Networking

Nathan Warner

Northern Illinois
University

Computer Science
Northern Illinois University

April 22, 2024
United States

Contents

A brief overview

In the context of network programming, ”networking” refers to the practice and techniques involved in designing, implementing, and managing communication between computers and devices over a network. This can include a wide array of tasks and principles, including but not limited to:

- **Data Communication:** The fundamental aspect of networking, involving the exchange of data between two or more devices over a network. This can be achieved through various communication protocols and standards.
- **Protocols and Standards:** Networking relies on a set of rules and conventions (protocols) for communication between network devices. These protocols define how data is formatted, transmitted, and received. Examples include TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (HyperText Transfer Protocol), and FTP (File Transfer Protocol).
- **Network Architecture:** The design and layout of a network, including its components (e.g., routers, switches, gateways) and topology (e.g., star, mesh, ring). Network architecture decisions impact the network's performance, scalability, and security.
- **Socket Programming:** A means of connecting two nodes on a network to communicate with each other. One node listens on a particular port at an IP, while another node connects to it. Socket programming is used to facilitate communication between applications running on different computing devices.
- **APIs for Network Communication:** Programming interfaces such as Winsock for Windows, POSIX sockets for Unix/Linux, and various cross-platform networking libraries (e.g., Boost.Asio) that allow developers to implement networking functionalities.
- **Network Services Development:** Creating software that provides specific functionalities over a network, such as web servers, email servers, and file sharing systems.
- **Network Security:** Ensuring the confidentiality, integrity, and availability of data in the network. This includes implementing secure protocols (like HTTPS), encryption, firewalls, and intrusion detection systems.
- **Network Management:** Monitoring and maintaining network operations. This involves performance analysis, troubleshooting network problems, and ensuring that network resources are allocated efficiently.

Network Terminology

2.1 Nodes, links, and paths

node refers to any device that can send, receive, or forward information over a communications channel. Nodes can be computers, mobile devices, routers, switches, and other devices capable of processing or storing data.

A link, on the other hand, is the physical or logical connection between two or more nodes, enabling them to communicate. Links can be wired connections like Ethernet cables, optical fibers, or wireless connections such as Wi-Fi or Bluetooth.

Together, nodes and links form the basic components of a network, allowing for the transmission of data across diverse and complex systems.

A path is a sequence of nodes and links

In other words...

- **Node:** Host or intermediary
- **Link:** Point-to-point or broadcast to many other nodes at the same time
- **Link medium:** wired or wireless
- **Path:** Routed or switched (Elaborated in later section)

2.2 Networking protocol

Networking protocols are standardized sets of rules that determine how data is transmitted and received across a network. These protocols specify the formats for data packets, the procedures for signaling, error handling, and data encryption to ensure successful communication between devices.

More broadly, information is exchanged between nodes via **messages**, each message has an exact meaning intended to provoke a defined response of the reciever

