Chapter 1
Communication Networks and Services

What is a communication network?

• A communication network is a set of equipment (hardware & software) and facilities that provide the basic communication service
• A communication service enables the exchange of information between users at different locations.
• Communication services & applications are everywhere.
Network Architecture Evolution

• Telegraph Networks
  – Message switching & digital transmission

• Telephone Networks
  – Circuit Switching
  – Analog transmission → digital transmission
  – Mobile communications

• Internet
  – Packet switching & computer applications

• Next-Generation Internet
  – Multiservice packet switching network

Circuit Switching

• Three phases
  – Establish
  – Transfer
  – Disconnect

• Dedicated communication path between two stations
Packet Switching

• Data transmitted in packets
  – Longer messages split into small packets
  – Each packet contains a portion of user data plus some control info (e.g. addressing info)
• Packets are received, stored and past on to the next node
  – Store and forward
• Different paths can be used to get packets to their destination.

Circuit vs Packet Switching

• Circuit Switching
  – Network is used for the entire duration of the call. Inefficient.
  – Routing is done at call setup. Relatively easy
  – Suited for voice traffic
• Packet Switching
  – Network is used on demand. Efficient.
  – Routing is difficult.
  – Suited for data traffic
Chapter 2
Applications and Layered Architectures

Layers, Services & Protocols

• The overall communications process between two or more machines connected across one or more networks is very complex
• **Layering** partitions related communications functions into groups that are manageable
• Each layer provides a **service** to the layer above
• Each layer operates according to a **protocol**
Protocols

- A protocol is a set of rules that governs how two or more communicating entities in a layer are to interact
- Messages that can be sent and received
- Actions that are to be taken when a certain event occurs, e.g. sending or receiving messages, expiry of timers
- The purpose of a protocol is to provide a service to the layer above

Layers

- A set of related communication functions that can be managed and grouped together
- Application Layer: communications functions that are used by application programs
  - HTTP, DNS, SMTP (email)
- Transport Layer: end-to-end communications between two processes in two machines
  - TCP, User Datagram Protocol (UDP)
- Network Layer: node-to-node communications between two machines
  - Internet Protocol (IP)
Why Layering?

- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts
- Protocol in each layer can be designed separately from those in other layers
- Layering provides flexibility for modifying and evolving protocols and services without having to change layers below
- Monolithic non-layered architectures are costly, inflexible, and soon obsolete

Example: HTTP

- HTTP is an application layer protocol
- Retrieves documents on behalf of a browser application program
- HTTP specifies fields in request messages and response messages
  - Request types; Response codes
  - Content type, options, cookies, …
- HTTP specifies actions to be taken upon receipt of certain messages
HTTP Protocol

- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server.
- In fact, HTTP client and server are processes running in two different machines across the Internet.
- HTTP uses the reliable stream transfer service provided by TCP.

Example: TCP

- TCP is a transport layer protocol.
- Provides *reliable byte stream service* between two processes in two computers across the Internet.
- TCP is *connection-oriented*: the sender and receiver must first establish an association and set initial sequence numbers before data is transferred.
- Connection ID is specified uniquely by `(send port #, send IP address, receive port #, receiver IP address)`.
Example: DNS Protocol

- DNS protocol is an application layer protocol
- DNS is a distributed database that resides in multiple machines in the Internet
- DNS protocol allows queries of different types
  - Name-to-address or Address-to-name
  - Mail exchange
- DNS usually involves short messages and so uses service provided by UDP
- Well-known port 53

Example: UDP

- UDP is a transport layer protocol
- Provides *best-effort datagram service* between two processes in two computers across the Internet
- Port numbers distinguish various processes in the same machine
- UDP is *connectionless*
- Datagram is sent immediately
- Quick, simple, but not reliable
Layers, Services & Protocols

- Layers: related communications functions
  - Application Layer: HTTP, DNS
  - Transport Layer: TCP, UDP
  - Network Layer: IP

