

Introduction

Madiba Hudson-Quansah

CONTENTS

CHAPTER 1	NETWORKS	PAGE 2
1.1	Introduction Switch vs Router — 3	2
1.2	OSI (Open Systems Interconnection) Model Application Layer — 3 • Presentation Layer — 3 • Session Layer — 4 • Transport Layer — 4 • Network Layer — 4 • Data Link Layer — 4 • Physical Layer — 4	3
CHAPTER 2	RANDOM STUFF	PAGE 5
2.1	Network and Domain Addresses Internet Protocol — 5	5
2.2	Domains	6
CHAPTER 3	HYPERTEXT TRANSFER PROTOCOL (HTTP)	PAGE 7
	Introduction — 7 • HTTP Methods / Verbs — 7 • HTTP Status Codes — 8	
CHAPTER 4	HYPERTEXT MARKUP LANGUAGE (HTML)	PAGE 9
4.1	Introduction	9
4.2	Cascading Style Sheets (CSS) Length Units — 10	9
CHAPTER 5	JAVASCRIPT	PAGE 11
5.1	Introduction	11
5.2	Representational State Transfer (REST) Architecture	11
5.3	Asynchronous JavaScript and XML (AJAX)	12

Chapter 1

Networks

1.1 Introduction

Definition 1.1.1: Local Area Network (LAN)

A network that is confined to a relatively small area. It is generally limited to a geographic area such as a writing lab, school, or building.

Definition 1.1.2: IP (Internet Protocol) Address

A unique string of numbers separated by periods that identifies each computer using the Internet Protocol to communicate over a network.

Definition 1.1.3: Wide Area Network (WAN)

A network that covers a broad area (i.e., any telecommunications network that links across metropolitan, regional, or national boundaries).

Definition 1.1.4: Intranet

A private network that is contained within an enterprise. It may consist of many interlinked local area networks and also use leased lines in the wide area network.

Definition 1.1.5: Router

A device that forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet.

Definition 1.1.6: Switch

A device that connects devices together on a computer network by using packet switching to receive, process, and forward data to the destination device.

Definition 1.1.7: Internet

A global network of networks that consists of millions of private, public, academic, business, and government networks of local to global scope that are linked by a broad array of electronic, wireless, and optical networking technologies.

Definition 1.1.8: Client

A computer that accesses a server for resources.

Definition 1.1.9: Server

A computer that provides data to clients.

1.1.1 Switch vs Router

A switch is used to connect various devices within a network. It is used to create a network. A router is used to connect various networks. It is used to route data packets between different networks.

1.2 OSI (Open Systems Interconnection) Model

Definition 1.2.1: OSI Model

A conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstraction layers.

This model is composed of seven layers, each specifying particular protocols and functions that are performed at each layer, with each subsequent layer building on the one below it. The layers are from top to bottom:

- Application Layer / Layer 7
- Presentation Layer / Layer 6
- Session Layer / Layer 5
- Transport Layer / Layer 4
- Network Layer / Layer 3
- Data Link Layer / Layer 2
- Physical Layer / Layer 1

1.2.1 Application Layer

Definition 1.2.2: Application Layer

Protocols designed for process-to-process communication and provide services directly to the application. This layer interacts with software applications that implement a communicating component. Examples of protocols include HTTP, SMTP, and FTP.

1.2.2 Presentation Layer

Definition 1.2.3: Presentation Layer

The presentation layer ensures that the information that the application layer of one system sends out is readable by the application layer of another system. It translates data between the application layer and the network format. This is where encoding and encryption is done. Examples of protocols include SSL and TLS.

1.2.3 Session Layer

Definition 1.2.4: Session Layer

Establishes, manages, and terminates connections between applications. It provides services such as dialogue control and synchronization. It also provides for full-duplex, half-duplex, or simplex operation, and establishes checkpointing, adjournment, termination, and restart procedures. Examples of protocols include NetBIOS and PPTP.

1.2.4 Transport Layer

Definition 1.2.5: Transport Layer

Provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer. Examples of protocols include TCP and UDP.

1.2.5 Network Layer

Definition 1.2.6: Network Layer

Provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. It is responsible for packet forwarding including routing through intermediate routers. Examples of protocols include IP and ICMP.

1.2.6 Data Link Layer

Definition 1.2.7: Data Link Layer

Provides node-to-node data transfer, and is responsible for error detection and correction. It ensures data is transmitted reliably. Examples of protocols include Ethernet and PPP.

