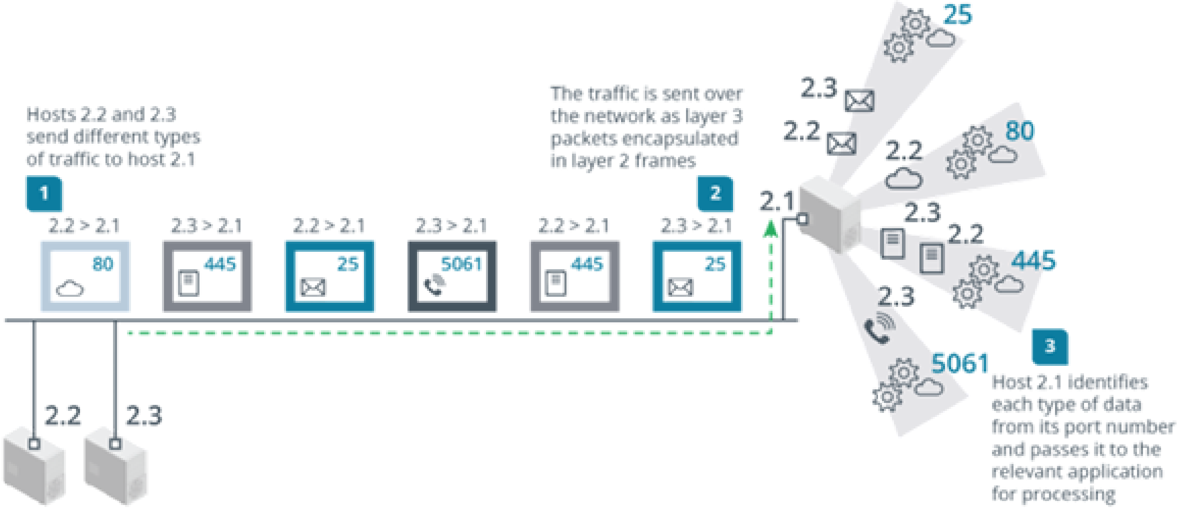
Topic 5C: Compare Protocols and Ports

The **Transport Layer** in the **TCP/IP model** ensures that data is correctly delivered to the right application on a device. Since a computer or server can handle **multiple types of communication at once** (e.g., browsing a website while checking email), the Transport Layer uses **port numbers** to differentiate these network services.

◆ How Port Numbers Work

Each application is assigned a **unique port number** (0-65535). Some common ports include:

- **Port 80 → HTTP (Web Browsing)**
- **Port 25 → SMTP (Email Sending)**
- **Port 445 → File Sharing (SMB)**
- **Port 5061 → VoIP (Internet Calls)**



Hosts 2.2 and 2.3 send different types of traffic to host 2.1

The traffic is sent over the network as layer 3 packets encapsulated in layer 2 frames

Host 2.1 identifies each type of data from its port number and passes it to the relevant application for processing

Communications at the transport layer. (Images © 123RF.com)

◆ Example:

Imagine **Host 2.2** and **Host 2.3** sending different types of data to **Host 2.1** (see image). Host 2.1 uses **port numbers** to identify which application should process each request.

◆ How Communication is Tracked Using Ports

Every communication session has **two port numbers**:

1. Destination Port (Well-Known Service Port) – Identifies the application **requested by the client** (e.g., port 80 for a web request).

2. Source Port (Random Client Port) – A temporary number assigned by the client (e.g., 47747) so the server knows **where to send the response**.

◆ Example:

- A web browser on Host 2.2 sends a request to 192.168.1.1:80 (**destination port 80**).
- Host 2.2 assigns a **random source port (e.g., 47747)**.
- The web server replies to **Host 2.2 on port 47747**, keeping the communication organized.

This system allows **multiple conversations** to happen simultaneously without confusion.

Transmission Control Protocol (TCP):

TCP (Transmission Control Protocol) ensures **reliable data delivery** across networks. Since **IP packets can be lost, damaged, or arrive out of order**, TCP provides **error checking, retransmission, and ordered delivery** to make sure data is received correctly. It is a **connection-oriented protocol**, meaning it first establishes a connection before sending data.

