Networks

A network is composed of

- Transmission media
- Hardware devices
- Software components

We call the hardware devices and software components collectively a “communication subsystem”.

Devices and computers that use the network for communication purposes are called hosts.

The term node refers to any computer or switching device attached to the network.
The Internet

- The Internet is a single communication subsystem split into distinct subnetworks (subnets) that act as routing units.
Requirements of Networks

- What sorts of requirements typically get imposed on networks?

- Performance
  - Latency and data transmission rates

- Scalability
  - Tolerate addition of many nodes

- Reliability

- Security

- Mobility

- Quality of Service
  - Performance guarantees/bounds.

- Multicasting
  - Facilitate more than point-to-point pair wise transfers.
Types of networks

- Personal area networks
  - Bluetooth
- Local area networks (LAN)
  - University or corporate network
- Wide area networks (WAN)
- Metropolitan area networks (MAN)
  - DSL/Cable networks
- Wireless LAN/WAN/MAN
  - Cellular data networks
- Internetworks
  - Multiple interoperating networks connected via routers or gateway devices.
Network principles

- What are some of the important principles in network technologies?
  - Packet transmission
  - Data streaming
  - Switching schemes
  - Protocols
    - Software layers and protocol “stacks”
  - Routing algorithms and hardware
  - Congestion control
  - Internetworking
Packet transmission and streaming

- Data is moved from one point to another in two different ways typically.
  - Packets: Data moves through the network in fixed length packets that contain the contents of the message, and information relevant for routing and transport protocols.
  - Streaming: Data flows through the network in a more continuous fashion with guarantees and bounds on performance measures like latencies. Popular for multimedia applications.
Switching schemes

- **Broadcast**
  - Send to everyone, rely on intended receiver to accept message and others to toss it out.
  - Example: Ethernet

- **Circuit switching**
  - Inspired by old telephone switching techniques.
  - A path through the network is set up and data streams through.

- **Packet switching**
  - Store-and-forward

- **Frame relay**
  - Similar to circuit switching + packet switching, minus store-and-forward
    - Store only enough of packet to make routing decision and then pass data through like a circuit switched network.
Protocols

- A protocol defines a set of rules and formats used when processes communicate with each other.
- Two important parts:
  - 1. Specification of the sequence of messages that are to be exchanged.
  - 2. Specification of the format of these messages.

- Protocols exist for many different levels of abstraction
  - Low-level blocks of bits on the wire
  - High-level, application specific abstractions
    - HTTP, SMTP, FTP, etc…
Protocol layers

- Network software protocols are separated into well-defined layers.

![Diagram showing protocol layers with a message sent from Layer n to Layer 2, then to Layer 1, and finally a message received on the recipient side via the communication medium.]

Protocol layers

- The layers are based on abstractions.
- Low levels: Abstraction above the wire protocol.
  - E.g.: Ethernet
- Middle levels: Abstraction of raw data transmission and reliability.
  - E.g.: TCP, IP, UDP
- Higher levels: Abstractions related to specific activities.
  - E.g.: FTP, HTTP
- Highest levels: Abstractions related to specific applications.
  - E.g.: Web services
Standard stack: ISO OSI

- Open Systems Interconnection stack, standardized by ISO.
- 7 layers
  - Application
  - Presentation
  - Session
  - Transport
  - Network
  - Data link
  - Physical
ISO OSI

Layers

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

Sender

Communication medium

Recipient

Message sent

Message received
<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service.</td>
<td>HTTP, FTP, SMTP, CORBA IIOP</td>
</tr>
<tr>
<td>Presentation</td>
<td>Protocols at this level transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required.</td>
<td>Secure Sockets (SSL), CORBA Data Rep.</td>
</tr>
<tr>
<td>Session</td>
<td>At this level reliability and adaptation are performed, such as detection of failures and automatic recovery.</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>Transport</td>
<td>This is the lowest level at which messages (rather than packets) are handled. Messages are addressed to communication ports attached to processes. Protocols in this layer may be connection-oriented or connectionless.</td>
<td>IP, ATM virtual circuits</td>
</tr>
<tr>
<td>Network</td>
<td>Transfers data packets between computers in a specific network. In a WAN or an internetwork this involves the generation of a route passing through routers. In a single LAN no routing is required.</td>
<td>Ethernet MAC, ATM cell transfer, PPP</td>
</tr>
<tr>
<td>Data link</td>
<td>Responsible for transmission of packets between nodes that are directly connected by a physical link. In a WAN transmission is between pairs of routers or between routers and hosts. In a LAN it is between any pair of hosts.</td>
<td>Ethernet base-band signalling, ISDN</td>
</tr>
<tr>
<td>Physical</td>
<td>The circuits and hardware that drive the network. It transmits sequences of binary data by analogue signalling, using amplitude or frequency modulation of electrical signals (on cable circuits), light signals (on fibre optic circuits) or other electromagnetic signals (on radio and microwave circuits).</td>
<td></td>
</tr>
</tbody>
</table>
Protocol stacks: Pros and Cons

Pro:
- Abstraction allows for different layers to be swapped in and out.
- Separation of concerns.

Con:
- Performance degradation. All messages must traverse stack on both receiver and sending side.
- Results in observed performance in applications being less than the advertised performance of the hardware.
Encapsulation

- Lower levels encapsulate messages from higher levels.
Encapsulation

- What is nice about encapsulation?
- The message being encapsulated isn’t treated as anything more than data by the wrapper.
- This allows you to design layers without having to specify much at all for the other layers.

- We see this in action daily:
  - HTTP over TCP/IP over ethernet.
  - FTP over TCP/IP over ethernet.
  - AppleTalk over ethernet.
    - Ok, this one is a bit old. 😊 Just an example of multiple protocols over ethernet.
Ports and addressing

- The transport layer is responsible for providing ports and addresses.

- Ports are essentially mailboxes or addressing units on a single machine that is associated with a specific program.
  - Program “foo” wants to receive data, so it opens up port 1234 from the outside world.
  - Outside processes connect to 1234, and don’t have to know much about the process that is listening on the port.
    - Other than the protocol/interaction pattern that it expects.

- Addresses allow hosts to find each other.

- The combination of a port and a host address allows processes to find other processes in a well defined, organized way.
Routing

Routing algorithms and hardware are what determine the path packets take through the network to get from the originator to the receiver.

The network is considered to be a graph.
- The nodes are routers, switches and hosts.
- The edges are the links between them.
- The edges are typically weighted with a cost of traversing the path based on performance characteristics of the link.

A route is a path through this graph.

Routes can be computed statically or dynamically.
- Adaptive routing is a dynamic scheme that addresses changing conditions
  - Such as load variations and failures within the network.
- The internet is based on adaptive routing for the most part.
Routing algorithms

- In ch. 3.3, you will see an example of a simple algorithm for routing.

- Tables are maintained at routers that represent the link to take to get closer to the destination. Traversing a link is often called taking a “hop”.

- Routers periodically update their tables based on conditions changing, and exchange them with each other.

- The routing algorithm must deal with undesirable states such as forming loops, and routing around parts of the network that vanish.