1.2.7 Physical Layer

Definition 1.2.8: Physical Layer

Transmits raw data bits over a physical medium. It is responsible for the actual physical connection between devices. Examples of protocols include Ethernet and DSL.

Chapter 2

Random Stuff

2.1 Network and Domain Addresses

2.1.1 Internet Protocol

Definition 2.1.1: IP Address

A unique string of numbers separated by periods that identifies each computer using the Internet Protocol to communicate over a network.

Definition 2.1.2: IPv4

The fourth version of the Internet Protocol. It is the most widely used version of the Internet Protocol. 32-bit addresses i.e. 2^{32} / 4 billion possible addresses. Structured as four octets (8 bits) separated by periods:

$$X_1.X_2.X_3.X_4$$

Where X_i is a number between 0 and 255. There are 3 classes of IPv4 addresses:

- Class A where X_1 is 10
- Class B where X_1 is 172. Most public IP addresses are Class B.
- Class C where X_1 is 192. Most private IP addresses are Class C.

Definition 2.1.3: IPv6

The most recent version of the Internet Protocol. It is designed to succeed IPv4. 128-bit addresses i.e. 2^{128} .

Definition 2.1.4: Localhost / Loopback Address

The address of the computer you are working on. It is always 127.0.0.1.

2.2 Domains

Definition 2.2.1: Domain

A human-readable address for a computer on the Internet. It is a string of characters that is easier to remember than an IP address. For an example of a domain, consider **https://www.google.com**, where:

https is the protocol, i.e. SSL, encrypted HTTP.

www is the subdomain or third level domain, i.e. the specific server.

google is the second level domain.

com is the top level domain.

Definition 2.2.2: Domain Name Server (DNS)

A server that translates domain names to IP addresses. It is like a phone book for the Internet.

Chapter 3

HyperText Transfer Protocol (HTTP)

3.0.1 Introduction

Definition 3.0.1: HTTP

The protocol used to transfer hypertext such as HTML. It is the foundation of data communication for the World Wide Web. HTTP requests are built upon the TCP/IP protocol, with HTTP requests and responses being sent over TCP connections.

A HTTP request is made up of the following components:

Request Line - Contains the request method / **verb**, the URL, and the HTTP version.

Headers - Contains additional information about the request, such as the user agent, the host, and the content type.

Body - Contains the data being sent in the request.

HTTP is a stateless protocol, meaning that each request is independent of others and no data is retained between requests, by the http server. To maintain state between requests, there are several methods:

- Cookies
- Sessions
- URL Parameters
- Hidden Form Fields
- Local Storage

3.0.2 HTTP Methods / Verbs

Definition 3.0.2: Verb

Indicates the action / operation to be performed on the resource identified by the URL.

The most common HTTP verbs are:

GET - Requests data from a specified resource.

POST - Submits data to be processed to a specified resource.

PUT - Updates a specified resource.

DELETE - Deletes a specified resource.

Some other HTTP verbs include:

PATCH - Updates a specified resource, with it differing from PUT in that it only updates the fields that are provided, whereas PUT updates the entire resource.

HEAD - Requests the headers of a specified resource, i.e. the same as GET but without the body.

CONNECT - Establishes a tunnel to the server identified by the target resource.

OPTIONS - Requests information about the communication options available for the target resource, i.e. what methods are allowed.

TRACE - Performs a message loop-back test along the path to the target resource.

3.0.3 HTTP Status Codes

Definition 3.0.3: Status Code

A three-digit number that indicates the status of the HTTP response.

Status codes are grouped into five categories:

1xx - Informational, i.e. the request has been received and the process is continuing.

2xx - Success, i.e. the request was successfully received, understood, and accepted.

3xx - Redirection, i.e. further action must be taken to complete the request.

4xx - Client Error, i.e. the request contains bad syntax or cannot be fulfilled.

5xx - Server Error, i.e. the server failed to fulfil an apparently valid request.

Chapter 4

HyperText Markup Language (HTML)

4.1 Introduction

Definition 4.1.1: HTML

A markup language (i.e. a language that defines the structure of content) for describing web documents.

Invented in 1991 by Tim Berners-Lee for use in the World Wide Web project for CERN, HTML was originally designed for static documents with links to other documents. It has since evolved to include multimedia content and dynamic content and is maintained by the World Wide Web Consortium (W3C). The structure of an HTML document is as follows:

Document Type Declaration - The document type declaration is an instruction to the web browser about what version of HTML the page is written in. It is not case sensitive.

