UNIT I

FOUNDATIONS OF COMPUTER NETWORKS

Introduction to Network Protocols

- Definition: A network protocol is a formal set of rules, conventions, and standards that define how data is formatted, transmitted, and received between computing devices. It's essentially a common language that all networked devices must agree upon to communicate successfully.
- Purpose: Protocols ensure reliable, ordered, and error-checked exchange of information. They specify things like:
    - How a connection is started and ended.
    - How data packets are structured (header, payload, trailer).
    - How errors are detected and corrected.
    - How security and authentication are handled.

Transmission Control Protocol/Internet Protocol (TCP/IP) suite is the foundation of the modern internet.

1. IP (Internet Protocol)

- Layer: Network Layer (Layer 3).
- Function: Handles logical addressing (IP addresses) and routing. It is responsible for delivering packets from the source host across multiple networks to the final destination host.
- Nature: Connectionless (each packet is treated independently) and unreliable (it doesn't guarantee delivery or correct order; this is handled by higher layers like TCP).
- Versions: IPv4 (32-bit addresses, e.g., 192.168.1.1) and IPv6 (128-bit addresses, designed to replace IPv4).

2. TCP (Transmission Control Protocol)

- Layer: Transport Layer (Layer 4).
- Function: Provides reliable, connection-oriented data transmission. It ensures packets are delivered, delivered without errors, and delivered in the correct sequence.
- Mechanism: Uses the Three-Way Handshake (SYN, SYN-ACK, ACK) to establish a connection and uses acknowledgements and retransmission for reliability.

3. UDP (User Datagram Protocol)

- Layer: Transport Layer (Layer 4).

- Function: Provides unreliable, connectionless data transmission. It sends data fast without guaranteed delivery or error checking.
- Use Case: Preferred for applications where speed is more important than guaranteed delivery, such as video streaming, online gaming, and DNS lookups.

4. Application Layer Protocols

These are the protocols students interact with most frequently:

- HTTP (Hypertext Transfer Protocol): Used to access and transfer web pages (requests and responses).
- HTTPS (HTTP Secure): The secure version of HTTP, which encrypts data using SSL/TLS.
- DNS (Domain Name System): Translates human-readable domain names (e.g., www.google.com) into machine-readable IP addresses.
- FTP (File Transfer Protocol): Used to transfer files between client and server.
- SMTP (Simple Mail Transfer Protocol): Used for sending email.
- POP3/IMAP: Used for receiving email.

Security Protocols

- SSL/TLS (Secure Sockets Layer / Transport Layer Security): Cryptographic protocols designed to provide communication security over a computer network. They are used to secure web traffic (HTTPS) and other application data.
- VPN Protocols (e.g., IPSec, OpenVPN): Used to create a secure, encrypted tunnel over a public network, allowing remote users to securely access a private network.

The Domain Name System (DNS)

- DNS is a distributed, hierarchical naming system that translates human-readable domain names (like `www.google.com`) into machine-readable Internet Protocol (IP) addresses (like `142.250.68.100`).
- It's often called the "phone book of the internet." You look up a person's name (domain name) to get their phone number (IP address).

- The Domain Name Space

Domain names are structured in an inverted tree hierarchy, separated by dots (`.`):

- Root (.): The very top level, represented by a single dot. Handled by Root Name Servers (13 logical server groups worldwide).

- Top-Level Domain (TLD): The rightmost part of the domain (e.g., `.com`, `.org`, `.edu`, country codes like `.uk`). Handled by TLD Name Servers.
- Second-Level Domain: The name registered by the organization (e.g., `google` in `google.com`).
- Subdomain/Host: Optional parts to the left (e.g., `www` or `mail`).

