Protocol Layering

Protocol layering allows us to divide a complex task into several smaller and simpler tasks

- **A layer (module)** is a black box with inputs and outputs without concerns about the conversion of inputs to outputs

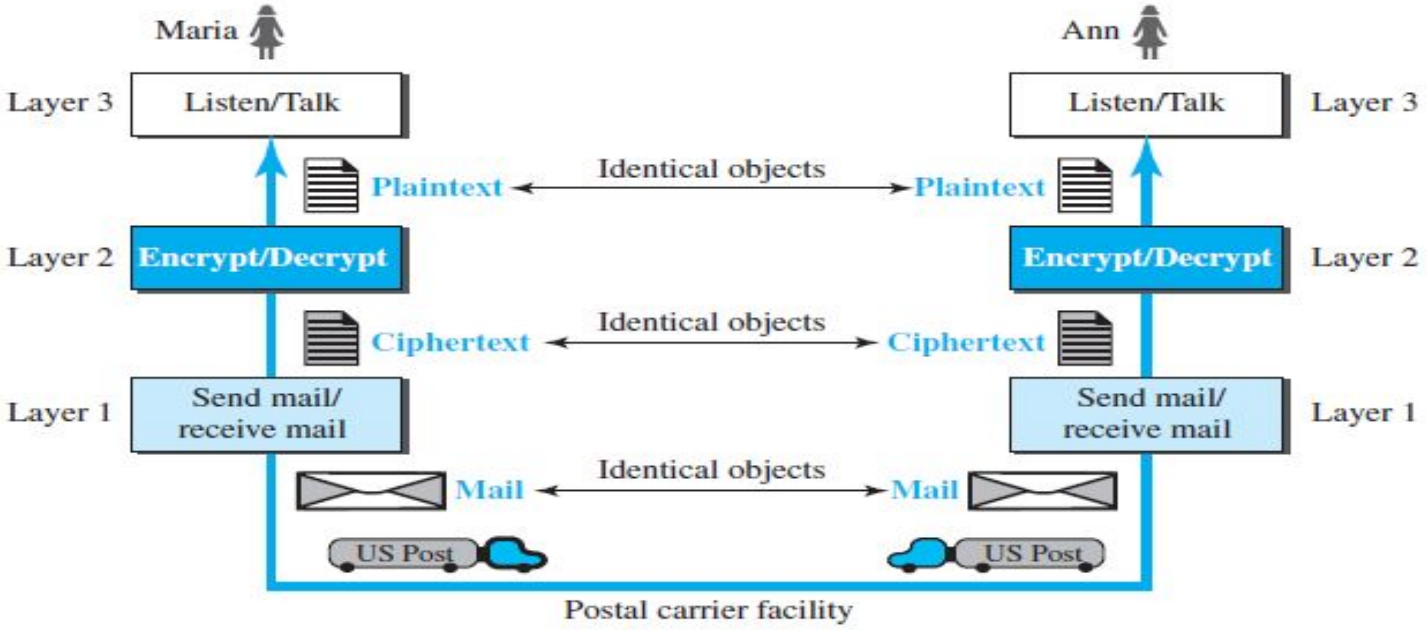
- A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer
  - We do not care about how the layer is implemented

- There are also **intermediate systems** other than **end-systems** used by **communication** with respect to protocol layering in the **Internet**

Protocol Layering

A simples scenario (only 1 layer)

A bit complex scenario (many layers)



Principles of Protocol layering

First principle

- For bidirectional communication: Each layer should be able to perform **two opposite tasks**
  - 3rd layer task is to ***talk*** (in one direction) and ***listen*** (in other direction)
  - 2nd layer ***encrypt*** (on one side) and ***decrypt*** (other side)
  - 1st layer, ***send*** and ***receive*** email

Principles of Protocol layering

Second principle

- Two objects under each layer at both sides should be identical
  - Objects under layer 3 : plain text
  - Objects under layer 2: Cipher text
  - Objects under layer 1: a piece of mail

Models of Layers

Principals Create a layer where a different abstraction is needed

- Each layer should perform a well-defined function
- The function of each layer should be chosen
  - With target for standardization
- Minimize information flow across layer boundaries
- Number of layers
  - Large: distinct functions are not in the same layer
  - Small : architecture does not become complex

INTERFACE BETWEEN LAYERS

- The passing of the data and network information down through the layers of the sending device and back up through the layers of the receiving device is made possible by an interface between each pair of adjacent layers.

