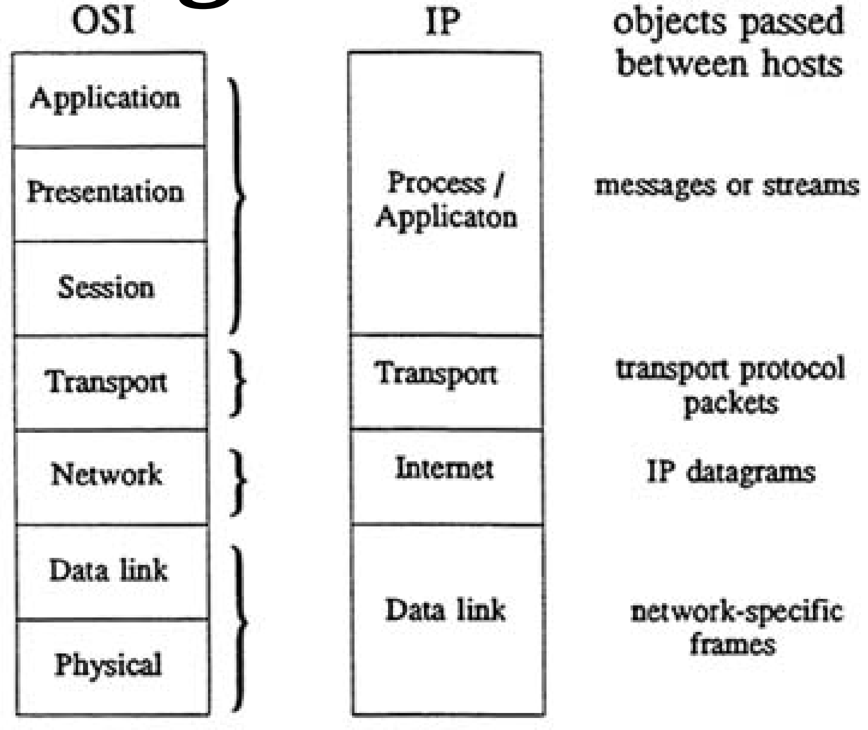
Protocol Layering and Internet



Contents

Protocol Layering

Advantages/Disadvantages

Reference Models:

- OSI
- ARPANET
- Internet

Encapsulation

Protocol Layering

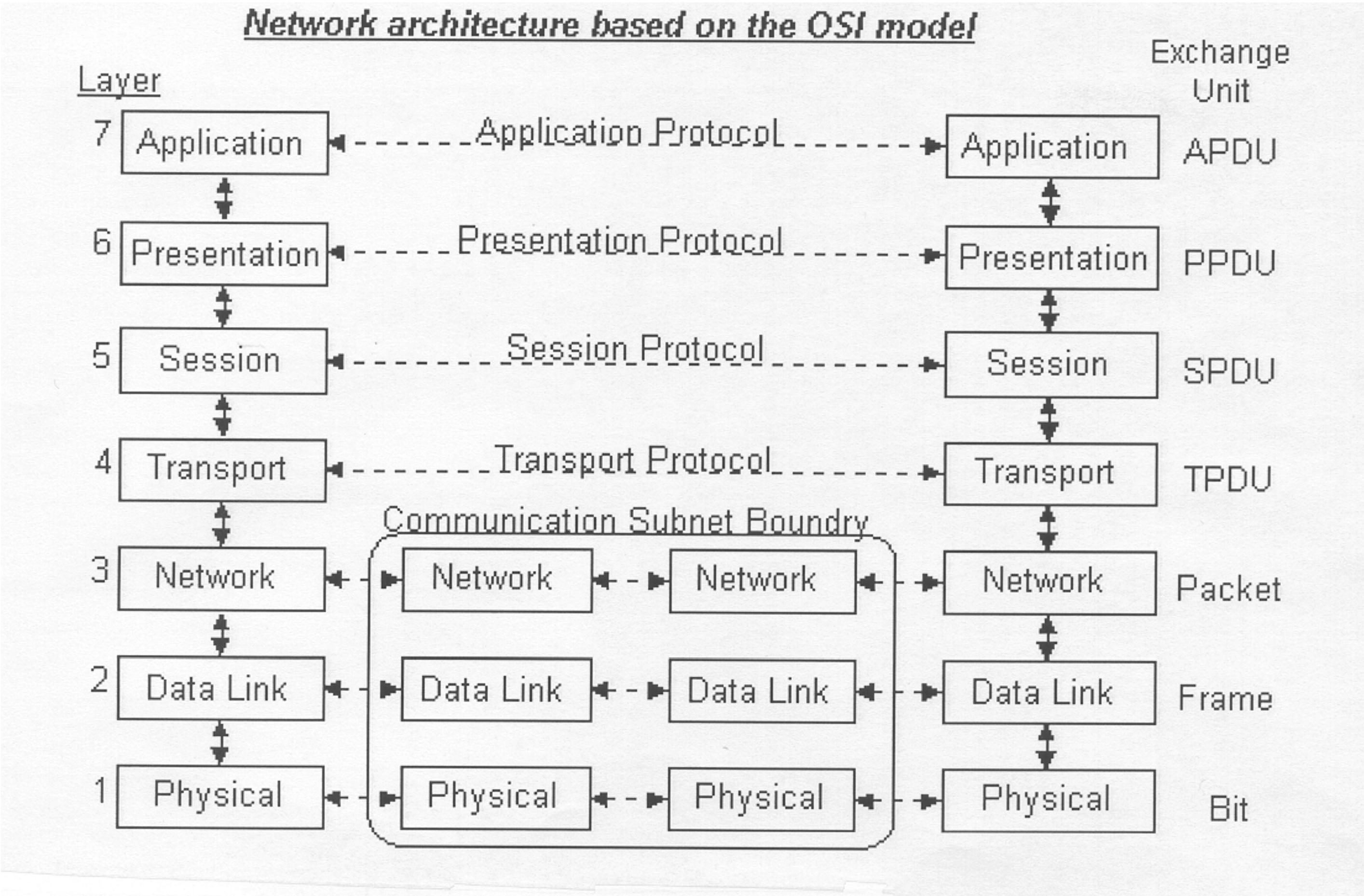
Advantages:

- Breaks down complex problem into more manageable components
- Implementation details of one layer are abstracted away from other layers; each layer has its own function

Disadvantages:

- Can introduce overhead, leading to intentional *layer violations*

The OSI Reference Model



The OSI Reference Model

Physical: transmits raw bits over a communication link

Data link: collects a stream of bits into a larger aggregate, frame

Network: routes packets among nodes

Transport: manages end-to-end delivery of information through error and flow control

Presentation: format of data exchanged between peers

Session: tie together potentially different transport streams

Ex. video and audio streams in a teleconferencing application

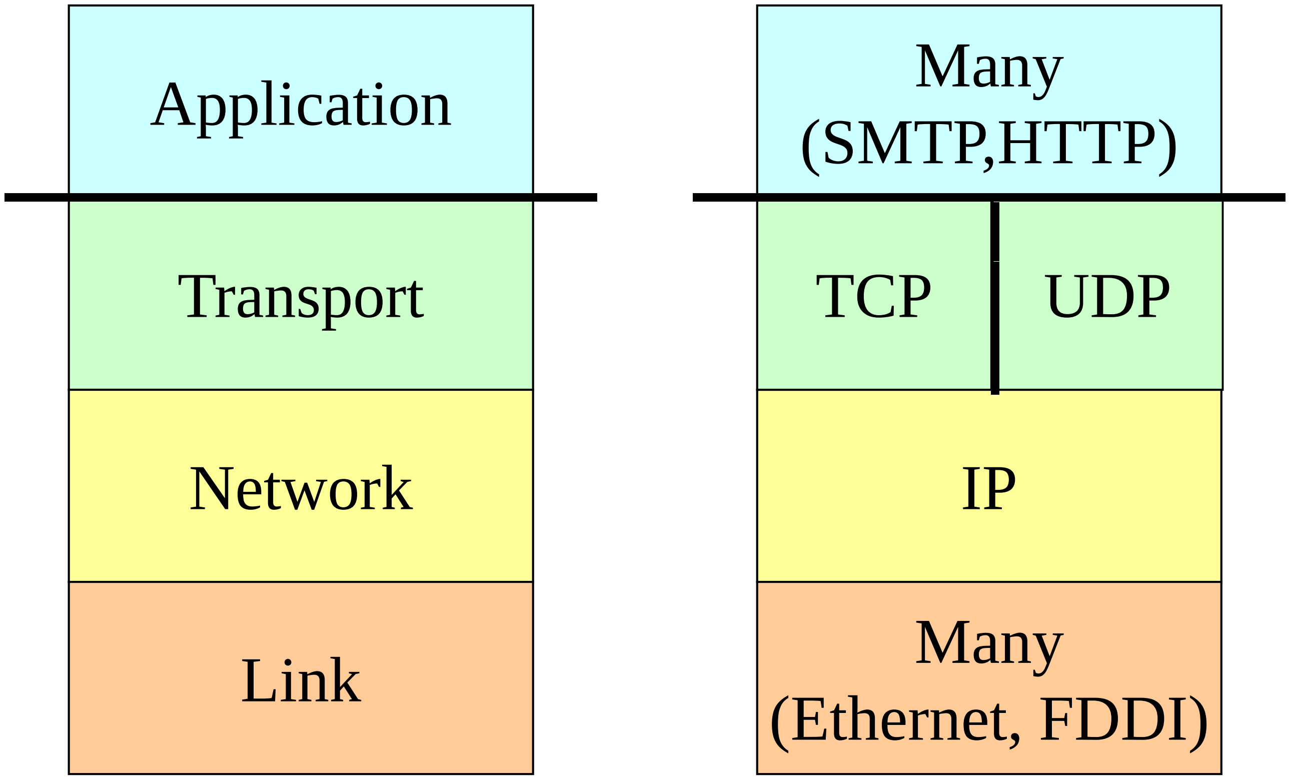
The ARPANet Reference Model

See RFC 871 by M. Padlipsky, *A Perspective on the ARPANET Reference Model* (1982)

3 Layer:

- network interface layer (link + physical)
- host-to-host layer (network)
- process/application (transport/application)

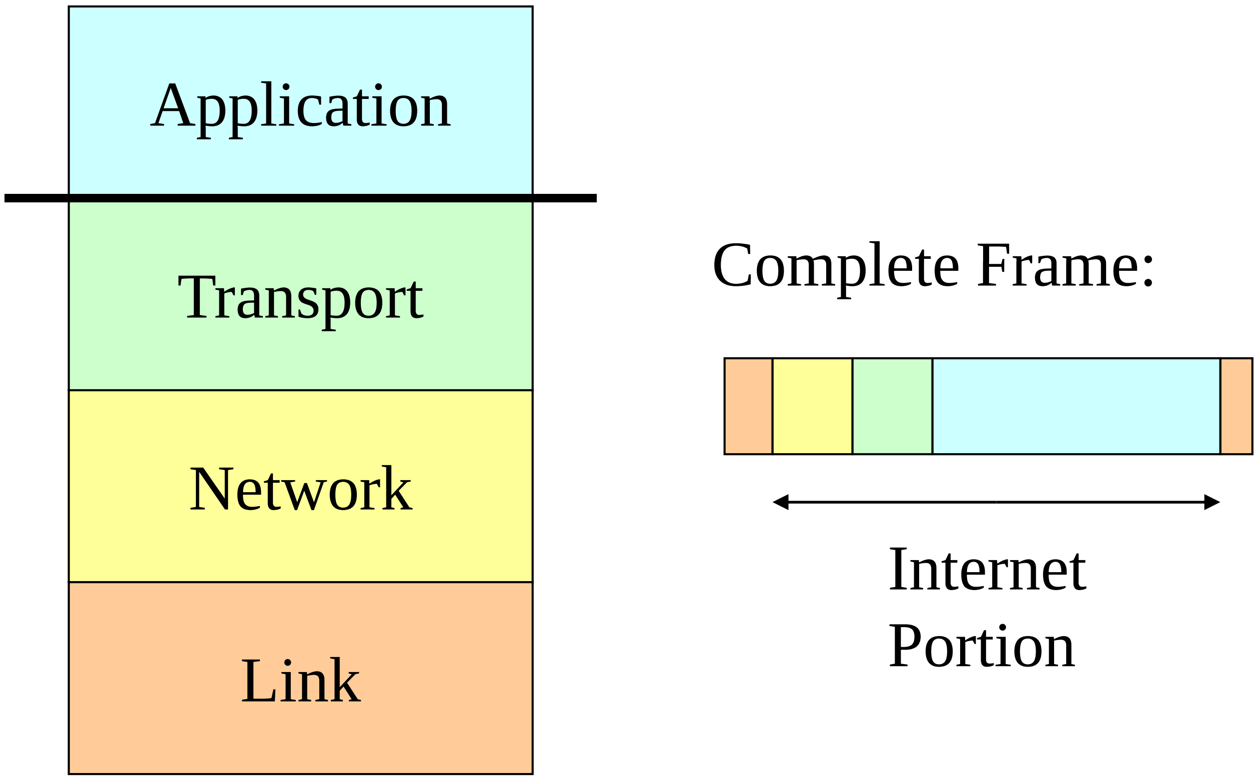
Internet Protocol Stack



Encapsulation

- Layer N messages being treated as opaque data to layer N-1
- Layer N-1 *multiplexes* among several layer N messages
- Each layer adds header (trailer)
- Receiver uses header as *demultiplexing key*