2. The Four Key DNS Servers

Server Type	**Role**	**Example Function**
DNS Resolver (or Recursor)	The client's first point of contact (usually provided by your ISP or a public service like Google DNS 8.8.8.8). It performs the entire lookup on the client's behalf.	Receives query for `google.com` and starts the search.
Root Name Server	Sits at the top. It doesn't know the IP, but it knows where to find the TLD servers.	Directs the resolver to the .com TLD server.
TLD Name Server	Manages a specific TLD (e.g., all `.com` domains).	Directs the resolver to the Authoritative server for `google.com`.
Authoritative Name Server	The final source of truth. It holds the actual DNS records for a specific domain (zone).	Returns the IP address for `www.google.com` to the resolver.

The DNS Lookup Process (Resolution)

1. Client Request: A user enters a domain name (e.g., `www.example.com`) in a browser.
2. Cache Check: The browser and Operating System (OS) check their local DNS cache. If the IP is found, the process stops here (very fast).
3. Resolver Query: If not found, the query is sent to the DNS Resolver.
4. Iterative Queries: The resolver performs a series of iterative queries through the hierarchy:
    - Asks the Root Server.
    - The Root Server refers it to the TLD Server (`.com`).
    - The TLD Server refers it to the Authoritative Server for `example.com`.

5. Final Answer: The Authoritative Server returns the target IP Address to the Resolver.
6. Caching and Response: The Resolver caches the answer and sends the IP address back to the client's browser.
7. Connection: The browser uses the IP address to initiate a connection (e.g., a TCP connection) and load the webpage.

Key DNS Record Types

DNS records are resource records (RRs) stored on the Authoritative Name Server that map a name to a specific piece of data.

Record Type	**Description**	**Purpose**
A Record	Address Record. Maps a hostname to an IPv4 address. (The most common type.)	`example.com` $\rightarrow$ `192.0.2.1`
AAAA Record	Maps a hostname to an IPv6 address.	`example.com` $\rightarrow$ `2001:db8::1`
CNAME Record	Canonical Name. Creates an alias from one name to another name.	`blog.example.com` $\rightarrow$ `posts.example.com`
MX Record	Mail Exchanger. Specifies the mail servers responsible for receiving email for a domain.	Directs email to the correct mail server (e.g., `mail.example.com`).
NS Record	Name Server. Specifies the authoritative name servers for a domain/zone.	Delegates authority to other DNS servers.

DHCP: Dynamic Host Configuration Protocol

- DHCP is a network management protocol that automates the process of assigning IP addresses and other vital network configuration parameters to client devices (hosts) when they connect to a network.
- It eliminates the need for a network administrator to manually configure every device, which is time-consuming and prone to errors (like IP conflicts). It allows devices to "plug and play."
- Protocol Layer: DHCP operates primarily at the Application Layer of the OSI model, but its messages are encapsulated in UDP packets (Transport Layer) using well-known ports:
    - Server listens on: UDP Port 67
    - Client listens on: UDP Port 68

1. DHCP Server:
    - Maintains a pool of available IP addresses (called a scope).
    - Assigns an IP address, subnet mask, default gateway (router), and DNS server addresses to clients.
    - Manages the lease duration (the time a client can use the IP).
2. DHCP Client:
    - The device (laptop, phone, printer, etc.) that requires an IP address to communicate on the network.
    - Initiates the communication process upon booting up or connecting to the network.
3. DHCP Relay Agent:
    - A component (often a router or switch) used in large networks segmented into subnets (different network IDs).
    - Forwards the client's broadcast messages (like DHCPDISCOVER) from one subnet to a DHCP server located on a different subnet, converting the broadcast into a unicast message.

The DHCP DORA Process

The core operation of DHCP is a four-step message exchange between the client and the server, easily remembered by the acronym DORA:

Step	**Message**	**Sender → Receiver**	**Description**
D	Discover (DHCPDISCOVER)	Client $\rightarrow$ Server (Broadcast)	Client broadcasts a message to find any available DHCP servers on the network.