- Each interface defines the information and services a layer must provide for the layer above it.

- Well-defined interfaces and layer functions provide modularity to a network.

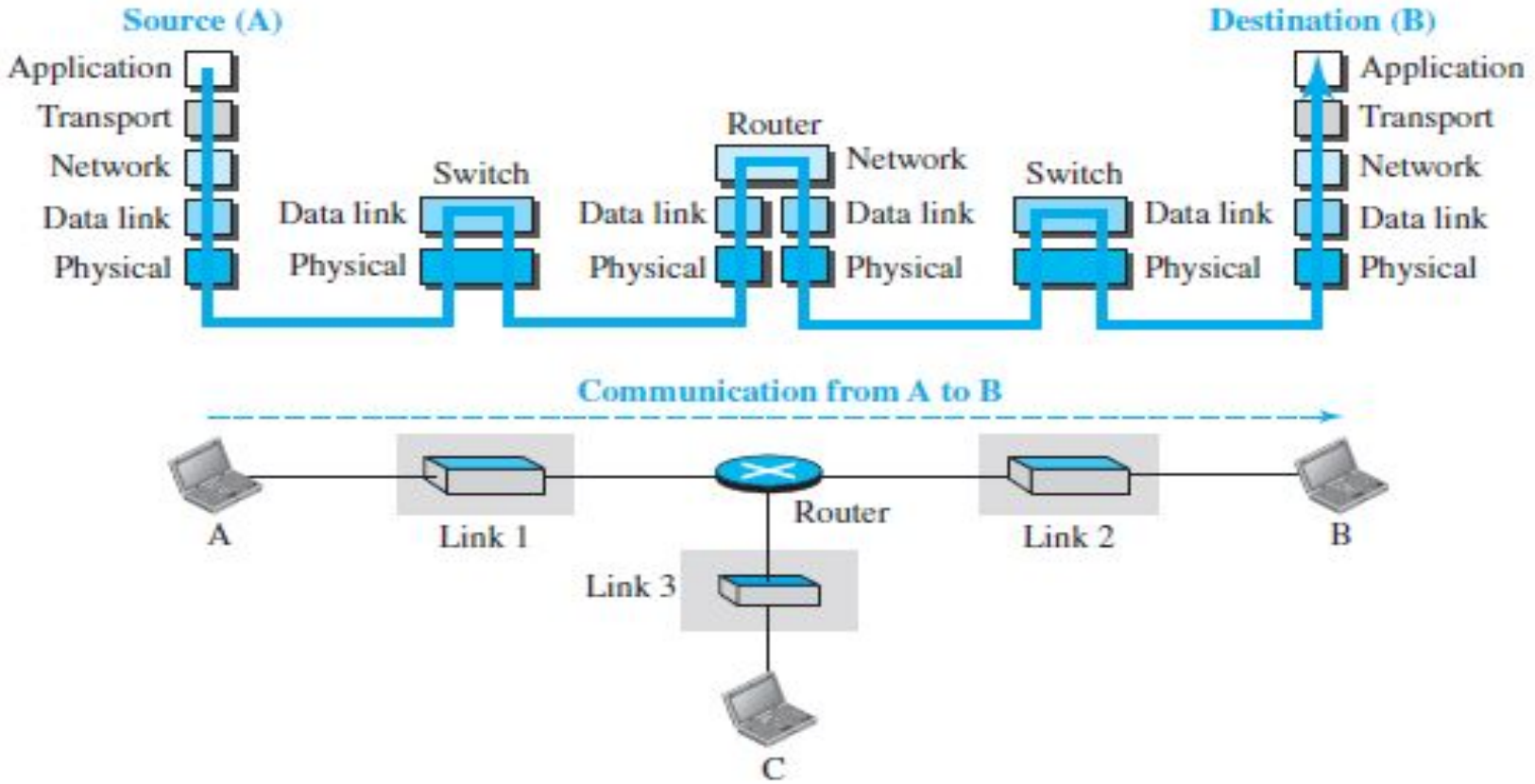
Reference Models

Layers model

- A theoretical approach describing how data is going to be transmit over the network
  - The TCP/IP Protocol Suit
    - Used in all WANs, the ARPANET, worldwide Internet
  - The OSI Reference Model (minus physical medium)

TCP/IP Layered Architecture