◆ **How TCP Works (Step-by-Step)**

1. Connection Establishment (3-Way Handshake)

Before sending data, TCP ensures both sender and receiver are **ready to communicate** using a **handshake process**:

- **SYN** → Sender requests to start a connection.
- **SYN/ACK** → Receiver acknowledges and agrees.
- **ACK** → Sender confirms, and the connection is established.

◆ **Example:** Like calling someone and waiting for them to **pick up before speaking**.

---

2. Reliable Data Transfer

- **Each packet gets a sequence number**, so data arrives **in order**.
- **Receiver sends an ACK (Acknowledgment) for each packet**.
- If a packet is **lost or corrupted**, the receiver sends a **NACK (Negative Acknowledgment)**, and the sender **resends the missing data**.

◆ **Example:** Like sending **numbered pages of a book**—if one is missing, the receiver asks for a resend.

---

3. Graceful Connection Termination (FIN Handshake)

When communication is complete, TCP **closes the connection safely** using a **FIN (Finish) handshake** to ensure no data is lost.

◆ **Example:** Like saying **"Goodbye" before ending a phone call** instead of hanging up suddenly.

TCP is used when the application protocol cannot **tolerate** missing or **damaged information.** For example, the following application protocols must use TCP:

- **Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS):**
  These protocols are used for web browsing, rely on TCP to deliver **web pages, images, and other resources** correctly. If even a single packet were lost, a web page might fail to load properly or appear incomplete. HTTPS adds an extra layer of security by encrypting data to protect user information. Other critical applications that use TCP include **file transfers (FTP, SFTP)** to prevent data corruption, **email services (SMTP, IMAP, POP3)** to ensure messages are delivered fully, and **remote access protocols (SSH, Telnet)** to provide smooth and uninterrupted connections. Without TCP, these applications would experience missing or out-of-order data, leading to **incomplete downloads, broken web pages, or corrupted emails**, making TCP the preferred choice for **reliable communication** over the internet.

- **Secure Shell (SSH):**
  Secure Shell (SSH) is a protocol used to securely access a computer's **command-line interface** over a network. It provides **encrypted communication**, ensuring that data transmitted between the user and the remote system remains **private and secure**. SSH is commonly used by **system administrators** to remotely manage servers, configure networks, and transfer files securely. Since SSH relies on a continuous and **error-free exchange of data**, it uses **TCP** to ensure that every packet is received in the correct order. If a packet were lost or corrupted, the **SSH session could disconnect**, making reliable transmission crucial. Unlike older remote access protocols like **Telnet**, which transmits data in plaintext, SSH encrypts all communications, preventing attackers from intercepting **passwords, commands, or sensitive data**.

User Datagram Protocol (UDP):

UDP (User Datagram Protocol) is a **fast, connectionless** communication method that does **not guarantee** delivery, sequencing, or error checking. Unlike **TCP**, which ensures reliable transmission with acknowledgments and retransmissions, UDP **prioritizes speed** by eliminating these extra steps. This makes UDP ideal for **time-sensitive applications** where slight data loss is acceptable, but delays are not.
For example, **video calls, online gaming, and live streaming** use UDP because a **missing or out-of-order packet** might cause a **minor glitch** but won't **interrupt the entire connection**. In contrast, TCP would **slow down** these applications by **retransmitting lost packets**, causing noticeable lag. While UDP itself does not ensure reliability, some applications use **error-checking mechanisms at the application layer** if needed. The **lower overhead of UDP** allows data to be transmitted **faster**, making it the preferred choice for **real-time communication** over networks.

Two other examples of protocols that use UDP are DHCP and TFTP:

- **Dynamic Host Configuration Protocol:**
  DHCP is a protocol used by devices to automatically obtain IP addresses and network settings from a DHCP server. It relies on UDP because it broadcasts requests across the network to find a server, and TCP does not support broadcasting. Since DHCP is a simple request-response process, if a packet is lost, the client just retries until it gets a response—there's no need for TCP's reliability mechanisms.

  ◆ Example: When you connect to Wi-Fi, your device sends a UDP DHCP request to obtain an IP address, subnet mask, and default gateway. If the response is lost, the device keeps retrying until it gets a valid configuration.

- **Doman Name System (DNS):**
  DNS is used to translate domain names (e.g., www.google.com) into IP addresses. It uses UDP on port 53 to ensure quick responses to user requests. Since DNS queries are small and independent, there's no need for the reliability of TCP. If a query fails, the client can simply resend the request or try a different DNS server.