Step	Message	Sender → Receiver	Description
O	Offer (DHCPOFFER)	Server $\rightarrow$ Client (Broadcast/Unicast)	One or more DHCP servers respond with an offered IP address and configuration settings from their pool.
R	Request (DHCPREQUEST)	Client $\rightarrow$ Server (Broadcast)	The client accepts one of the offers and broadcasts a request to confirm the chosen IP address and server. (This also tells other servers to withdraw their offers).
A	Acknowledge (DHCPACK)	Server $\rightarrow$ Client (Unicast)	The selected server sends a final confirmation packet containing the full configuration details and the lease duration. The client then configures its network interface.

SCADA (Supervisory Control and Data Acquisition) is a fundamental concept in industrial automation and control systems.

Introduction to SCADA

SCADA is an acronym for Supervisory Control and Data Acquisition. It is a control system architecture that uses computers, networked data communications, and graphical user interfaces (HMIs) for high-level supervisory management of plants and processes.

- Core Function: It monitors, gathers, and processes real-time data from field devices (like sensors and motors) and allows operators to issue control commands to those devices from a central location, often miles away.
- Role: SCADA handles the "supervisory" role (monitoring the big picture, adjusting setpoints, and handling alarms), while local controllers (PLCs/RTUs) handle the minute-to-minute "real-time control" logic.
- Analogy: Think of SCADA as the central nervous system of an industrial facility or infrastructure network, providing oversight and intelligence.

Key Components of a SCADA System

A SCADA system is composed of both hardware and software elements:

1. Field Devices (Sensors and Actuators):
    - Sensors: Measure physical parameters (temperature, pressure, flow rate, voltage, etc.) and convert them into electrical signals.
    - Actuators: Receive control signals (outputs) and perform a physical action (open a valve, start a motor, trip a breaker).
2. Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs):
    - These are rugged, microprocessor-based devices located in the field, close to the equipment.
    - They interface directly with the sensors and actuators.
    - They perform local, real-time control logic (e.g., if tank level is high, turn off the pump) and communicate data back to the central SCADA system. PLCs are typically faster and used more in plant floors, while RTUs are designed for geographically scattered remote sites (like pipelines or substations).
3. Communication Infrastructure:
    - The network that connects the central site to the RTUs/PLCs.

- Can use a mix of industrial protocols, wired (Ethernet, fiber optic), and wireless (radio, cellular) technology, often spanning very large distances.

4. Master Terminal Unit (MTU) / Central Host Computer:
   - The central server that houses the SCADA software.
   - It collects, processes, and archives data from all the PLCs/RTUs.
   - It hosts the HMI software and manages the overall system.
5. Human-Machine Interface (HMI):
   - The operator's graphical interface to the system.
   - It displays real-time data, historical trends, system schematics (mimic diagrams), and alarms.
   - Operators use the HMI to monitor the process, acknowledge alarms, and send supervisory control commands (e.g., change the temperature setpoint).
   - A Historian is often a specialized database component that archives vast amounts of time-stamped process data for reporting and analysis.

Applications and Benefits

SCADA systems are critical for managing large-scale, distributed infrastructure and industrial processes.

Key Applications

- Utilities: Power Generation, Transmission, and Distribution (Smart Grids).
- Water: Water Treatment and Distribution, Wastewater Collection and Treatment.
- Oil and Gas: Pipeline monitoring, pumping stations, and refinery operations.
- Manufacturing: Factory and production line automation (e.g., automotive, food & beverage).
- Transportation: Rail and subway systems, traffic control.

Primary Benefits

- Real-Time Monitoring and Control: Provides instantaneous visibility into the entire operation, allowing for immediate corrective action.
- Improved Efficiency and Optimization: Collects data to analyze performance, optimize resource use, and maintain optimal setpoints.

- Enhanced Safety: Automatically monitors critical parameters and triggers alarms for abnormal conditions (e.g., high pressure, equipment failure), reducing the risk of human error and catastrophic failure.
- Reduced Costs: Minimizes the need for personnel to travel to remote sites for manual monitoring or adjustment.
- Data Analysis: Stores historical data (in the Historian) for reporting, compliance, and long-term process improvement.