Web Technologies

Bachelor of Computer Applications

III Year I Semester

G Prembihari Saran

Berhampur University

Unit 1: Web Essentials - Clients, Servers and Communication

The Internet and Basic Internet Protocols

What is the Internet?

The Internet is a **global network of interconnected computer networks** that communicate using standardized protocols. It represents the major example of a Wide Area Network (WAN) that connects billions of computers globally. The Internet serves as a medium for communication and interaction in interconnected networks, making information constantly and instantly available to anyone with a connection.

Basic Internet Protocols

Protocol Definition: Protocols are agreed formats for transmitting data between devices. The protocol determines:

- The error checking required
- Data compression method used
- The way the end of a message is signaled
- The way the device indicates that it has received the message

Key Internet Protocols:

1. **TCP/IP (Transmission Control Protocol/Internet Protocol)**
    - **TCP**: Controls disassembly of message into packets at the origin and reassembles at the destination
    - **IP**: Specifies the addressing details for each packet. Each packet is labeled with its origin and destination
    - Developed by Vincent Cerf and Robert Kahn
2. **HTTP (Hypertext Transfer Protocol)**
    - Communication protocol used by the Internet to transfer hypertext documents
    - Developed by Tim Berners-Lee in 1991
    - Designed to transfer pages between machines
3. **FTP (File Transfer Protocol)**
    - Used for transferring files over the internet
4. **UDP (User Datagram Protocol)**
    - Faster but less reliable than TCP

5. **Email Protocols**
    - **IMAP (Internet Message Access Protocol)**
    - **POP (Post Office Protocol)**

How Internet Protocols Work

Step-by-Step Process:

1. **Dividing Data into Packets**: When information is sent over the internet, IP splits it into small parts called packets. Each packet contains a piece of the data and the address of where it needs to go
2. **Addressing**: Every device connected to the internet has its own IP address to identify where data is being sent from and where it should be delivered
3. **Routing the Packets**: Packets travel through routers that direct them toward the correct destination
4. **Reassemble the Data**: Once all packets arrive at the destination, they are put back together to recreate the original message
5. **Handling Missing Packets**: If some packets don't arrive, the system can request that they be sent again

World Wide Web (WWW)

The World Wide Web is built on top of the Internet infrastructure and uses HTTP as its primary communication protocol. It provides a system for accessing and sharing documents and resources across the global network.

Client-Server Architecture

Definition and Components

Client: The software that resides on the remote machine, communicates with the server, fetches information, processes it, and displays it on the remote machine.

- Initiates contact with server ("speaks first")
- Typically requests service from server
- For web: client is implemented in a browser

Server: The software that distributes information and the machine where the information and software reside.

- Provides requested service to client
- Example: Web server sends requested web page

Web Server: Software that delivers web pages and other documents to browsers using the HTTP protocol.

Client-Server Paradigm

The Client-Server paradigm is the most prevalent model for distributed computing protocols. It is service-oriented and employs a request-response protocol. The primary idea is having a central repository of information that you want to distribute on demand to a set of people or machines.

HTTP Request and Response Messages

HTTP Message Structure

HTTP provides a way for communication between a client and server. An HTTP message can be a **request** sent by the client to the server or a **response** sent by the server to the client.

HTTP Request Message

Structure of HTTP Request:

- **Start Line**: Contains the HTTP method, URI, and HTTP version
- **Headers**: Provide additional information about the request
- **Message Body**: Contains data being sent (optional)

Example HTTP Request:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
```

Common HTTP Methods:

- **GET**: Retrieves data from server, adds data to URL
- **POST**: Sends data in the request body

HTTP Response Message

Structure of HTTP Response:

- **Status Line**: Contains HTTP version, status code, and reason phrase
- **Headers**: Provide metadata about the response
- **Message Body**: Contains the requested content

Example HTTP Response:

```
HTTP/1.1 200 OK
Content-Type: text/html
```

```
Content-Length: 154

<html>
<body>
<h1>Welcome to Example.com!</h1>
</body>
</html>
```

Status Line Components:

- **Protocol**: HTTP version (e.g., HTTP/1.1)
- **Status Code**: Three-digit number indicating result (e.g., 200, 404, 302)
- **Status Text**: Brief description of status code (e.g., OK, Not Found)

Request-Response Cycle

The communication follows this cycle:

1. **Client Sends Request**: Browser sends HTTP request to server specifying desired resource
2. **Server Processes Request**: Server receives request, processes it (queries databases, accesses files), and prepares response
3. **Server Sends Response**: Server responds with status code, headers, and requested content
4. **Client Receives Response**: Browser processes response and displays content

Web Clients and Web Servers

Web Browsers (Clients)

User agents for the Web include:

- Internet Explorer
- Firefox
- Chrome
- Safari

Web Servers

Popular web servers include:

- **Apache** (public domain)
- **Microsoft Internet Information Server (IIS)**

Case Study: Complete Web Communication

Life Cycle of an HTTP Request:

1. **Client Initiates Request**: Process begins when client sends HTTP request (clicking link, submitting form, API call)
2. **DNS Resolution**: Domain names are resolved to IP addresses via DNS
3. **TCP Connection**: Browser establishes TCP connection with server
4. **Request Sent**: HTTP request with method, headers, and body sent to server
5. **Server Processing**: Server processes request and prepares response
6. **Response Sent**: Server responds with status code, headers, and data
7. **Client Processing**: Browser processes response and renders content
8. **Connection Closure**: Connection closed or kept alive for reuse

Unit 2: Introduction to HTML

What is HTML?

HTML (HyperText Markup Language) is the standard markup language for creating web pages. It describes the structure of a web page using elements and tags, allowing for the display of text, images, links, and multimedia content.

HTML Domains and Applications

HTML is used in over 95% of all websites today, making it essential for modern web development. It serves as the foundation for:

- Static websites
- Dynamic web applications
- Mobile web interfaces
- Email templates
- Documentation systems

Basic Structure of an HTML Document

Standard HTML Document Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document Title</title>
</head>
<body>
    <!-- Visible content goes here -->
    <h1>Main Heading</h1>
    <p>Paragraph content</p>
</body>
</html>
```

Essential HTML Document Components

1. **<!DOCTYPE html>**: Declares the document as HTML5
2. **<html>**: Root element containing all HTML content
3. **<head>**: Contains metadata not displayed on webpage
4. **<title>**: Defines document title shown in browser tab
5. **<body>**: Contains all visible content of the webpage

Creating an HTML Document

Steps to Create HTML Document:

1. Create a new file with `.html` extension
2. Add the DOCTYPE declaration
3. Structure the document with html, head, and body tags
4. Add meta tags for character encoding and viewport
5. Include a descriptive title
6. Add content within the body section

Markup Tags and Elements

Understanding Tags vs Elements

- **Tag**: The markup enclosed in angle brackets (e.g., `<h1>`)
- **Element**: The complete structure including opening tag, content, and closing tag

Types of HTML Tags

1. **Paired Tags**: Have both opening and closing tags

```
<h1>Heading Content</h1>
<p>Paragraph content</p>
```

2. **Empty/Self-closing Tags**: Don't require closing tags

```
<br>
<img src="image.jpg" alt="Description">
<hr>
```

Working with Text Elements

Headings

HTML provides six levels of headings, from `<h1>` (largest) to `<h6>` (smallest):

```
<h1>Main Heading</h1>
<h2>Sub Heading 1</h2>
<h3>Sub Heading 2</h3>
<h4>Sub Heading 3</h4>
<h5>Sub Heading 4</h5>
<h6>Sub Heading 5</h6>
```

Best Practices:

- Use `<h1>` for main page title (only once per page)
- Use headings to create document hierarchy
- Don't skip heading levels

Paragraphs

The `<p>` tag creates paragraphs with automatic spacing[23]:

```
<p>This is a paragraph of text.</p>
<p>This is another paragraph.</p>
```

Line Breaks

The `<br>` tag creates line breaks without paragraph spacing[23]:

```
<p>First line<br>
Second line<br>
Third line</p>
```

Lists in HTML

Unordered Lists

```
<ul>
    <li>First item</li>
    <li>Second item</li>
    <li>Third item</li>
</ul>
```

Ordered Lists

```
<ol>
    <li>First step</li>
    <li>Second step</li>
    <li>Third step</li>
</ol>
```

Tables in HTML

Basic Table Structure

```
<table>
    <tr>
        <th>Header 1</th>
        <th>Header 2</th>
        <th>Header 3</th>
    </tr>
    <tr>
        <td>Data 1</td>
        <td>Data 2</td>
        <td>Data 3</td>
    </tr>
</table>
```

Table Elements

- **<table>**: Defines the table structure
- **<tr>**: Table row
- **<th>**: Table header cell (bold and centered by default)
- **<td>**: Table data cell

- **<caption>**: Table title or description
- **<thead>**: Groups header content
- **<tbody>**: Groups body content
- **<tfoot>**: Groups footer content

Frames in HTML

Frame Structure

Note: Frames are deprecated in HTML5 but may appear in legacy systems.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<html>
<head>
    <title>Frameset Document</title>
</head>
<frameset cols="20%, 80%">
    <frameset rows="100, 200">
        <frame src="frame1.html">
        <frame src="frame2.html">
    </frameset>
    <frame src="frame3.html">
    <noframes>
        <p>Your browser does not support frames.</p>
    </noframes>
</frameset>
</html>
```

Working with Hyperlinks

Basic Link Structure

```
<a href="https://www.example.com">Link Text</a>
```

Types of Links

1. **External Links**:

   ```
   <a href="https://www.google.com">Visit Google</a>
   ```

2. **Internal Links**:

```
<a href="about.html">About Us</a>
```

3. **Email Links**:

```
<a href="mailto:someone@example.com">Send Email</a>
```

4. **Phone Links**:

```
<a href="tel:+1234567890">Call Now</a>
```

5. **Image Links**:

```
<a href="destination.html">
    <img src="image.jpg" alt="Description">
</a>
```

Link Attributes

- **href**: Specifies the destination URL
- **title**: Provides tooltip text
- **target**: Specifies where to open the link
  - `_blank`: Opens in new window/tab
  - `_self`: Opens in same window (default)

Images and Multimedia

Image Element

```
<img src="image.jpg" alt="Description" width="300" height="200">
```

Essential Image Attributes:

- **src**: Image file path or URL
- **alt**: Alternative text for accessibility
- **width** and **height**: Dimension specifications

Audio Elements

```
<audio controls>
    <source src="audio.mp3" type="audio/mpeg">
    <source src="audio.ogg" type="audio/ogg">
```

```
    Your browser does not support the audio element.
</audio>
```

Video Elements

```
<video controls width="400" height="300">
    <source src="video.mp4" type="video/mp4">
    <source src="video.webm" type="video/webm">
    Your browser does not support the video element.
</video>
```

HTML5 Media Elements

Element	Description
`<audio>`	Embeds sound files into web pages
`<video>`	Embeds video files into web pages
`<source>`	Specifies multiple media resources
`<embed>`	Embeds external applications/multimedia
`<track>`	Defines text tracks for media elements

Forms and Controls

Basic Form Structure

```
<form action="submit.php" method="post">
    <!-- Form controls go here -->
</form>
```

Input Controls

1. **Text Input**:

   ```
   <input type="text" name="username" id="username" required>
   ```

2. **Password Input**:

   ```
   <input type="password" name="password" id="password">
   ```

3. **Email Input**:

```
<input type="email" name="email" id="email">
```

4. **Checkboxes**:

```
<input type="checkbox" name="subscribe" id="subscribe">
<label for="subscribe">Subscribe to newsletter</label>
```

5. **Radio Buttons**:

```
<input type="radio" name="gender" value="male" id="male">
<label for="male">Male</label>
<input type="radio" name="gender" value="female" id="female">
<label for="female">Female</label>
```

Button Controls

```
<!-- Submit button -->
<button type="submit">Submit Form</button>

<!-- Reset button -->
<button type="reset">Reset Form</button>

<!-- Generic button -->
<button type="button">Click Me</button>

<!-- Input buttons -->
<input type="submit" value="Submit">
<input type="reset" value="Reset">
<input type="button" value="Button">
```

Select Controls

```
<select name="country" id="country">
    <option value="usa">United States</option>
    <option value="canada">Canada</option>
    <option value="uk">United Kingdom</option>
</select>
```

Textarea Control

```
<textarea name="message" rows="4" cols="50" placeholder="Enter your message"></textarea>
```

Form Labels and Accessibility

```
<label for="username">Username:</label>
<input type="text" name="username" id="username">
```

HTML Best Practices

1. **Use Semantic HTML**: Choose elements based on meaning, not appearance
2. **Proper Nesting**: Close tags in reverse order of opening
3. **Accessibility**: Always include alt text for images and labels for form controls
4. **Validation**: Use HTML5 validation attributes (required, pattern, etc.)
5. **Clean Code**: Use proper indentation and commenting
6. **Mobile-First**: Include viewport meta tag for responsive design

Case Study: Complete HTML Page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Complete HTML Example</title>
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="#home">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#contact">Contact</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <section id="home">
            <h1>Welcome to Our Website</h1>
            <img src="banner.jpg" alt="Website banner">
            <p>This is a complete HTML page example demonstrating various elements.</p>
        </section>

        <section id="about">
```

```
            <h2>About Us</h2>
            <table>
                <tr>
                    <th>Service</th>
                    <th>Description</th>
                </tr>
                <tr>
                    <td>Web Design</td>
                    <td>Creating beautiful websites</td>
                </tr>
                <tr>
                    <td>Development</td>
                    <td>Building functional applications</td>
                </tr>
            </table>
        </section>

        <section id="contact">
            <h2>Contact Us</h2>
            <form action="contact.php" method="post">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required>
                <br><br>

                <label for="email">Email:</label>
                <input type="email" id="email" name="email" required>
                <br><br>

                <label for="message">Message:</label>
                <textarea id="message" name="message" rows="4" cols="50"></textarea>
                <br><br>

                <button type="submit">Send Message</button>
            </form>
        </section>
    </main>

    <footer>
        <p>&copy; 2024 Our Website. All rights reserved.</p>
    </footer>
</body>
</html>
```

This comprehensive example demonstrates the integration of all HTML concepts covered in this study guide, providing a practical foundation for web development.