3 Computer Networks

Hui Xu, xuh@fudan.edu.cn

Learning Objectives:

- Understand IP and Routing.
- Understand how reliable data transmission is achieved via TCP/IP.
- Know what HTTP and DNS are.

3.1 Internet

The Internet is a network of networks. It' s the global infrastructure that connects billions of devices, including computers, smartphones, servers, sensors. All devices speak the same "language" or protocols, named Internet Protocol (IP). The former version of Internet is named ARPANET (Advanced Research Projects Agency Network) funded by the U.S. Department of Defense' s Advanced Research Projects Agency (now DARPA). The first ARPANET connection went live on October 29, 1969, between the University of California, Los Angeles (UCLA) and the Stanford Research Institute (SRI). Throughout the 1970s, ARPANET grew to dozens of universities and research labs.

3.1.1 MAC Address

Every device that connects to a network contains a Network Interface Controller (NIC). Each NIC is assigned a Media Access Control (MAC) address, which is a globally unique identifier used within local networks. You can think of a MAC address like a "citizen ID" for a piece of hardware. A standard MAC address is 48 bits long. The first 24 bits are assigned by the IEEE to a specific manufacturer, so all NICs produced by that manufacturer share the same prefix. The remaining 24 bits are assigned by the manufacturer to individual devices, making each NIC' s full address unique. This structure allows network devices to automatically distinguish one another on a local area network without any central registry.

MAC addresses are designed for local network communications. When data leaves a local network and travels across the Internet, MAC addresses are stripped off and replaced at each hop.

3.1.2 IP Address and Networks

The Internet Protocol (IP) is the fundamental addressing system that allows devices to locate and communicate with each other over Internet. Unlike MAC addresses, which are used within a local network, IP addresses are designed for end-to-end communication across multiple networks. Think of IP as the "street address" for computers: it tells data packets where to go so they can reach the correct destination.

An IP address (version 4) is 32 bits long, and it is usually written as four decimal numbers separated by dots, *e.g.,* `192.168.1.1`. It consists of two parts:

- *Network part:* The left n bits that identify the network.
- *Host part:* The remaining bits that identify a specific device (host) on that network.

To determine the network part of an IP address, we use either the CIDR notation or subnet masks. The *CIDR notation* is a slash followed by the number of prefix bits. For example, `192.168.1.1/24` means the first 24 bits identify the network. Similarly, *subnet masks* specifies which bits are network bits. For instance, `192.168.1.1/255.255.255.0` also means the first 24 bits are the network part.

All IP addresses within the same network must share the same network prefix. For example, `192.168.1.1/24` and `192.168.1.100/24` are the same network because their first 24 bits are both `192.168.1.X`, and there are 2^8 IP addresses within the network. `192.168.1.1/26` and `192.168.1.100/26` are *not* in the same network, because their first 26 bits differ, and the network capacity is only 2^6.

Every IP network has a gateway, which is the device that connects the local network to other networks or the Internet. Figure 3.1 demonstrates a typical network with three devices, *i.e.,* a smartphone, a laptop, and a Wi-Fi access point (AP). These devices are configured with IP addresses `192.168.1.1`-`192.168.1.3`, and the gateway is the IP of the Wi-Fi AP, `192.168.1.1`.

3.1.3 Address Resolution Protocol

While IP addresses are used for logical addressing, actual communication on a local network happens via MAC addresses. The Address Resolution Protocol (ARP) is a network protocol used to map a device' s IP address to its MAC address within a local network.

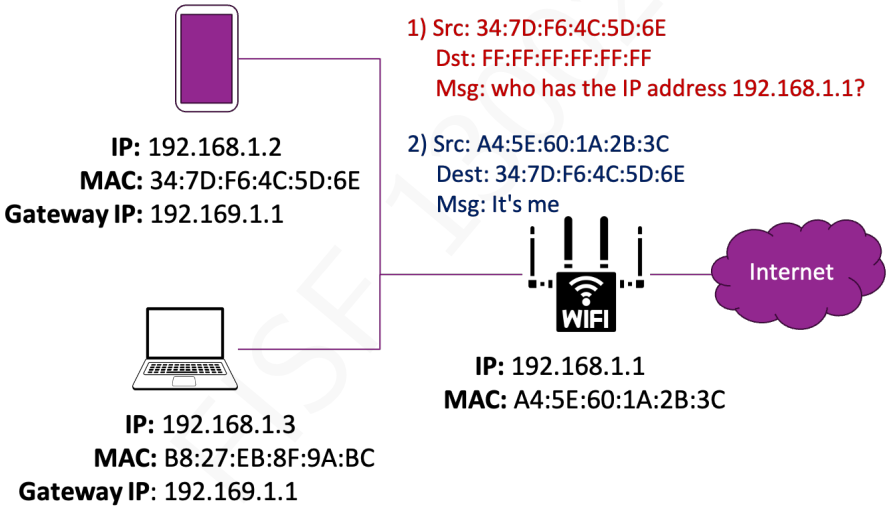

Figure 3.1: Demonstration of ARP.