- Consider a small internet of three LANs (each with a link layer switch).
- Further, assume that links are connected by one router



TCP/IP Layered Architecture

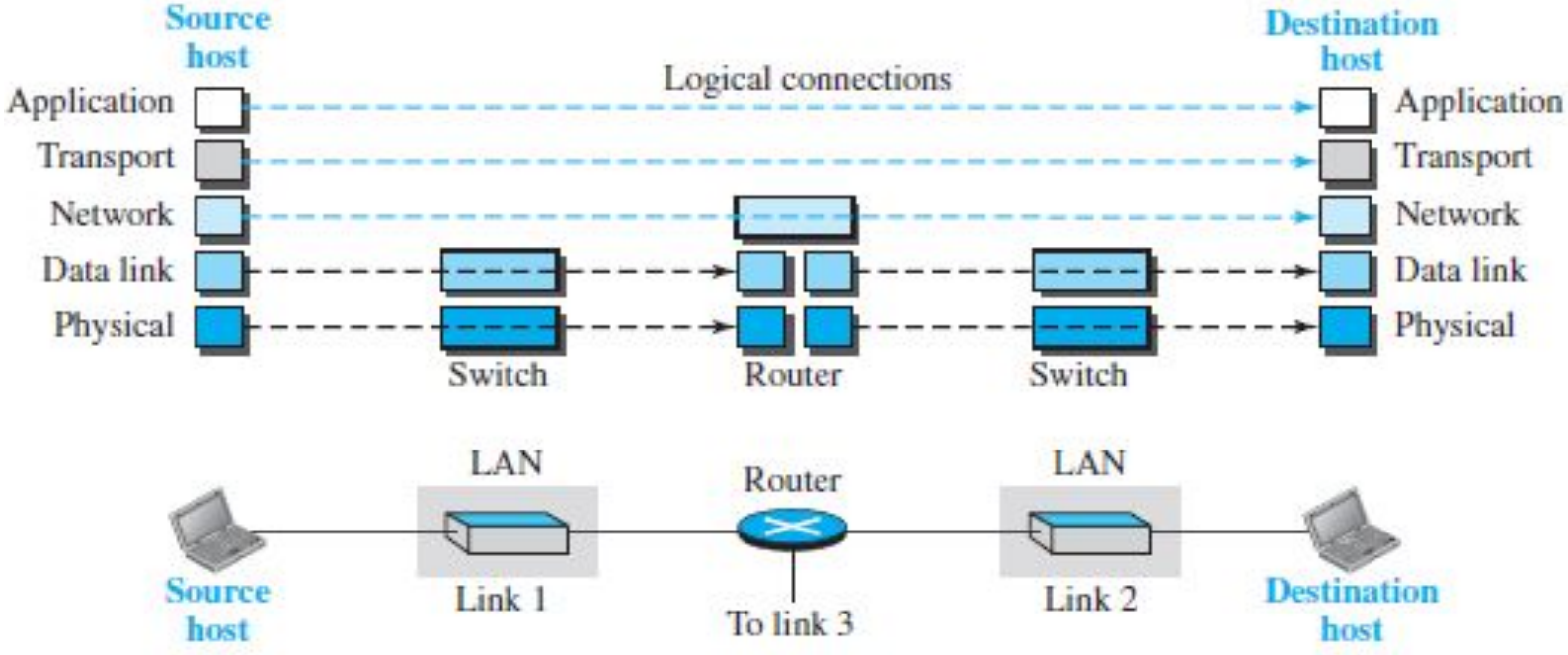
- All the hosts are involved in all five layers
- **Source host** creates a message in the application layer and sends it down the layer for physical delivery to the destination host
- **Destination host** receives the communication at physical and then deliver to the application layer through other layers