Note:-

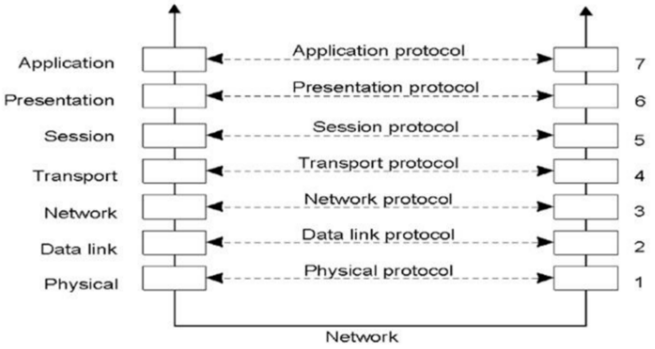
Messages used **well-defined format**

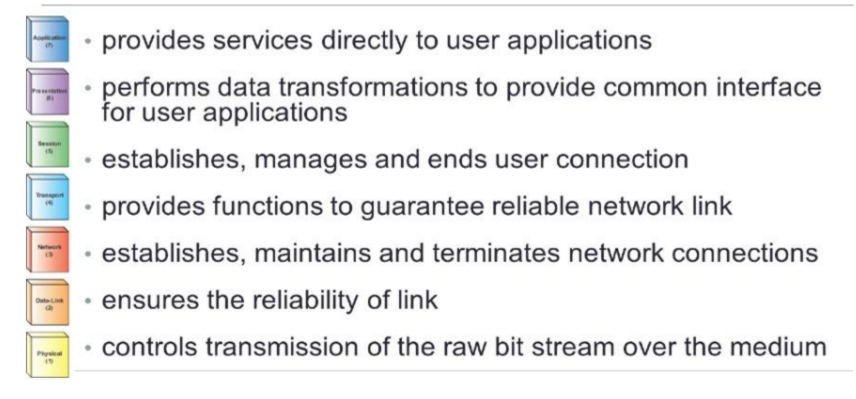
The OSI (Opens systems interconnection) Model

The OSI (Open Systems Interconnection) model is a conceptual framework used to understand and standardize the functions of a telecommunications or computing system without regard to its underlying internal structure and technology. Developed by the International Organization for Standardization (ISO), the OSI model divides the process of communication between two end-points in a network into seven layers. Each layer serves a specific function and communicates with the layers directly above and below it. From top to bottom, the layers are **APS TNDP**:

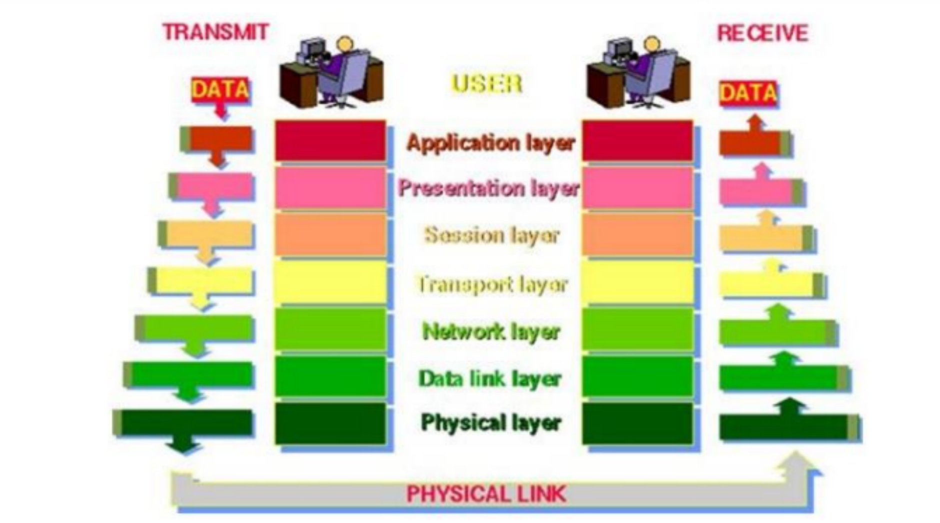
- **Application Layer (Layer 7):** The closest to the end user, this layer interacts with software applications that implement a communicating component. It provides protocols that allow software to send and receive information and present meaningful data to the user (e.g., HTTP for web browsing, SMTP for email).
- **Presentation Layer (Layer 6):** Translates data between the application layer and the network format. It ensures data is in a usable format and can encrypt or compress data if necessary.
- **Session Layer (Layer 5):** Manages sessions between applications, establishing, managing, and terminating connections between local and remote applications.
- **Transport Layer (Layer 4):** Responsible for data transfer between end systems and provides reliable data transfer services to the upper layers. This includes breaking down messages into smaller units if needed, and ensuring error-free data transfer (e.g., TCP and UDP).
- **Network Layer (Layer 3):** Manages device addressing, identifies the best paths for data transmission, and routes data packets between devices that are not locally attached. Routers operate at this layer.
- **Data Link Layer (Layer 2):** Provides data transfer across the physical link established by the physical layer. It deals with MAC addresses, error detection and correction, and frames data packets.
- **Physical Layer (Layer 1):** Concerns the physical equipment involved in data transfer, such as cables, switches, and the electrical signals that traverse these media.

