Physical Layer

- Transmitting raw bits over a “wire”
- – Make sure a “1” bit is sent as a 1
- EE/ECE problem:
- – How many volts represents a “1” or “0”?
- – How long does a bit time last?
- – How many pins does the connector have?
- – How many wires does the transmission media

Data Link Layer

- Communication between two machines
- Transforms raw transmission of physical layer into error-free channel
- Divides physical layer physical layer into *frames*
- – messages containing data and control information
- Handles lost, damaged, and duplicate frames
- Handles slowing down a fast transmitter

– *flow-control*

Network Layer

- Controls operation of the *subnet*
- – communication between hosts
- Routes *packets* from source to destination
- – not guaranteed delivery
- Handles *congestion*
- – too many packets in network
- Handles *addressing*

– Which machine?

Transport Layer

- Makes sure data gets delivered to a specific process on a specific machine
- *End-to-end* protocol

– sender and receiver

- Handles retransmissions, if needed
- Handles duplicates, if needed
- Also deals with addressing

– Which process on a particular machine?

– The *port* specification in a `socket`

Session Layer

- Long-term connections between processes
- Clean interface to the transport layer
- – Not OS specific (sockets in BSD Unix, or TLI in System V streams)
- Provides synchronization

– recovering from transport layer failure

– *token* for floor control

Presentation Layer

- Apply semantics to data

– example: name, address …

- Format in agreed upon way
- General services:

– Format data (ASCII to Unicode)

– Compressing data

– Encryption

Application Layer

- The user programs themselves

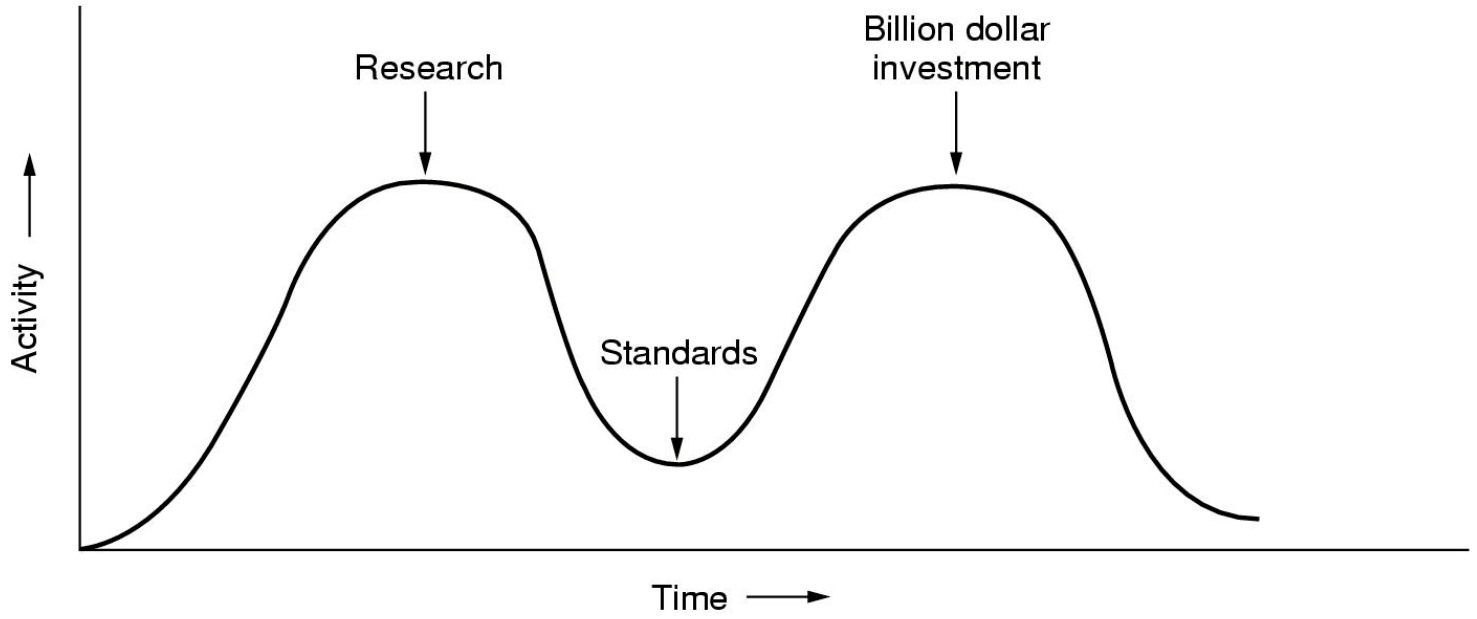
– ftp

– telnet

– X

– talk

Critique of OSI



- Plus, bad technology (big specification)
- Plus, bad politics (pushed by govt. orgs)

