

EECS 122, Lecture 2

Kevin Fall

kfall@cs.berkeley.edu

Protocol Layering

- Advantages
 - breaks down complex problem into more manageable components
 - implementation details of one layer are abstracted away from other layers
- Disadvantages:
 - can introduce overhead, leading to intentional *layer violations*

The OSI Reference Model

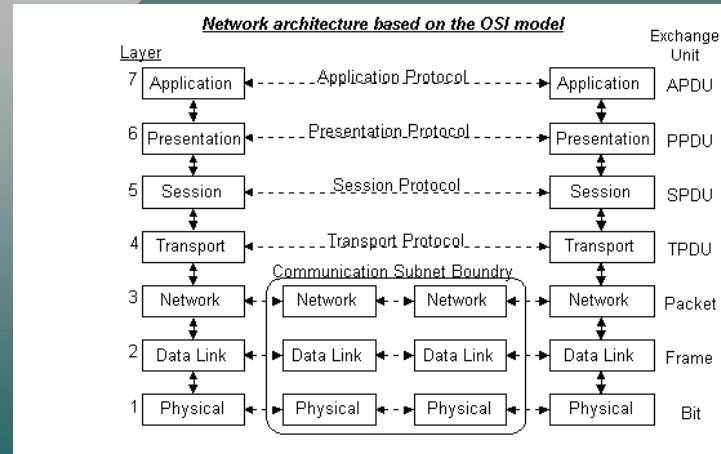
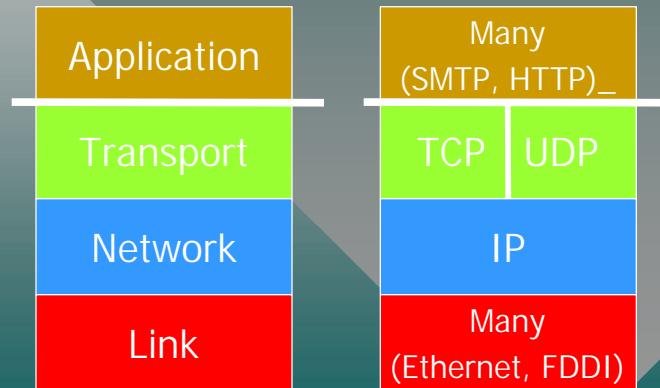


Illustration from Tel Aviv Univ., Israel, ramif@zot.tau.ac.il

The ARPANet Ref. Model

- See RFC 871 by M. Padlipsky, *A Perspective on the ARPANET Reference Model* (1982)
- 3 Layers:
 - network interface layer (link + physical)
 - host-to-host layer (network)
 - process-level/application (transport/app)

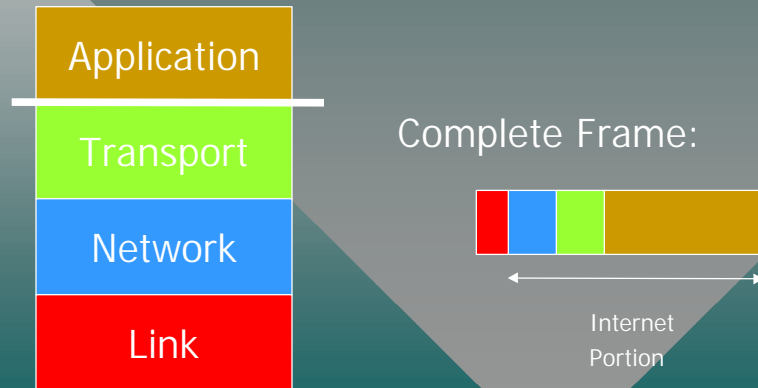
Internet Protocol Stack



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

- **Scope:** support a wide range of approaches
- **Scalability:** work well with very large networks (encourages simplicity)
- **Robustness:** operate (well) under partial failures
- **Incremental deployment:** compatibility with existing system(s)

Placing Functionality

- Which functions belong at which layers?
(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.)

Consider a 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?

Consider Encryption ...

- Which layer should encrypt data?
- Higher: data is in the clear in fewer places, keys are nearest the user, every application must encrypt
- Lower: more opportunity to intercept, how to provide key material, applications are simpler (don't worry about crypto)
- User vs Administrator locus of control

Performance Impact

- Consider reliability... assume a link has probability p of losing a packet; $(1-p)$ of not losing a packet
- Traversing n hops give $(1-p)^n$ prob of delivery and $1 - (1-p)^n$ prob of drop
- Assume “typical” Internet path of $n = 15$

Performance Impact (contd)

- For a low loss rate ($p = 10^{-5}$),
Prob(loss) = 1.5×10^{-5}
- But for a higher rate ($p = 10^{-2}$, say, for wireless), Prob(loss) = 0.14 !!
- Internet was designed with $< 1\%$ path loss in mind; unfortunately, some parts today have much higher rates (later)

Who Decides?

- Each layer uses its own frame format to provide its service
- Application needs may not be communicated easily across layers
- Idea: allow application to decide the frame size most convenient to it... ALF

Application Layer Framing (ALF) [CT90]

- Application breaks up data into suitable aggregates
- These Application Data Units (ADUs) may be processed by the application in any order
- Lower layers preserve ADU boundaries

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 nets
 - distributed management, cost effective attachment, simple attachment, accountable

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* links
- Major issues: addressing, packet sizes

Connection Robustness

- endpoints need not re-establish communication during intermediate failures
- thus, must protect *connection state*
- *Fate Sharing*: place state only in endpoints. If they are lost the communication is lost anyway.

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

Internet Design Futures

- Desire to differentiate some traffic and treat it specially (CoS or QoS)
- “Soft State” in routers/switches:
 - not critical in enabling communications
 - provides for enhanced services
 - times out if not refreshed by end-points
 - Issues: traffic overhead, time-out values

Today's Internet

- a “network of networks,” comprising about 50,000 networks
- all hosts/routers run the IP protocol:
 - datagram interface, “best-effort” host-to-host delivery
 - routing based on global addressing
 - common datagram format

Best Effort Delivery

- lost packets (usually due to congestion)
- duplicated packets (retransmission)
- damaged packets (channel noise)
- re-ordered packets (routing changes)

Measuring Performance (Capacity and Utilization)

- Capacity
 - the rate (bits/second) of a communication channel
 - typically fixed by oscillator rate, noise, coding
- Utilization
 - the fraction of capacity in actual use measured over some interval of time

Measuring Performance (Throughput and Delay)

- Throughput
 - the data rate (bits/second) available to a particular application over an interval of time
- Delay/Latency (one-way)
 - the time required to send a minimum-sized data unit from sender to receiver

Measuring Performance (Derived Values)

- Jitter: variability in delay
- Round-Trip Time (RTT)
 - two-way delay from sender to receiver and back
- Bandwidth-Delay Product
 - product of bw and delay, indicates “storage” capacity of network

A Note on Units

- “Mega” is sometimes 1,000,000, and sometimes 1,048,576 (similar for kilo-units)
- 1 MB is 8 times larger than 1 Mb
- data rates are often powers of 10 (100 Mb/s Ethernet), whereas messages are often powers of 2 (a 1KB message is 1024 bytes)