HTML Element - The root element of an HTML page. It is the container for all other HTML elements.

Head Element - Contains meta-information about the document, such as its title, links to stylesheets, and scripts.

Body Element - Contains the content of the document, such as text, images, and links.

4.2 Cascading Style Sheets (CSS)

Definition 4.2.1: CSS

A style sheet language used for describing the presentation of a document written in HTML.

Before CSS was introduced in 1996, the style of web pages was defined within the HTML markup. This came with the following drawbacks:

- Each element needed to be styled individually, leading to redundancy and repetition.
- The style of a document was mixed with its content, making it difficult to read, maintain and update.

Invented alongside HTML by Håkon Wium Lie, CSS was designed to address these issues. It allows for the separation of content from presentation, making it easier to maintain and update the style of a document. The core concepts of CSS are:

Specificity - The process of determining which style rule applies to an element when multiple rules conflict. Where the precedence of rules is described as:

Tag Selector - Lowest precedence.

Class Selector - Middle precedence.

ID Selector - Highest precedence.

Cascading - The process of combining multiple style sheets and resolving conflicts between them. This leads to the order of rules being important, with most recently defined rules taking precedence if two rules conflict.

Inheritance - The process of an element inheriting the style of its parent element. This allows for the creation of a consistent style across a document. Properties like font size, colour, and line height are inherited by default.

4.2.1 Length Units

There are several units of length in CSS:

Pixels (px) - A pixel is a single dot on a screen. It is a fixed-size unit that is the same size regardless of the resolution of the screen, i.e. uses the monitor's pixel as a reference.

Percentage (%) - A percentage is a relative unit that is a percentage of the parent element's size.

Ems (em) - An em is a relative unit that is the size of the font of the element. It is useful for setting the size of text relative to the size of the parent element.

Rems (rem) - A rem is a relative unit that is the size of the font of the root / `<html>` element. It is useful for setting the size of text relative to the size of the root element.

Viewport Width (vw) - A viewport width is a relative unit that is a percentage of the viewport's width. It is useful for setting the size of an element relative to the width of the viewport. Where the viewport is the visible area of a web page.

Viewport Height (vh) - A viewport height is a relative unit that is a percentage of the viewport's height. It is useful for setting the size of an element relative to the height of the viewport.

Chapter 5

JavaScript

5.1 Introduction

Definition 5.1.1: JavaScript

A high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is primarily used for client-side scripting in web development.

Invented by Brendan Eich in 1995 for use in the Netscape Navigator web browser, JavaScript was originally designed to add interactivity to web pages. It has since evolved to include support for server-side scripting and is now used in a wide variety of applications, including web development, game development, and mobile app development. With the development of Google's V8 engine, JavaScript has escaped the browser and now runs on servers, desktops, and even IoT.

5.2 Representational State Transfer (REST) Architecture

Definition 5.2.1: REST

A software architectural style that defines a set of constraints to be used for creating web services. Representational state transfer means that a server will with the representation of a resource, data in HTML, XML or JSON format, and that the response will contain hypermedia links that can be followed to make the state of the system change.

The REST architecture is based on the following principles:

- Client-Server** - Client concerns should be separate from server concerns, i.e. the client should not be concerned with data storage and the server should not be concerned with the user interface. This avoids tying server logic to a specific client type.
- Stateless** - Each client request should be independent and contain all the information needed to fulfil the request. This allows the server to handle each request without needing to store information about the client between requests.
- Cacheable** - Servers can label responses as cacheable, enabling clients to reuse responses for equivalent requests, improving performance.
- Uniform Interface** - The server must offer a consistent interface for all components, following constraints, such as resource identification, manipulation through representations, self-descriptive messages, and hypermedia as the engine of application state.
- Layered System** - Server components should only interact with their immediate layers, limiting system complexity and promoting component isolation.
- Code on Demand (Optional)** - Servers can provide code for clients to execute as needed reducing the initial payload and allowing clients to load only what they need.

REST uses standard HTTP methods to perform operations on resources:

GET - Retrieves a representation of the resource.

POST - Creates a new resource.

PUT - Updates an existing resource.

DELETE - Deletes a resource.

5.3 Asynchronous JavaScript and XML (AJAX)