Internetworking

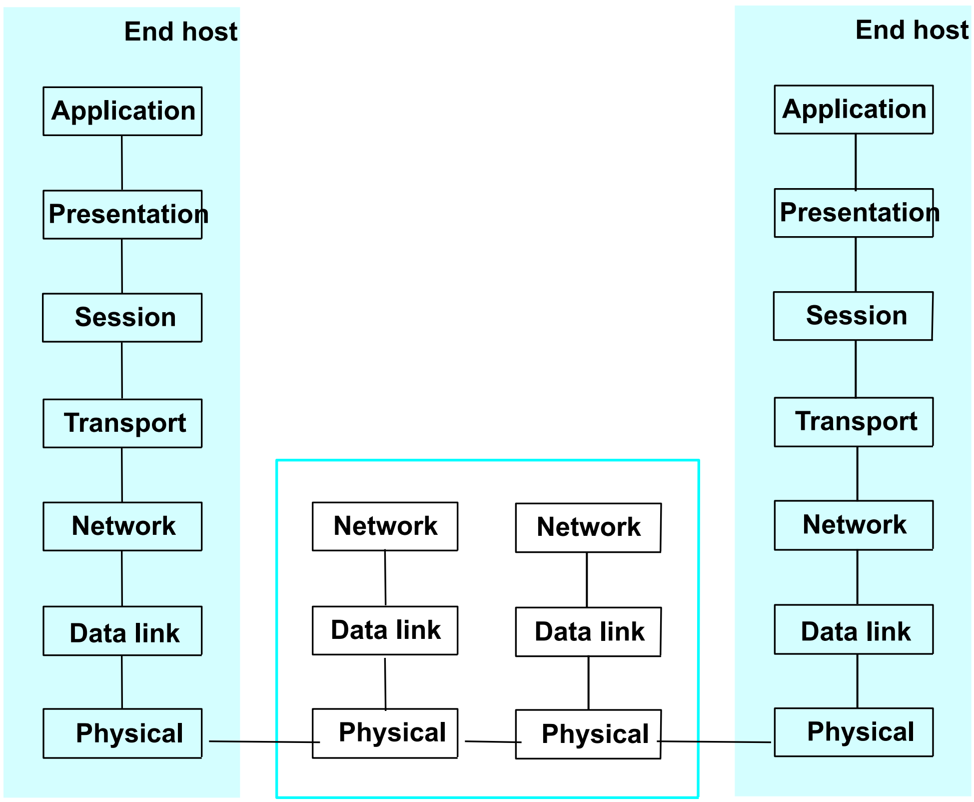
Outline

- Internet Architecture
- Best Effort Service Model

Layering revisited (cause it's real important)

- Recall pros and cons of packet switched networks
  - Pros: High utilization, low startup overhead
  - Cons: No guarantees on delay and loss
- But files are transmitted without "holes"?
- Reliable transfer is a function of a *specific protocol layer* (TCP)
- Distinct functions are separated into layers
  - lower layers as black boxes (like C library)
- Layering simplifies description of functions and enables interoperability in heterogeneous environment

ISO Architecture



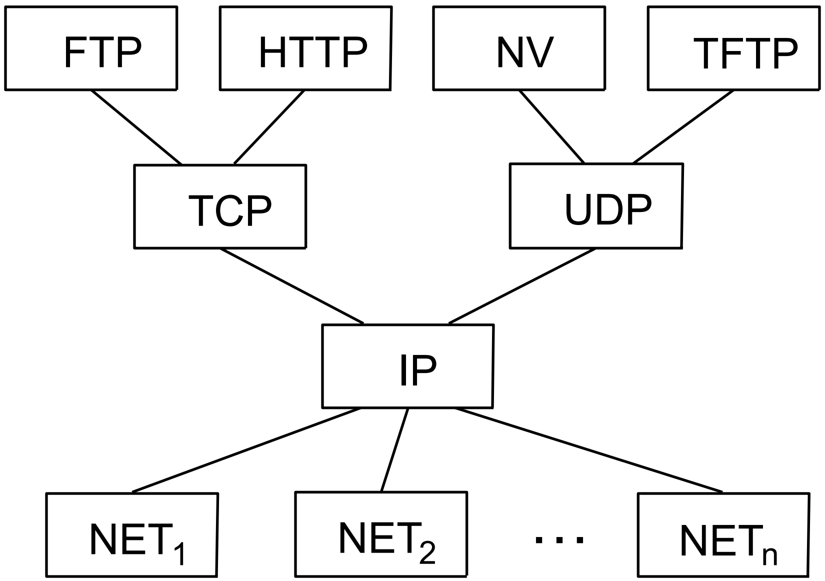
One or more nodes
within the network

Internet Architecture

- Defined by Internet Engineering Task Force (IETF)
  1. Application: interacts with user to initiate data transfers (browser, media player, command line)
  2. Transport: reliable, in-order delivery of data (TCP and UDP)
  3. Network: addressing and routing (IP)
  4. Data Link: defines how hosts access physical media (Ethernet)
  5. Physical: defines how bits are represented on wire (Manchester)
- Information is passed between layers via encapsulation
  - Header information is attached to data passed down layers
- Multiplexing between layers
- Layers access other layers via API's (eg. sockets)
- Communication at a specific layer is enabled by a protocol

