Internet History and Architectural Principles

Lecture 1

Advanced Computer Networks

Roadmap

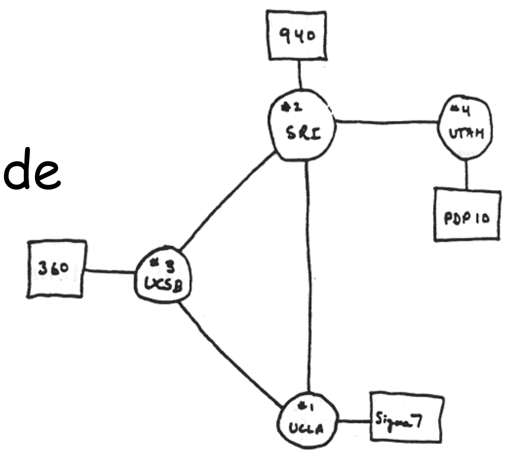
- A brief history of the Internet
- Circuits vs. packets
- Design philosophy of Internet protocols
- Placement of functionality
  - End-to-end principle
  - Layering

A brief history of the Internet

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK

A brief history of the Internet

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

A brief history of the Internet

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control

- New national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

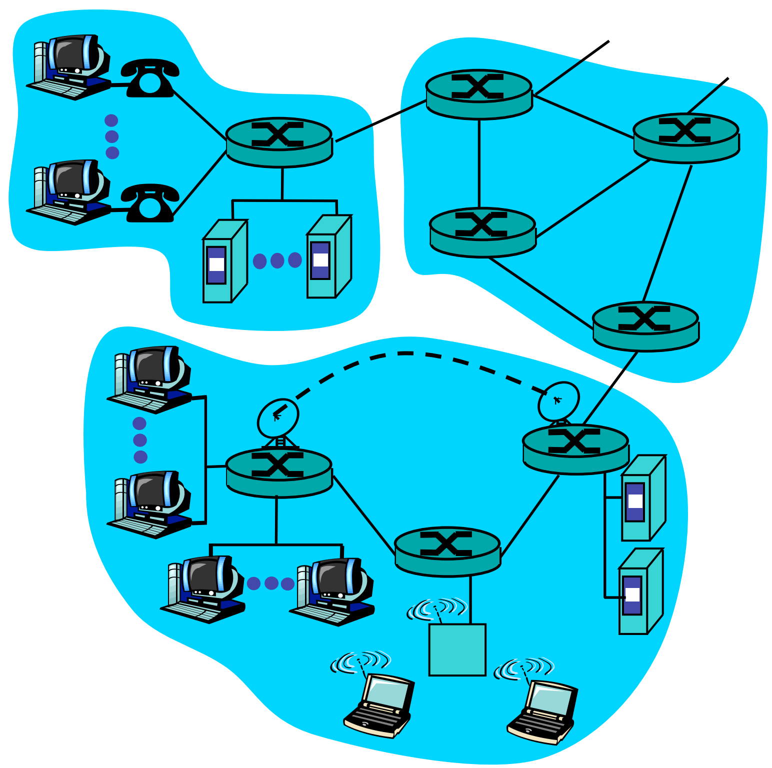
A brief history of the Internet

1990, 2000's: commercialization, the Web, new apps

- Early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- Early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web
- Late 1990's – 2000's:
  - more killer apps: instant messaging, P2P file sharing
  - network security to forefront
  - est. 50 million host, 100 million+ users
  - backbone links running at Gbps

A closer look at network structure:

- network edge: applications and hosts
- network core:
  - networked routers
  - network of networks
- access networks, physical media
  - communication links