  ◆ Example: When you type www.google.com in your browser, your device sends a UDP DNS request to find Google's IP address. If the response is lost, it asks again, ensuring minimal delay.

- **Trivial File Transfer Protocol (TFTP):**
  TFTP (Trivial File Transfer Protocol) is a simple protocol used to transfer files between devices, mainly for network device configuration and booting. Unlike FTP (File Transfer Protocol), which uses TCP, TFTP runs on UDP (port 69) because it prioritizes speed over reliability.

  Since UDP does not provide acknowledgments, TFTP has its own built-in acknowledgment system at the application layer. This allows it to resend lost data without relying on TCP's complex error-checking mechanisms.

  Example:
- **Network routers and switches** often use TFTP to **download firmware updates** or **retrieve configuration files** when rebooting.
- Since these files are small and do not require advanced security, **TFTP over UDP** is a **quick and efficient choice**.

Some Important Well-Known Ports:

1️⃣ File Transfer & Remote Access

Port	Protocol	TCP/UDP	Purpose
20	FTP (File Transfer Protocol - Data)	TCP	Transfers files over a network (data connection).
21	FTP (Control)	TCP	Manages FTP commands and control signals.
22	SSH (Secure Shell)	TCP	Secure remote access to command-line interfaces.
23	Telnet	TCP	Unsecured remote access (not recommended).
3389	RDP (Remote Desktop Protocol)	TCP	Secure graphical remote access to a computer.

2️⃣ Email Services

Port	Protocol	TCP/UDP	Purpose
25	SMTP (Simple Mail Transfer Protocol)	TCP	Sends email messages.
110	POP3 (Post Office Protocol v3)	TCP	Retrieves email from a mailbox (downloads to client).
143	IMAP (Internet Message Access Protocol)	TCP	Manages emails on a server (keeps messages stored remotely).

3️⃣ Web & Network Services

Port	Protocol	TCP/UDP	Purpose
53	DNS (Domain Name System)	TCP/UDP	Resolves domain names to IP addresses.
67	DHCP Server	UDP	Assigns dynamic IP addresses to clients.
68	DHCP Client	UDP	Requests an IP address from a DHCP server.
80	HTTP (HyperText Transfer Protocol)	TCP	Unsecured web browsing.
443	HTTPS (Secure HTTP)	TCP	Secure web browsing with encryption.

4️⃣ File Sharing & Network Management

Port	Protocol	TCP/UDP	Purpose
137-139	NetBIOS/NetBT	UDP/TCP	Supports older Windows networking functions.
445	SMB (Server Message Block)	TCP	Windows file and printer sharing.
161	SNMP (Simple Network Management Protocol)	UDP	Queries network device status.
162	SNMP Trap	UDP	Sends alerts to a management server.
389	LDAP (Lightweight Directory Access Protocol)	TCP	Queries user and device information in a directory.

CompTIA A+ Well-Known Ports - Questions & Answers

---

1. True or False? At the Transport layer, connections between hosts to exchange application data are established over a single port number.

✅ **Answer: False.** The server application is identified by one port, but the client must also assign its own port to track the connection.

---

2. What feature of DHCP means that it must use UDP at the transport layer?

✅ **Answer: DHCP uses broadcast addressing**, which is not supported by the connection-oriented **Transmission Control Protocol (TCP)**. Consequently, **DHCP uses the connectionless UDP** to allow clients to broadcast requests for an IP address.

---

3. Another technician has scribbled some notes about a firewall configuration. The technician has listed only the port numbers 25 and 3389. What is the purpose of the protocols that use these ports by default?

✅ **Answer:**

- **Port 25 (TCP)** → Used by **Simple Mail Transfer Protocol (SMTP)** for sending and receiving emails.
- **Port 3389 (TCP)** → Used by **Remote Desktop Protocol (RDP)** to connect to a computer's graphical interface over the network.

---

4. The technician has made a note to check that port 445 is blocked by the firewall. What is the purpose of the protocol that uses this port by default, and why should it be blocked?

✅ **Answer:**

- **Port 445 (TCP)** → Used by **Server Message Block (SMB)**, which allows **Windows file and printer sharing** over a network.
- **Why should it be blocked? SMB is meant for local networks** and should **not be exposed to the internet** due to **security risks**, as it has been a target for ransomware and malware attacks (e.g., WannaCry).