Encapsulation - Example



Issues in Network Design

Objectives

Placing Functionality

Internet Design Philosophy

Objectives of Network Design

Scope: support a wide range of approaches

Scalability: work well with very large network (encourages simplicity)

Robustness: operate (well) under partial failures

Incremental deployment: compatibility with existing system(s)

Placing Functionality in Network Design

Which functions belong at which layer?

(reliability, routing, encryption, compression, data conversion)

- the end-to-end argument
- application layer framing (ALF)

The End-to-End Argument

See [SRC84], "End-To-End Arguments in System Design"

The function in question can completely and correctly be implemented ***only with the knowledge of the application standing at the endpoints*** *of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)*

Placing Functionality: File Transfer…

Goal: to transfer a file correctly between peers

Method: break up file into messages, transfer messages

Threats: network may drop, reorder, duplicate, or corrupt messages

What if we have hop-by-hop reliability?

Where must correct delivery be checked?

Placing Functionality: Performance Impact

Consider reliability? Assume a link has probability p of losing a packet; (1-p) of not losing a packet

Traversing n hops gives $(1-p)^n$ prob of delivery and $1- (1-p)^n$ prob of drop

Assume typical Internet path of n = 15

Placing Functionality: Performance Impact

For a low loss rate ($p = 10^{-5}$),

$P_{loss} = 1- (1- 10^{-5})^{15} = 1.5x10^{-3} = .0015$ (< 1%)

But for a higher rate ($p = .01$, say, for wireless),

$P_{loss} = 1- (1-.01)^{15} = 0.14$!!

Internet was designed with < 1% path loss in mind; unfortunately, some parts today have much higher rates

Placing Functionality: Who Decides?

Each layer uses its own frame/packet/message format (size, layout) to provide its service

Application needs may not be communicated easily across layers

Idea: allow application to decide the frame format most convenient to it (ALF)

Internet Design Philosophy

Develop an effective technique for multiplexed utilization of existing interconnected networks

Other goals:

- Robustness in the face of failure
- Multiple types of communication services
- Compatibility with large variety of networks
- Distributed management, cost effective attachment, simple attachment, accountable

Internet Design Philosophy: Using Varieties of Networks

Make minimum assumptions on underlying networks

- Capable of transporting a message of reasonable size (say, 100 bytes minimum)
- Some form of addressing for non *point-to-point* or *multi-access* links

Major issues: addressing, packet sizes

Internet Design Philosophy: Connection Robustness

Endpoints need not re-establish communication during failures of intermediate devices

Protect *connection* state (where?)

Fate Sharing:

Place state only in endpoints

If connection is lost the communication is lost anyway

Internet Design Philosophy: Packet Switching

Packets: chunks of data

Consequences of fate sharing:

- Intermediate nodes must not have any essential connection state
- Desire to use packet switching with **datagrams**
- More trust is placed in end hosts
- Less trust in intermediate devices

Today's Internet

- A network of networks, comprising about 100,000 networks
- All hosts/routers run the IP protocol (today, IP version 4):
  - Datagram interface, best-effort host-to-host delivery
  - Routing based on global addressing
  - Common datagram format (IP packet)

Best Effort Delivery

Lost packets (usually due to congestion)

Duplicated packets (retransmission)

Damaged packets (channel noise)

Re-ordered packets (routing changes)

Internet Design Futures

- Desire to differentiate some traffic and treat it specially (QoS)
- Using “Soft State” (state info for each flow to make resource allocation decision) in routers/switches:
  - Does not need to be explicitly deleted when it is no longer needed
  - Provides for enhanced services
  - Times out if not refreshed by end-points
  - Issues: traffic overhead, time-out values

Summary

Protocol layering breaks down complex problem into more manageable components,

but introduces overhead

Internet Protocol Stack

Application/Transport/ Network/Link

HTTP, TCP/UDP, IP, Ethernet/FDDI

All hosts/routers run the IPv4

Best-effort host-to-host datagram delivery