- **Router** is involved three layers
  - Involved in only **one network layer**; but involved in **n** (number of links) combinations of link and physical layers
  - Each link may use its own data link and physical protocol

- Link layer **switch** is **involved** in only two layers
  - It is involved only in one data-link and one physical layer

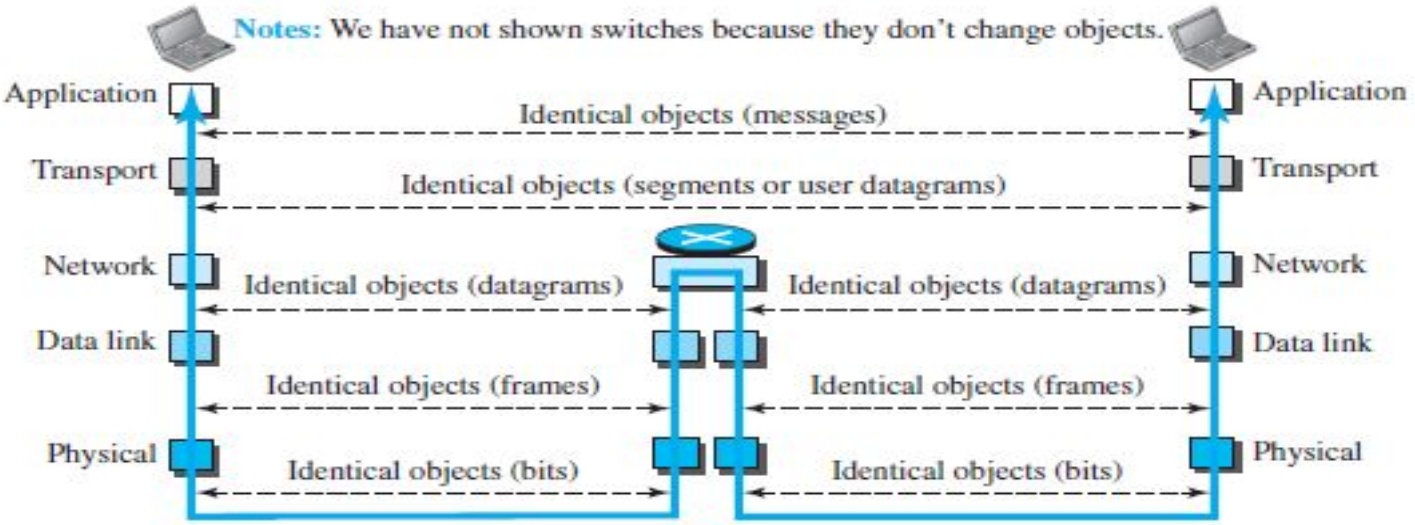
LAYERS IN THE TCP/IP Protocol Suit (Book2:2.2

- **Logical connections** make easier to think about the duty of each layer.
- The duty of the application, transport, and network layers is **end-to-end (so domain is internet)**.
- However, the duty of the data-link and physical layers is **hop-to-hop**, a hop being a host or router (**domain is link)**.