Figure 3.1 illustrates a scenario in which a smartphone (IP address: `192.168.1.2`) queries the MAC address of the gateway (IP address: `192.168.1.1`). We assume the smartphone already knows the gateway' s IP address, as it is typically configured at the same time as the device' s own IP address. However, it does not yet know the gateway' s MAC address. To obtain it, the smartphone sends a broadcast message to all devices on the local network using the special MAC address `FF:FF:FF:FF:FF:FF`. This broadcast ensures that all devices on the local network receive the inquiry. The Wi-Fi access point recognizes that the requested IP address matches its own (`192.168.1.1`) and replies to the smartphone with its MAC address: `A4:5E:60:1A:2B:3C`.

3.1.4 Routing

When sending a message to a remote device, it needs routing. Routing is the process of determining the path that data packets take from the source device to the destination across one or more networks.

Unlike local communication within the same network, where packets can be delivered directly using MAC addresses, communication across multiple networks requires intermediate devices, called *routers*, to forward packets toward their destination.

A router typically has multiple Network Interface Controllers (NICs), each connected to a different network. Each NIC is assigned an IP address that belongs to the network it interfaces with. This allows the router to receive packets from one network and forward them to another, effectively connecting multiple networks together. For example, the Wi-Fi AP in Figure 3.2 has two NICs, the inside NIC is connected to the local network with an IP address `192.168.1.1/24`, and the outside NIC is connected to the Wide Area Network (WAN) with an IP address `58.40.90.117/24`. If a router has multiple NICs, it must have a routing table to determine where to forward incoming packets. The routing table contains information about which network is reachable via which NIC or next-hop router.

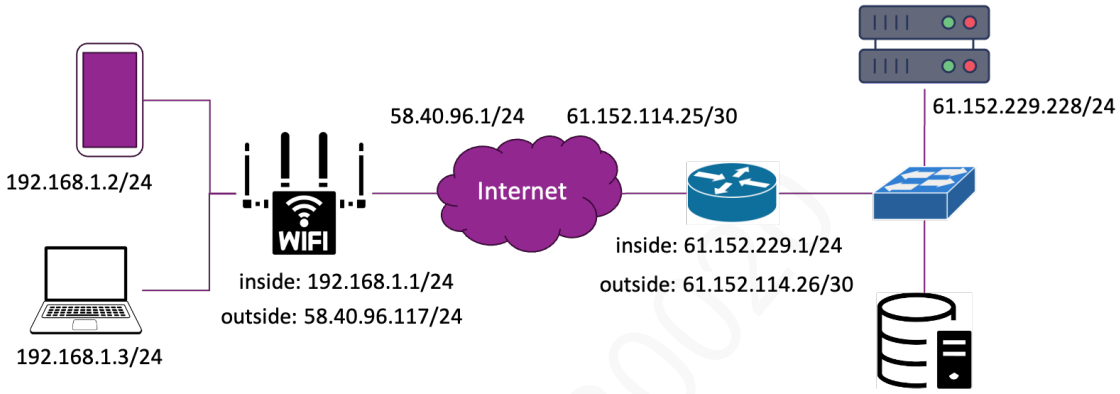

Figure 3.2: Demonstration of Wide Area Network and routing.

Now, we can wrap up things together. Suppose a message is sent from host `192.168.1.2` to destination `61.152.229.228`. The process is as follows:

1) The sender first queries the MAC address of the gateway using ARP.
2) It composes the message with the destination IP address `61.152.229.228`.
3) The message is sent to the gateway's MAC address on the local network.
4) The gateway forwards the message via its external NIC to the next hop `58.40.96.1`.
5) After traveling multiple hops across the Internet, the message reaches a router in the destination network, `61.152.114.26/30`.
6) The router checks its routing table and forwards the message to the destination via its internal NIC, `61.152.229.1`.