In other words, these seven layers help to describe communications in a network




Application (7)
Presentation (6)
Session (5)
Transport (4)
Network (3)
Data Link (2)
Physical (1)

- provides services directly to user applications
- performs data transformations to provide common interface for user applications
- establishes, manages and ends user connection
- provides functions to guarantee reliable network link
- establishes, maintains and terminates network connections
- ensures the reliability of link
- controls transmission of the raw bit stream over the medium



3.1 Main idea

The complexities of communication is organized into successive layers of protocols

- **Lower-level layers**: Specific to medium
- **Higher-level layers**: Specific to application

3.2 Physical layer: Wired media

- **Ethernet**
  - 10BASE-T, 100BASE-T, 1000BASE-T
  - 10GbE, 40GbE, 100GbE
- **Business/backbone**
  - DS1(T1): 1.54Mbs to DS5: 400Mbs
  - OC-1: 50Mbs to OC-768: 40Gbs
- **Last mile:**
  - Modem
  - DSL (Digital subscriber lines)
  - Cable: DOCSIS
  - FiOS (Fiber optic service)

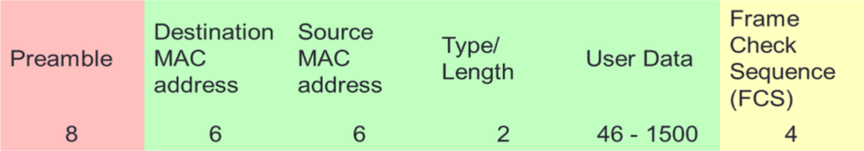
3.3 Physical layer: Wireless media

- **Cellphone Data:**
  - EDGE, GPRS, HSPA+
  - 4G LTE up to 100Mbs
  - 5G over 100Mbs
- **Satellite**
  - Wildblue: 12Mbs
  - Hughesnet: 15Mbs
  - Starlink: 200Mbs
- **WiFi:** 802.11
  - Up to 150Mbs & MIMO
  - New: ”ac” up to 1Gbs
- **WiMax:** 802.16
  - up to 40Mbs
- **WPAN**
  - Bluetooth up to 2Mbs
  - NFC up to 423Kbs
  - ZigBee up to 256Kbs

3.4 Data link layer: Functionalities

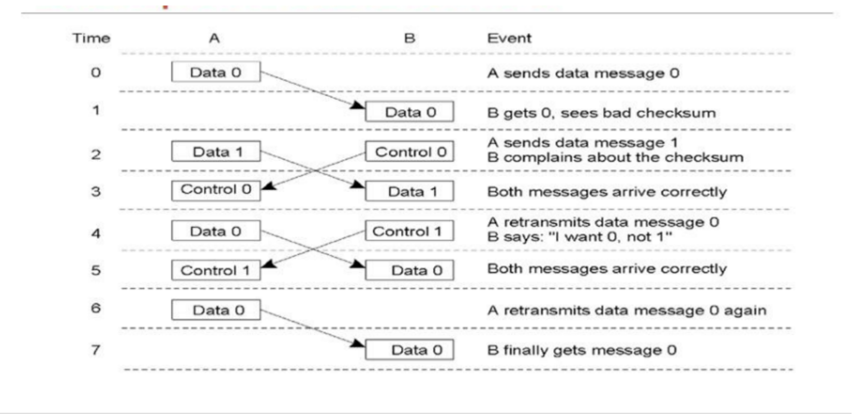
- **Medium access control**: Arbitrate who transmits
- **Addressing**: address of receiver, address of sender
- **Framing:** Delimited unit of transmission for data & control
- **Error control and reliability**
- **Flow control**

3.4.1 Example: Ethernet frame



Preamble	Destination MAC address	Source MAC address	Type/ Length	User Data	Frame Check Sequence (FCS)
8	6	6	2	46 - 1500	4

3.4.2 Example: Data link flow



Time	A	B	Event
0	Data 0		A sends data message 0
1		Data 0	B gets 0, sees bad checksum
2	Data 1	Control 0	A sends data message 1 B complains about the checksum
3	Control 0	Data 1	Both messages arrive correctly
4	Data 0	Control 1	A retransmits data message 0 B says: "I want 0, not 1"
5	Control 1	Data 0	Both messages arrive correctly
6	Data 0		A retransmits data message 0 again
7		Data 0	B finally gets message 0