- Services: a protocol provides a communications service to the layer above
  - TCP provides connection-oriented reliable byte transfer service
  - UDP provides best-effort datagram service

- Each layer builds on services of lower layers
  - HTTP builds on top of TCP
  - DNS builds on top of UDP
  - TCP and UDP build on top of IP

OSI Reference Model

- By the 1970s every computer vendor had developed its own proprietary layered network architecture
- Problem: computers from different vendors could not be networked together
- Open Systems Interconnection (OSI) was an international effort by the International Organization for Standardization (ISO) to enable multivendor computer interconnection
OSI Reference Model

- Describes a seven-layer abstract reference model for a network architecture
- Purpose of the reference model was to provide a framework for the development of protocols
- OSI also provided a unified view of layers, protocols, and services which is still in use in the development of new protocols
- Detailed standards were developed for each layer, but most of these are not in use
- TCP/IP protocols preempted deployment of OSI protocols
Physical Layer

• Transfers bits across link
• Definition & specification of the physical aspects of a communications link
  – Mechanical: cable, plugs, pins...
  – Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  – functional/procedural: how to activate, maintain, and deactivate physical links…

Data Link Layer

• Transfers *frames* across *direct* connections
• Groups bits into frames
• Detection of bit errors; Retransmission of frames
• Activation, maintenance, & deactivation of data link connections
• Medium access control for local area networks
• Flow control
Network Layer

- Transfers *packets* across multiple links and/or multiple networks
- Addressing must scale to large networks
- Nodes *jointly* execute routing algorithm to determine paths across the network
- Forwarding transfers packet across a node
- Congestion control to deal with traffic surges
- Connection setup, maintenance, and teardown when connection-based

Transport Layer

- Transfers data end-to-end from process in a machine to process in another machine
- Reliable stream transfer or quick-and-simple single-block transfer
- Port numbers enable multiplexing
- Message segmentation and reassembly
- Connection setup, maintenance, and release
Application & Upper Layers

- Application Layer: Provides services that are frequently required by applications: DNS, web access, file transfer, email…
- Presentation Layer: Machine-independent representation of data…
- Session Layer: Dialog management, recovery from errors, …

TCP/IP Protocol Suite

- Reliable stream service: TCP
- Best-effort connectionless packet transfer: UDP
- Distributed applications: HTTP, SMTP, DNS, RTP
- User datagram service: (ICMP, ARP)
- Diverse network technologies: Network interface 1, Network interface 2, Network interface 3
Headers & Trailers

- Each protocol uses a header that carries addresses, sequence numbers, flag bits, length indicators, etc…
- CRC check bits may be appended for error detection

Encapsulation

TCP Header contains source & destination port numbers

IP Header contains source and destination IP addresses; transport protocol type

Ethernet Header contains source & destination MAC addresses; network protocol type
OSI Unified View

- Layer n in one machine interacts with layer n in another machine to provide a service to layer n +1
- The entities comprising the corresponding layers on different machines are called peer processes.
- The machines use a set of rules and conventions called the layer-n protocol.
- Layer-n peer processes communicate by exchanging Protocol Data Units (PDUs)

![Diagram showing layer n peer protocol with n-PDUs](image)

OSI Unified View

- Communication between peer processes is virtual and actually indirect
- Layer n+1 transfers information by invoking the services provided by layer n
- Services are available at Service Access Points (SAP’s)
- Each layer passes data & control information to the layer below it until the physical layer is reached and transfer occurs
- The data passed to the layer below is called a Service Data Unit (SDU)
- SDU’s are encapsulated in PDU’s
Connectionless & Connection-Oriented Services

- **Connection-Oriented**
  - Three-phases:
    1. Connection setup between two SAPs to initialize state information
    2. SDU transfer
    3. Connection release
  - E.g. TCP, ATM

- **Connectionless**
  - Immediate SDU transfer
  - No connection setup
  - E.g. UDP, IP

- Layered services need not be of same type
  - TCP operates over IP
  - IP operates over ATM