3.2 Messaging

3.2.1 OSI Model

In real-world scenarios, the OSI (Open Systems Interconnection) model defines seven distinct layers to standardize network communication, as shown in Figure 3.3. Each layer has a specific role in processing, transmitting, and delivering data, from the software application down to the physical medium. This layered approach allows different hardware and software systems to communicate seamlessly, while

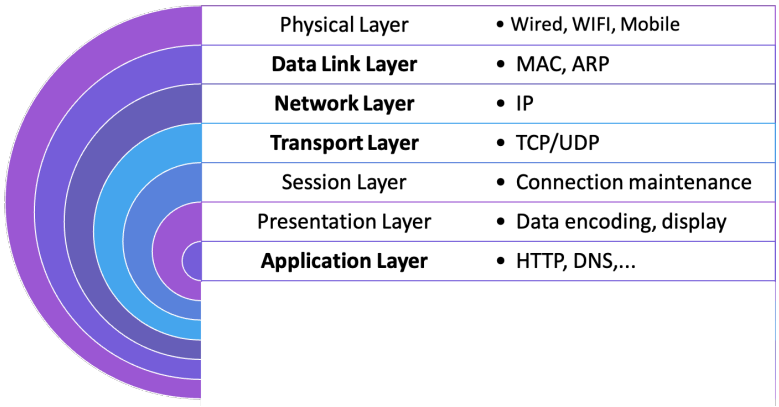

Figure 3.3: Demonstration of the OSI model.

clearly separating responsibilities such as application services, transport reliability, logical addressing, local delivery, and physical transmission.

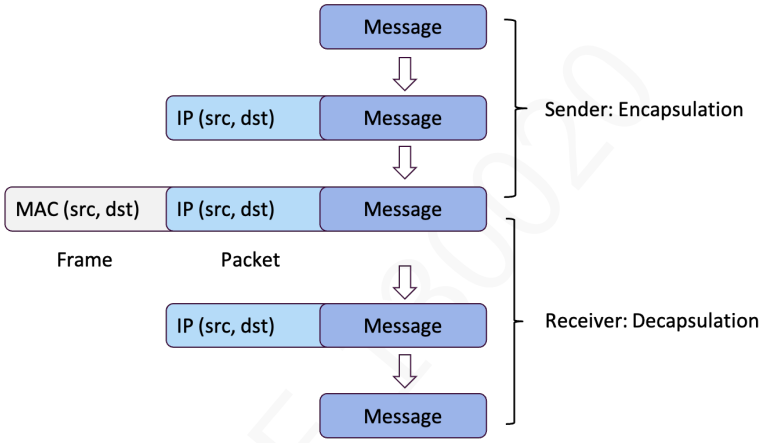

Figure 3.4: Demonstration of data encapsulation and decapsulation.

We have already introduced the MAC and IP which belong to the data link layer and network layer, respectively. To better understand how a message is sent using both MAC and IP addresses, Figure 3.4 provides a clear layered demonstration. The original data is first encapsulated with an IP header, which contains the source and destination IP addresses. Next, it is encapsulated with the source and destination MAC addresses. The MAC addresses are used for communication within the same local network, allowing devices to deliver frames directly to each other. Once the packet reaches a router or gateway, the MAC addresses are replaced for the next hop, while the IP addresses remain unchanged to guide the packet to its final destination.

3.2.2 Transmission Control Protocol

In practice, a network packet has a length limit, usually determined by the Maximum Transmission Unit (MTU) of the network, which is typically around 1500 bytes for Ethernet. If a message is larger than this limit, it cannot be sent in a single packet and must be segmented into multiple packets. A big problem arises concerning the integrity of the message: when a message is split into multiple packets, there is a risk that some packets may be lost, duplicated, or arrive out of order during transmission. Without a mechanism to handle these issues, the receiver might reconstruct the message incorrectly or miss parts of it entirely. This is where Transmission Control Protocol (TCP) plays a crucial role. TCP ensures reliable,

ordered delivery by:

- *Sequencing*: Assigning a sequence number to each segment so the receiver can reorder them correctly.
- *Acknowledgment*: The receiver sends an acknowledgment for every segment received, letting the sender know it arrived safely.
- *Retransmission*: If an acknowledgment is not received within a certain time, the sender retransmits the segment.
- *Error Checking*: TCP uses checksums in each segment to detect corruption, ensuring that only valid data is accepted.

By combining these mechanisms, TCP guarantees that the original message is reassembled accurately at the destination, preserving both its order and integrity, even over unreliable networks. Next, we employ Figure 3.5 to demonstrate how the first three designs are achieved.