LAYERS IN THE TCP/IP Protocol Suit

- **Logical connections may be thought of as data unit created at each layer**, a hop being a host or router.
- In **top three layers**
  - Data unit (packets) should not be changed by any router or link-layer switch.
- **In the bottom two layers**,
  - the packet created by the host is changed only by the routers, not by the link-layer switches.



Description of Layers

PHYSICAL LAYER IN THE TCP/IP

- ***responsible for carrying individual bits in a frame across the link.***

- Transmission medium does not carry bits; it carries electrical or optical signals.
- So the bits received in a frame from the data-link layer are transformed and sent through the transmission media,
  - but we can think that the **logical unit** between two physical layers in two devices is a ***bit***.

- There are several protocols that transform a bit to a signal.

DATA-LINK LAYER IN THE TCP/IP (1)

- Several overlapping sets of links for datagram to travel from source to the destination.
  - The **routers** are responsible for choosing the ***best*** **links**.

- When the next link to travel is determined by the router,
  - the data-link layer takes datagram and moves across the link.

- The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN.

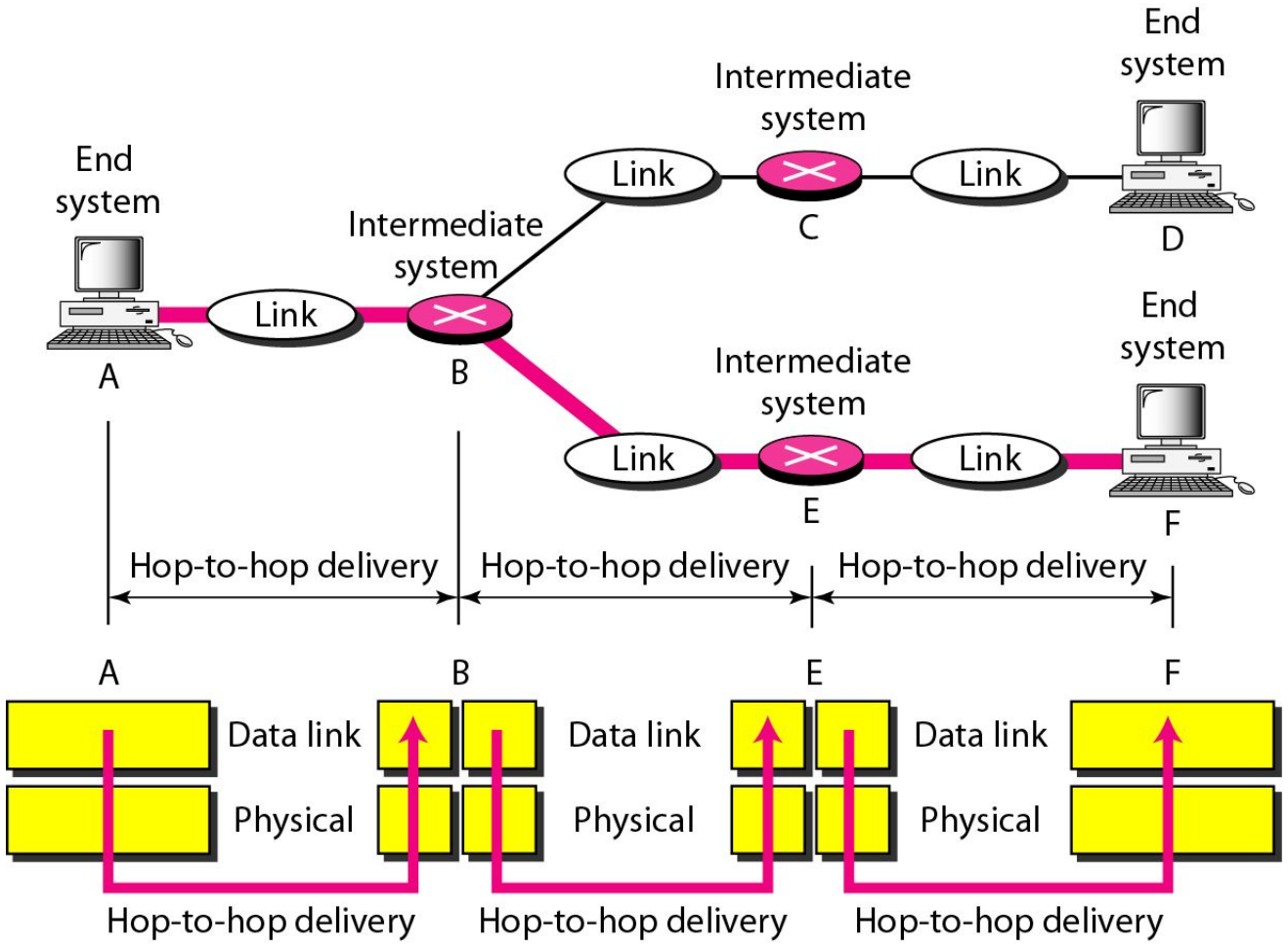
- In each case, the ***data-link layer is responsible for moving the packet (frame) through the link.***