ARPANET

- Predecessor to the Internet
- Phone lines first, satellite and radio later

– req: connect multiple networks seamlessly

- DoD worry about routers going down

– req: survive loss of subnet hardware without

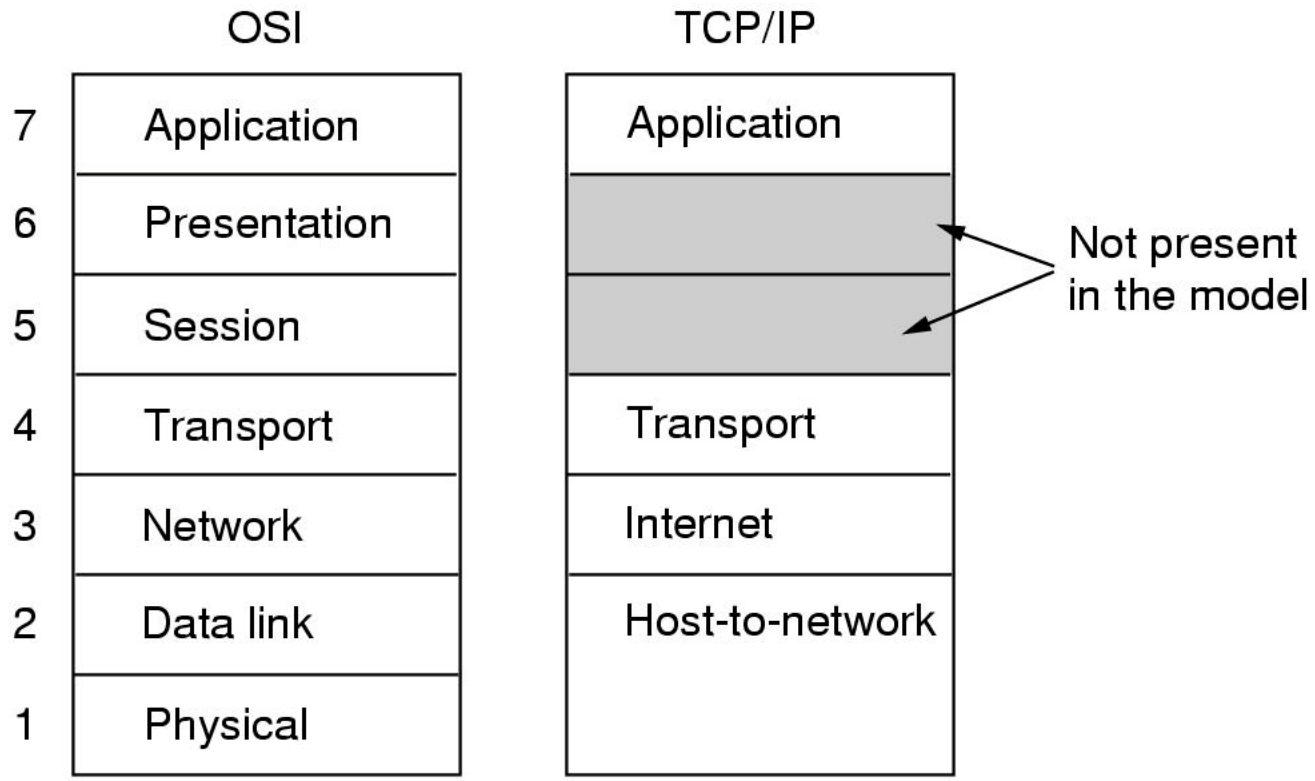
- losing connections
- Applications with diverse requirements

– req: flexible architecture

- Used TCP/IP protocols

– **then** came their reference model

TCP/IP Reference Model



Internet Layer

- Packet switched
- Connectionless
- Packets can be:

– travel different routes

– lost

– out of order

- Called *IP (Internet Protocol)*

Transport Layer

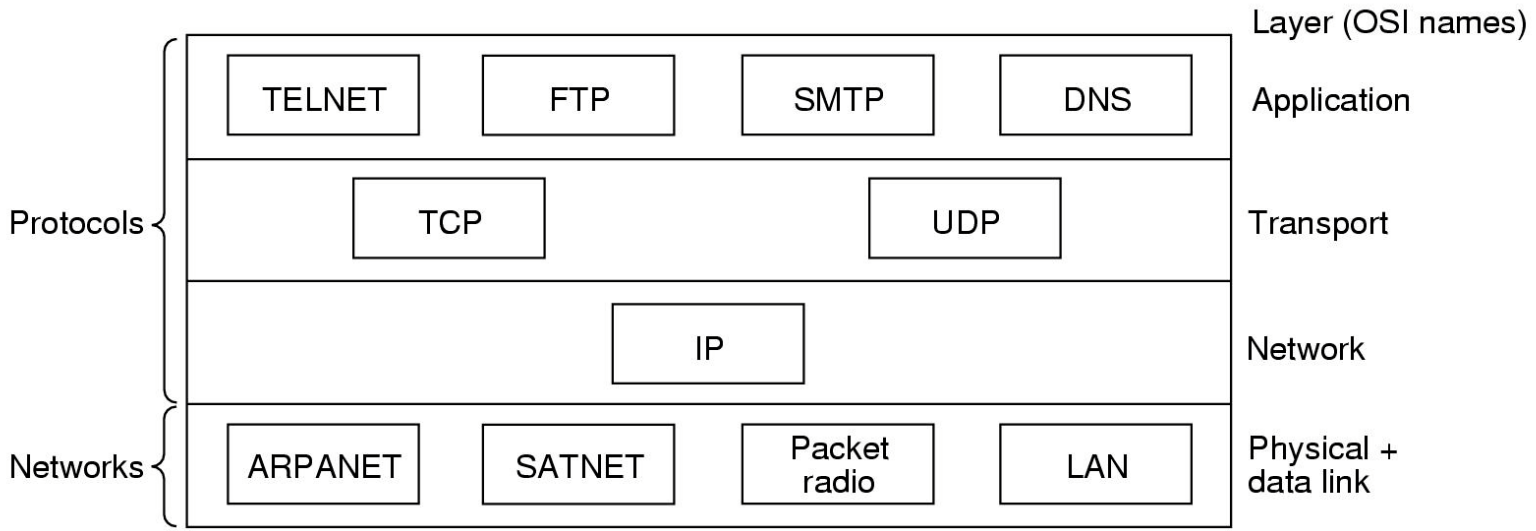
- Similar to OSI Transport Layer

– end-to-end, “conversation”

- Two protocols

– TCP: reliable, stream, flow control, connection

– UDP: unreliable, no flow control, connectionless



Application Layer

- No session/presentation layers -- no need
- High-level protocols:

– original: telnet, ftp, smtp, dns

– new: http, nntp

Host-to-Network Layer

- Great void
- Not specified, not talked about in research literature

Critique of TCP/IP Model

- Not clean in describing service, interface
- and protocol

– not a good guide for new technologies

- Not general, tied to protocols

– hard to describe other networks

- No physical and data link layers

– hard to abstract from physical hardware

– “re-invent the wheel”

- IP, TCP well-thought out, but others not

– TELNET: 10 cps, no GUI, no mouse

Model Differences: OSI and TCP/IP

- OSI concepts:

– services: what layer does

– interface: how processes above access it

– protocols: how it works, private to layer

– great for OO!

- Not so clean in TCP/IP

– harder to replace as technology changes

Differences: OSI and TCP/IP

- OSI model before protocols

– implementations hacked (ex - broadcast instead

- of point-to-point needed new layer)
- TCP/IP protocols before model

– model does not fit other protocols

– not useful for non TCP/IP networks

- OSI transport

– connection oriented only

- TCP/IP transport

– connection + connectionless

Hybrid Model

- OSI useful for discussing networks
- TCP/IP provides better protocols for using them

ATM Overview

- Telephone companies coordinate multiple networks
- – ex: POTS circuit-switched, other packet switched
- Invent network of future to manage all

– Broadband-ISDN

- B-ISDN made possible by *Asynchronous*
- *Transfer Mode (AT*M)

ATM Basics

- Transmit data if fixed sized *cells*
  - Flexible (audio, video, text)
  - Fast (155 Mbps and 622 Mbps)
  - But , huge break from circuit switching
- Connection oriented
- Niche, for now, is connecting LAN's