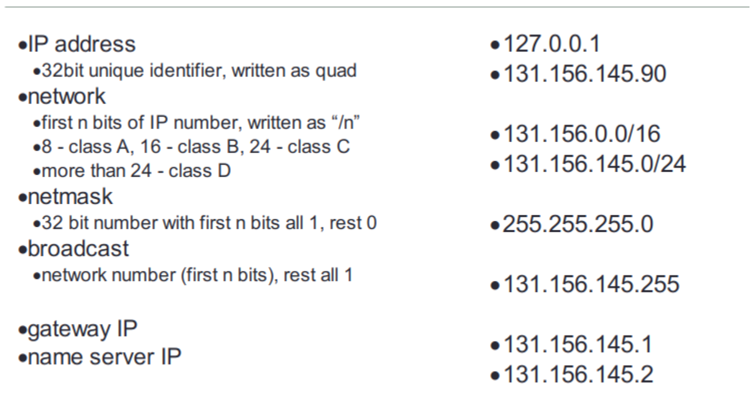
3.5 Network layer (Internet protocol layer)

Provides host to host transmission service, where hosts are not necessarily adjacent

Layer provides services

- Addressing
  - Hosts have global addresses: IPv4, IPv6
  - Uses data link layer protocol to translate address
- **Routing and forwarding:** Find path from host to host

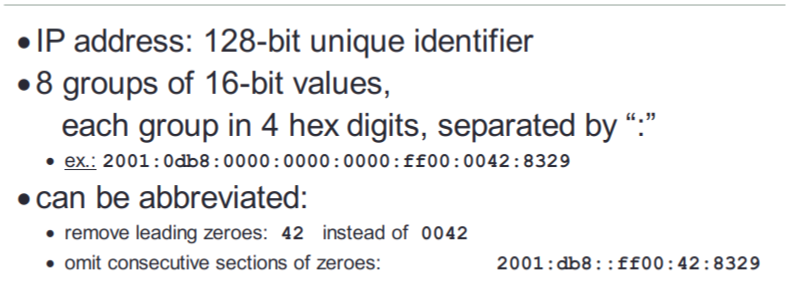
3.5.1 IPv4 Address



- IP address
  - 32bit unique identifier, written as quad
- network
  - first n bits of IP number, written as "/n"
  - 8 - class A, 16 - class B, 24 - class C
  - more than 24 - class D
- netmask
  - 32 bit number with first n bits all 1, rest 0
- broadcast
  - network number (first n bits), rest all 1
- gateway IP
- name server IP

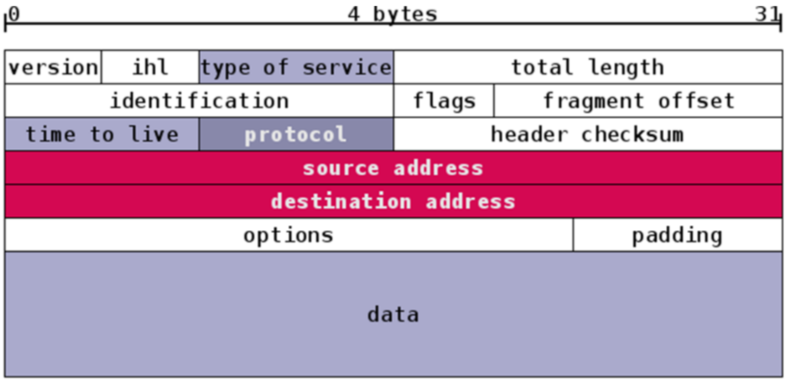
- 127.0.0.1
- 131.156.145.90
- 131.156.0.0/16
- 131.156.145.0/24
- 255.255.255.0
- 131.156.145.255
- 131.156.145.1
- 131.156.145.2