DATA-LINK LAYER IN THE TCP/IP (2)

- **The data-link layer takes a datagram and encapsulates it in a packet called a *frame*.**

- Each link-layer protocol may provide a different service.
  - Some link-layer protocols provide complete error detection and correction,
  - some provide only error correction.

Hop-to-hop delivery



NETWORK LAYER IN THE TCP/IP (1)

- Responsible for creating a connection between the source and the destination.
- The communication at the network layer is **host-to-host.**

- Routers in the path are responsible for choosing the best route for each packet.

So, network layer is responsible for host-to-host communication and routing the packet through possible routes.

- In the Internet, it includes the main protocol, Internet Protocol (IP), that defines
  - the format of the packet, called a datagram at the network layer.
  - the format and the structure of addresses used in this layer.

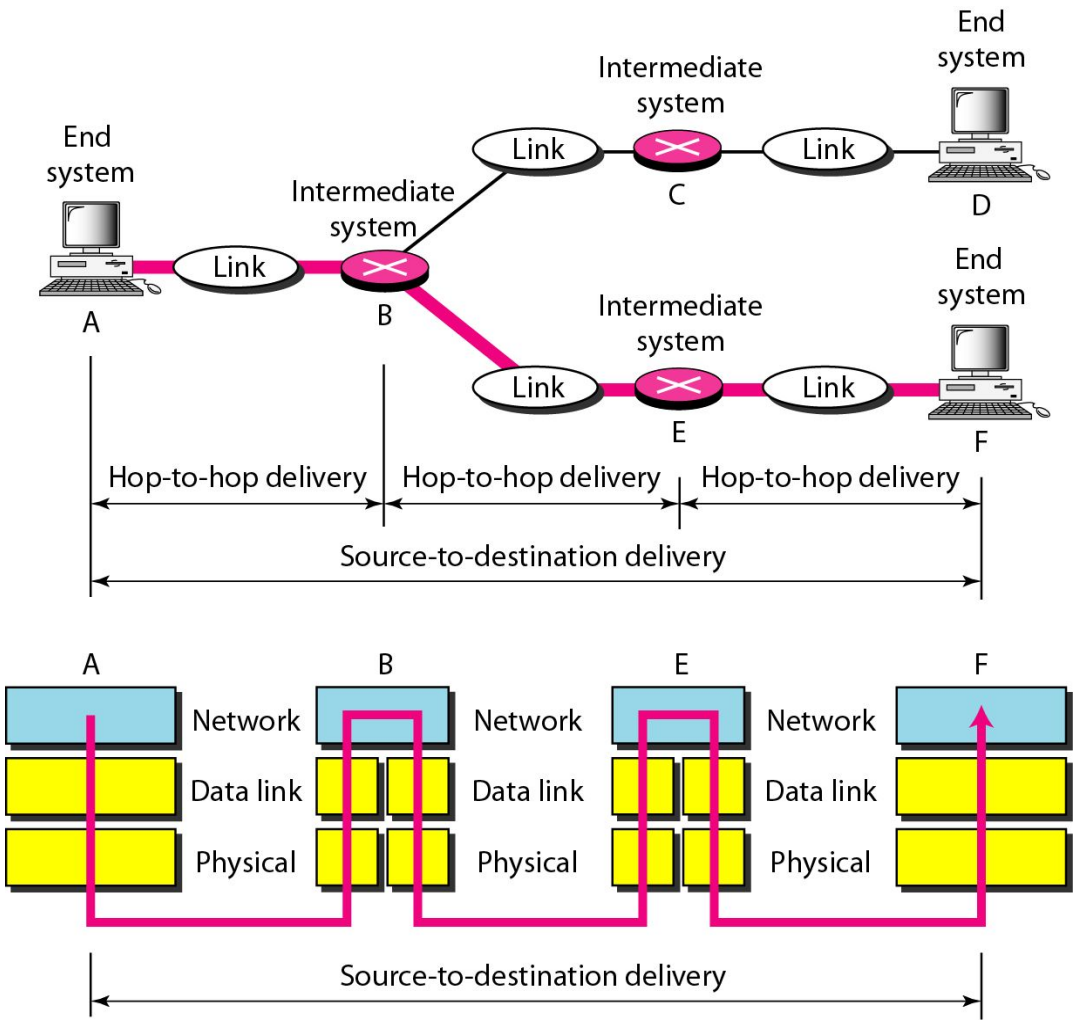
NETWORK LAYER IN THE TCP/IP (3)