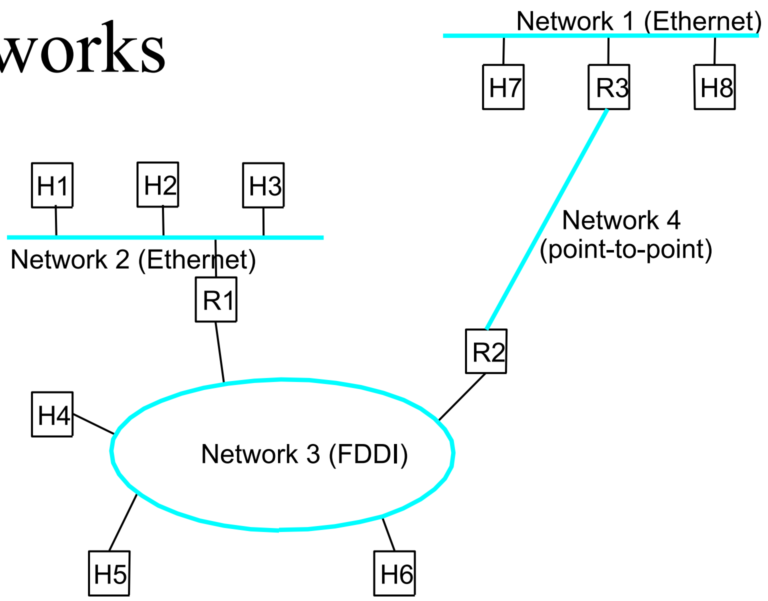
Hourglass Design

- Single protocol at network level insures packets will get from source to destination while allowing for flexibility

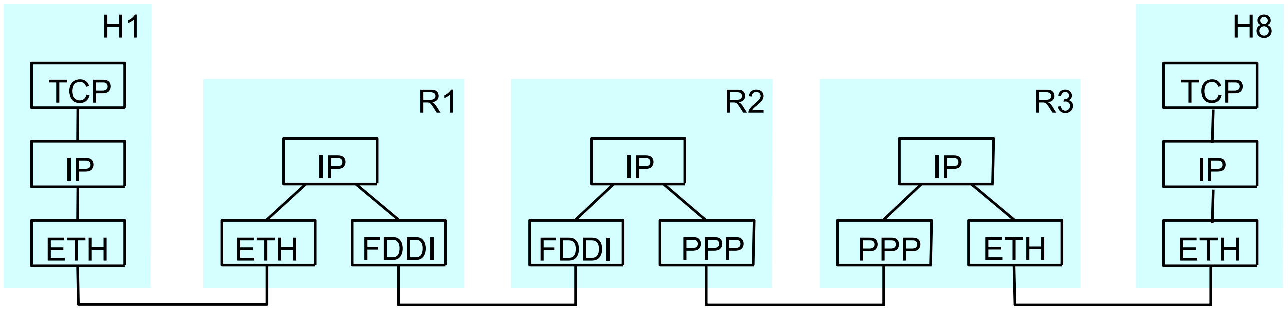


IP Internet

- Concatenation of Networks

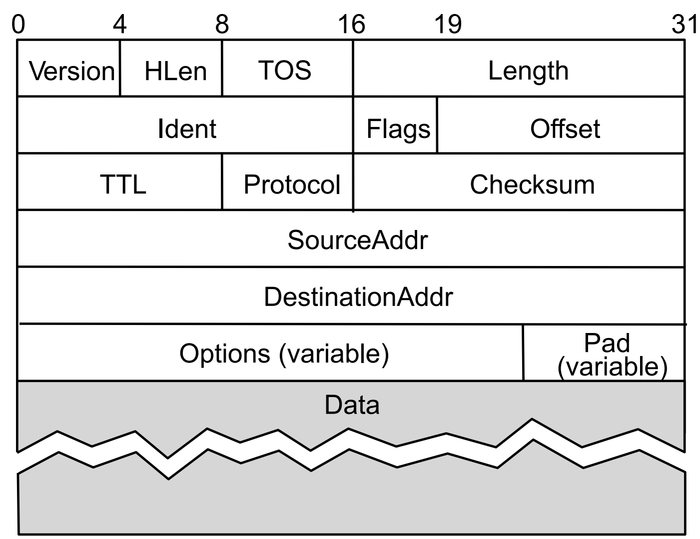


- Protocol Stack



IP Service Model

- Connectionless (datagram/packet-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- Datagram format



Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table
- Example

Network Number	Next Hop
1	R3
2	R1
3	interface 1
4	interface 0

Forwarding Tables

- Suppose there are *n* possible destinations, how many bits are needed to represent addresses in a routing table?
  - $\log_2 n$
- So, we need to store and search $n * \log_2 n$ bits in routing tables?
  - We're smarter than that!