3.5.2 IPv6 Address



- IP address: 128-bit unique identifier
- 8 groups of 16-bit values,
  each group in 4 hex digits, separated by ":"
  - ex.: `2001:0db8:0000:0000:0000:ff00:0042:8329`
- can be abbreviated:
  - remove leading zeroes: `42` instead of `0042`
  - omit consecutive sections of zeroes: `2001:db8::ff00:42:8329`

3.5.3 IP Packet



0 4 bytes 31

version | ihl | type of service | total length

identification | flags | fragment offset

time to live | protocol | header checksum

source address

destination address

options | padding

data

3.6 IP Layer: Routing and Forwarding

Done by hosts on path from sending to reciever

- **Forwarding:** Host has 2 network interfaces, transfers packet from incoming to outgoing interface
- **Routing:**
  - Finds path from sender to receiver
  - **Simple routing:** know receiver or send to gateway
  - **Advanced routing:** determine which gateway to send to (typically with multiple outgoing network interfaces)

3.7 Transport layer

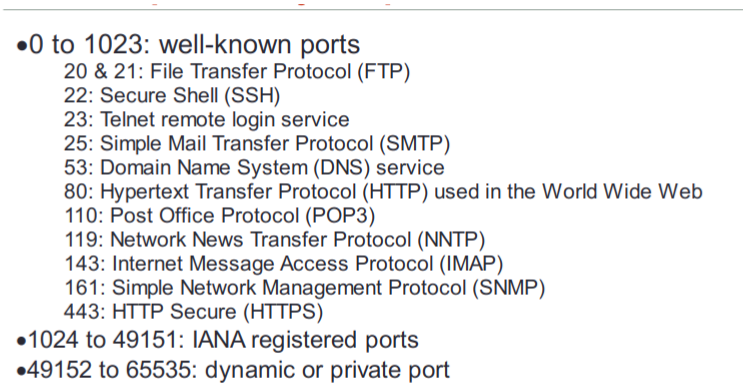
Provides end-to-end communication services for applications

Btye format as abstraction on underlying system format. Raises reliability

Also enables multiplexing, which provides multiple endpoints on a single node: **port**.

Refines connection address via port number

3.8 Ports

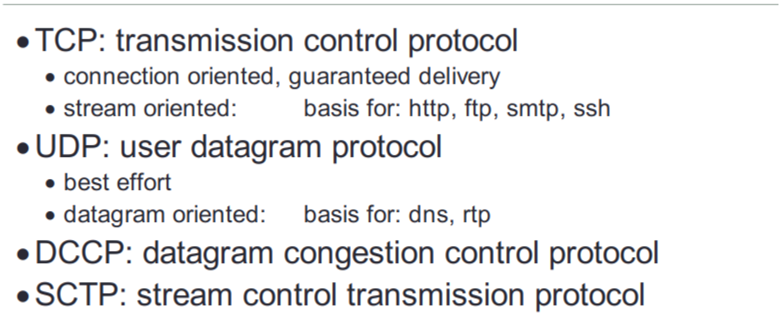


- 0 to 1023: well-known ports
  - 20 & 21: File Transfer Protocol (FTP)
  - 22: Secure Shell (SSH)
  - 23: Telnet remote login service
  - 25: Simple Mail Transfer Protocol (SMTP)
  - 53: Domain Name System (DNS) service
  - 80: Hypertext Transfer Protocol (HTTP) used in the World Wide Web
  - 110: Post Office Protocol (POP3)
  - 119: Network News Transfer Protocol (NNTP)
  - 143: Internet Message Access Protocol (IMAP)
  - 161: Simple Network Management Protocol (SNMP)
  - 443: HTTP Secure (HTTPS)
- 1024 to 49151: IANA registered ports
- 49152 to 65535: dynamic or private port

3.8.1 Transport layer programming

- **Common abstraction:** Socket
- Socket is end-point of communication link
  - Identified as Ip address + port number
- operates as client and server

3.8.2 Transport layer protocols



- TCP: transmission control protocol
  - connection oriented, guaranteed delivery
  - stream oriented: basis for: http, ftp, smtp, ssh
- UDP: user datagram protocol
  - best effort
  - datagram oriented: basis for: dns, rtp
- DCCP: datagram congestion control protocol
- SCTP: stream control transmission protocol