- IP is also responsible for **routing a packet** from its source to its destination,
  - achieved by each router forwarding the datagram to the next router in its path.
- It includes unicast and multicast routing protocols

- Routing is duty of **IP** (not of routing protocol)

- Routing protocol helps routers in routing process
  - By creating forward tables for routers

Source-to-destination delivery



Transport Layer (Layer 4)

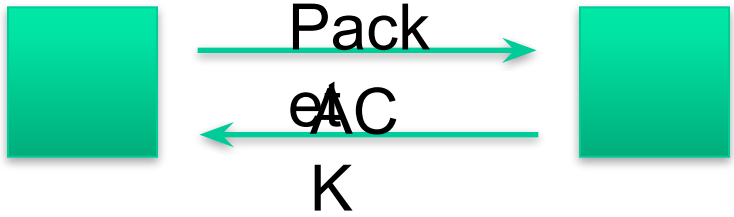
- **UDP** (User Datagram Protocol)

- Adds more addressing: "**ports**"
  - IP address tell you which computer
  - Ports tell you which application on that computer
  - Example: a web server "listens" to requests on port 80
  - Web browser: http://www.google.com:80 = http://216.58.216.100:80
    - ":80": optional

  - **Unreliable!**
    - Packets can get lost; packets can arrive out of order

Transport Layer

- **TCP** (Transmission Control Protocol)
- **Reliable** protocol!
- Adds ports (just like UDP), but also provides:
  - In-order delivery of packets (using sequence numbers)
  - Reliable delivery: using acknowledgment (ACK) packets