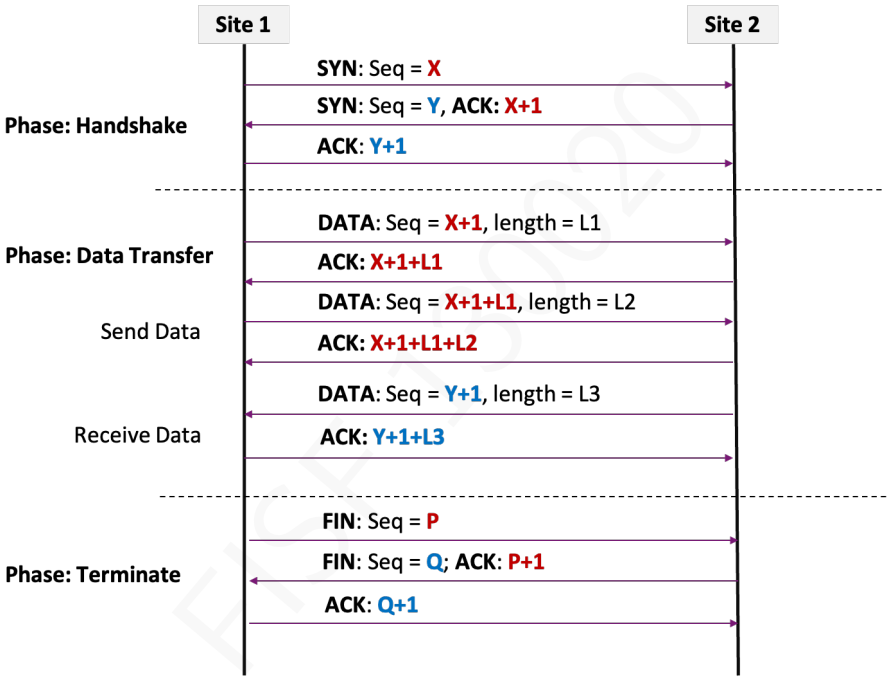

Figure 3.5: Demonstration of TCP.

There are three major phases in a TCP connection:

1) **Handshake:** This phase establishes a reliable connection between the sender and receiver. TCP uses a *three-way handshake*:
   - The client sends a `SYN` packet with a random sequence number X to the server to request a connection.
   - The server responds with a `SYN-ACK` packet with its own random sequence number Y and acknowledges the client's sequence number X by setting the acknowledgment number to $X+1$, indicating it expects the next byte to be $X+1$.
   - The client sends an `ACK` packet with acknowledgment number $Y+1$ back to the server, completing the handshake.

   After this phase, both sides know each other's sequence numbers and are ready to exchange data.

2) **Data Transfer:** In this phase, the actual message is transmitted. TCP divides the message into smaller segments, assigns a **sequence number** to each segment, and sends them reliably. Each sequence number is calculated as the previous sequence number plus the length of the segment in bytes. The receiver acknowledges each segment by sending an `ACK` with the sequence number of the next expected segment. If a segment is lost, corrupted, or received out of order, the sender retransmits it, ensuring that all segments are delivered in order and without errors.

3) **Connection Termination:** Once the communication is complete, TCP gracefully closes the connection to free resources. This typically involves a *four-way handshake*:

   - One side sends a `FIN` packet to indicate it has finished sending data.
   - The other side acknowledges with an `ACK`.
   - The second side then sends its own `FIN` packet.
   - Finally, the first side acknowledges with an `ACK`, completing the termination.

3.2.3 Network Applications

Above the transport layer, there are different applications. The most popular ones are Domain Name System (DNS) and Hypertext Transfer Protocol (HTTP). Application-layer protocols define the actual structure and meaning of the messages exchanged between programs. HTTP is the foundation of the World Wide Web. It enables browsers and servers to exchange web pages, images, and data through a request–response model. A client, such as a browser, sends a request, and the server replies with the requested content or an error status code. DNS acts like the Internet' s "phone book". Humans remember host names like `www.example.com`, but routers need IP addresses. DNS translates these names into IP addresses by querying a hierarchy of distributed name servers. Together, DNS and HTTP show how application protocols build on the transport layer to deliver real services to users.

Hands-on Exercise

1) **Packet Analysis with Wireshark.** Wireshark is a free, open-source protocol analyzer that captures and displays packets moving across a network. Download and install Wireshark from its official website at `https://www.wireshark.org`. Launch the program, select your active network interface, and start a live capture. Then access a chosen website in your browser and observe the traffic captured by Wireshark.

   - Check the sequence numbers and acknowledgements of TCP segments to understand reliable delivery.
   - Identify the HTTP or HTTPS packets generated when the page loads.
   - Filter for DNS packets and note the query and response for the website' s domain name.