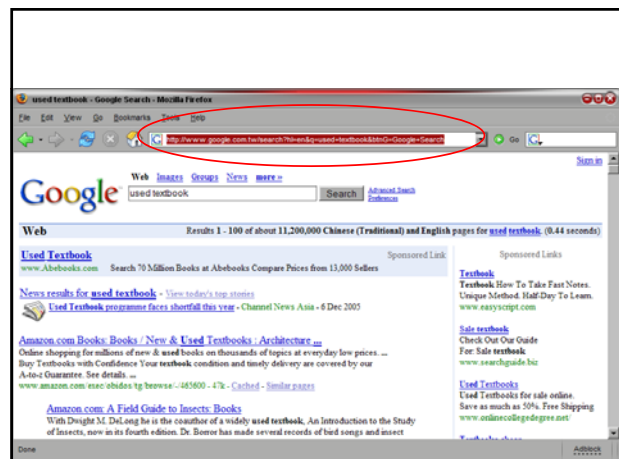


34

How HTML forms transmit data

- Forms allow a series of controls to be placed on the page
 - Each control has a [name](#) and a [value](#)
 - Entire form is associated with the URL of a server side program that will process the input data
 - Form data is sent when user presses 'Submit' button (control)
 - Data is sent to URL as string of form:
 - `Name1=Value1&Name2=Value2&...NameN=ValueN`
 - If data is sent with HTTP GET command, it is appended to end of GET string after a `?`:
 - `GET /Index.html?Name1=Value1...`
 - If data is sent with HTTP PUT command, it is sent after the blank line as the original string
 - Server programs (ASP, JSP, servlet, etc.) have methods to extract the data from the string and use it in the program
- Example: www.google.com

35



HTML 4.0 tags for Forms

Tag	Definition
<form>	Start a form
<input type=	
text	Single line of text entry
password	Single line password entry
file	File to upload, with "Browse" button
checkbox	Checkbox
radio	Radio button (option box)
image	A graphical submit button
hidden	Track user, store predefined inputs
submit	Submit button for form
reset	Button to restore default values
<select>	List box or combo box
<option>	Item in scrolling list or popup menu
<textarea>	Start multiple-line text entry field

Example (1/2)

```
<html>
<body>
<form>
  text input:      <input type=text> <BR>
  password input: <input type=password> <BR>
  file:           <input type=file> <BR>
  checkbox:       <input type=checkbox> <BR>
  radio:          <input type=radio> <BR>
  hidden:         <input type=hidden> <BR>
  image:          <input type=image src="buy.gif"> <BR>
  submit:         <input type=submit> <BR>
  reset:          <input type=reset> <BR>
</form>
</body>
</html>
```

[LINK](#)

In this example, a single form with several controls are created. However, these controls are useless until we define an ACTION

38

Example (2/2)

```
<html>
<body>
<form action="http://somesite.com/prog/adduser" method="post">

  First name: <INPUT type="text" name="firstname"> <br>
  Last name:  <INPUT type="text" name="lastname"> <br>
  Email:     <INPUT type="text" name="email"> <br>
  <input type="radio" name="sex" value="Male"> Male <br>
  <input type="radio" name="sex" value="Female"> Female <br>
  <input type="submit" value="Send">
  <input type="reset">

</form>
</body>
</html>
```

[LINK](#)

<http://www.w3.org/TR/REC-html40/interact/forms.html#h-17.3>

39

Important Attributes

```
<form action="http://someURL.php" method="post">
```

- **action:** defines the processing page to handle the data on the form filled by users
- **method:** defines how the data is sent to the processing page

```
<input type="text" name="firstname">
```

- **type:** defines what kind of control is used to get data
- **name:** defines an identifier for the action form to get data

40

ASP.Net

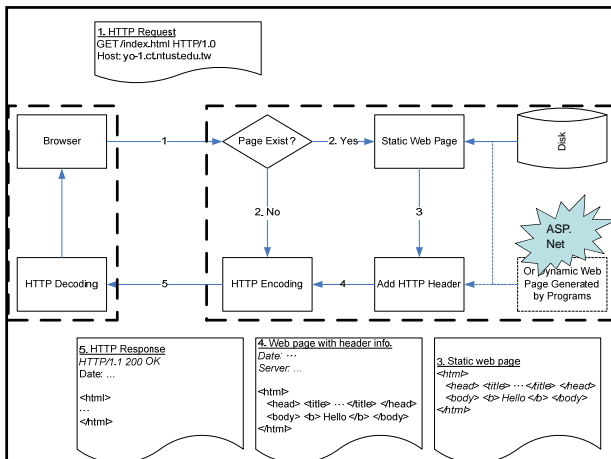
Active Server Pages .Net

What is ASP.NET

- ASP: Active server pages
- .NET: .NET Framework
- ASP.NET is a server-side programming environment to create and run dynamic, interactive Web server applications and services.
 - Server-side scripting means computer codes (scripts) are executed on the web server.
 - Client-side scripting means computer codes are executed on the web browser.

<http://authors.aspalliance.com/aspxtreme/webapps/aspnetarchitecture.aspx>

<http://www.west-wind.com/presentations/howaspnetworks/howaspnetworks.asp>



ASP.NET (2/2)

- Web server requirement for ASP.NET
 - Microsoft IIS
 - .NET Framework Runtime installed
 - Windows 2000+
- Web browser requirement
 - Since ASP.NET are server-side programs, it has nothing to do with the browser. However, the dynamically generated page still need to conform to W3C specifications.
- We are going to write very little ASP.NET code here. Most ASP.NET pages are generated “automagically” by Expression Web.

Summary

- Networking & Internet
- Web organization
 - Web pages
 - Web resources
 - Web client
 - Web Server
- HTTP Protocol
- HTML & HTML Forms

45

Assignment #7 (1/2)

- Due: 12/28/2011
- Assuming everyone has completed assignment #6 (thus we’ve completed a 2-tier application)
- In this assignment, we’re going to build a static website (without database connectivity) using Microsoft Expression Web.

<http://www.microsoft.com/expression/try-it/Default.aspx>

Assignment #7 (2/2)

- Please use Microsoft Expression Web’s template to create a web site for your company.
 - Please assume you’re starting up a business and try to make a good corporate image on the web. (in other words, make it pretty!)
 - Please include all web pages for your core function. For now, just leave them as blank or “under construction”.
 - Note: TRY TO BE CURIOUS
 - Try observe HTML codes generated by Expression Web to familiarize yourself with HTML. If you are already familiar with HTML, try to predict what kind of tags Expression Web will generate for you.
 - Try to be curious about the functions & formats provided by Expression Web.