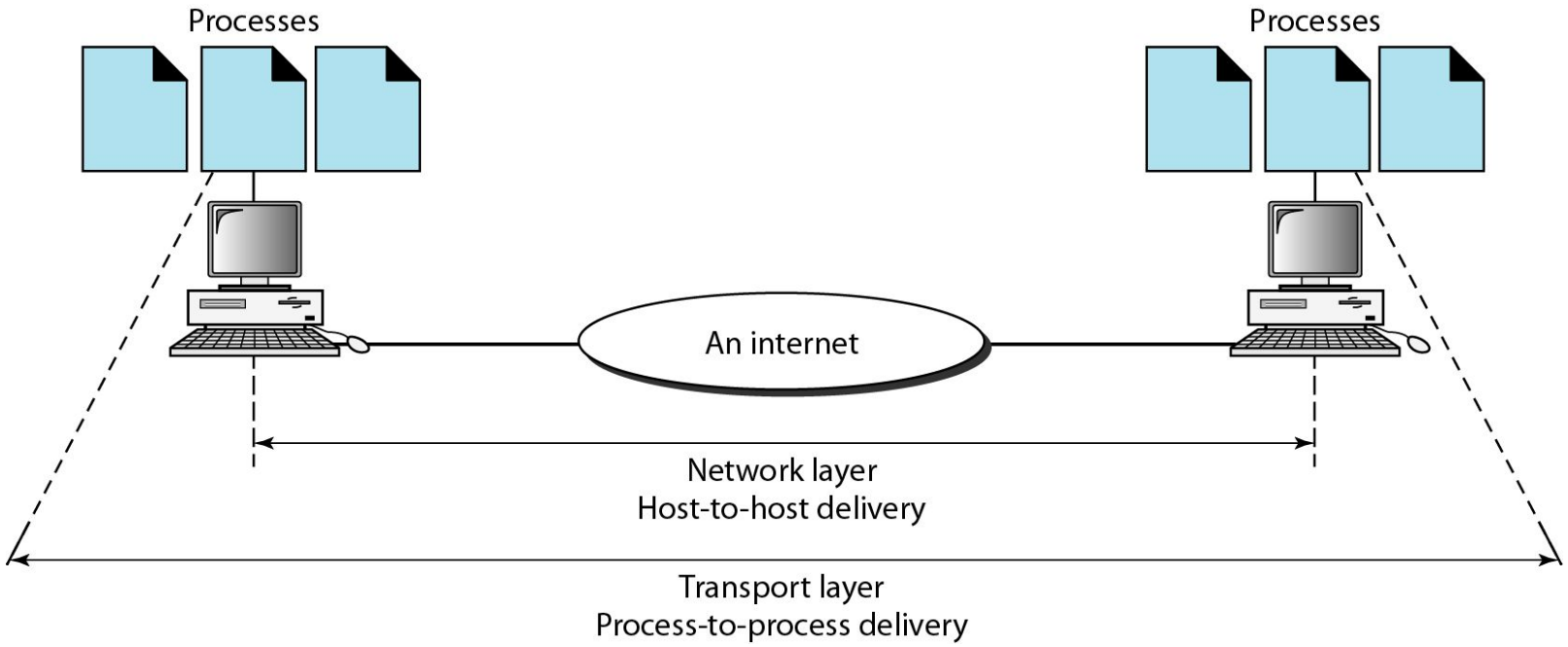


  - **Flow control & congestion control:**
    - Allows receiver to slow down sender
    - Allows "network" to slow down sender

UDP vs TCP

- TCP:
  - typical choice of most applications
  - do not want to lose data, out-of-order arrival, etc.
  - email, web traffic, financial transactions, etc.
- UDP:
  - can be "faster"
    - no flow/congestion control "slowing down" traffic
    - no retransmissions
    - good for "real-time" traffic
  - out-of-order arrival: can also "reorder" at application level
  - loss of data: can be acceptable
    - missing frames in video/audio stream

Reliable process-to-process delivery of a message



APPLICATION LAYER

The application layer is responsible to allow access to network resources (to send and receive data)

- Interface between user applications and lower network services

All applications and utilities communicating with the network fall in this layer

Examples: Browsers, Email clients (outlook express, Opera mail), FTP clients (WinSCP, Filezila)

The two application layers exchange *messages* between each other as though there were a bridge between the two layers.

- However, the communication is done through all the layers.

Protocols: Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), *Simple Network Management Protocol (SNMP), Domain Naming System (DNS),* Telnet etc.