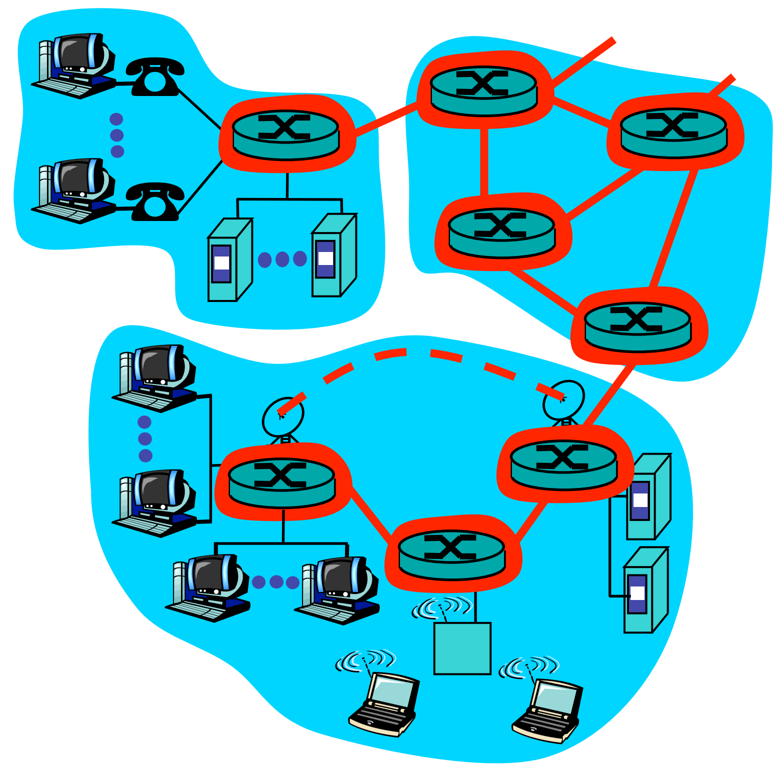


Fundamental architectural questions

- Q1: How should the network allocate resources to users?
- Q2: How should functionality be divided between the edge and the core?

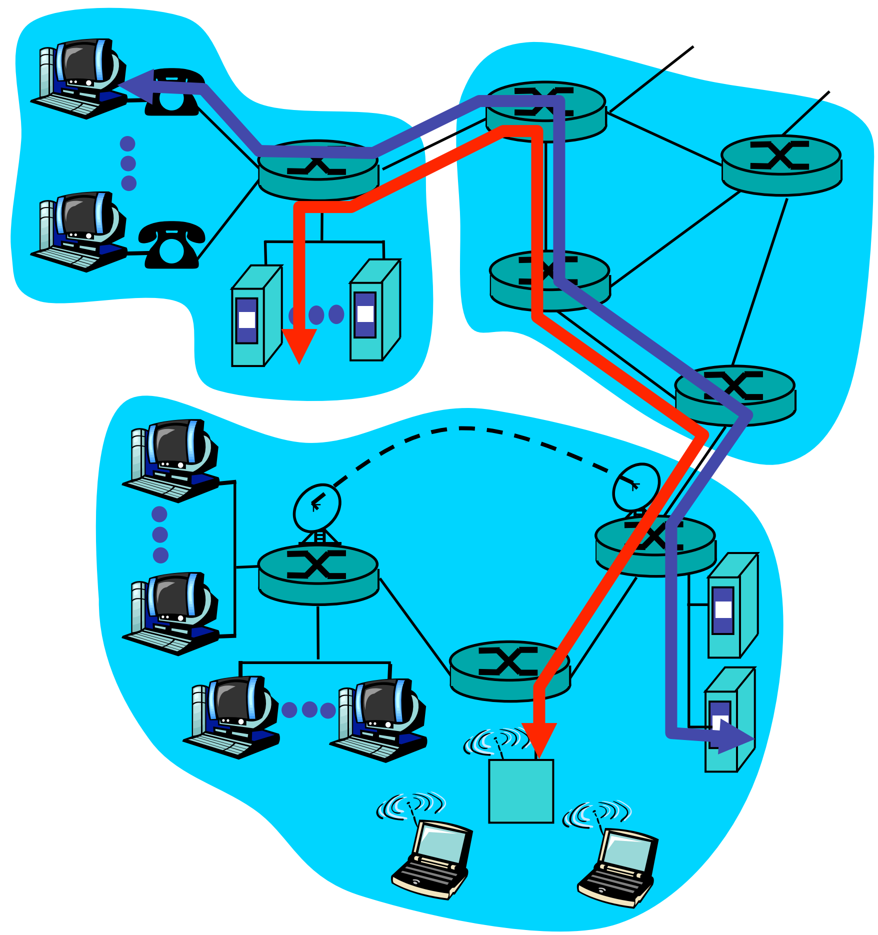
How should network allocate resources

- *The* fundamental question, with two possible answers
  - circuit switching: dedicated circuit per call: telephone network
  - packet-switching: data sent through network in discrete "chunks"



Circuit Switching

- End-to-end resources (eg link bandwidth, switch capacity) reserved for "call"
- Dedicated allocation, ie, no sharing
- Guaranteed performance
- Call setup required



Circuit Switching

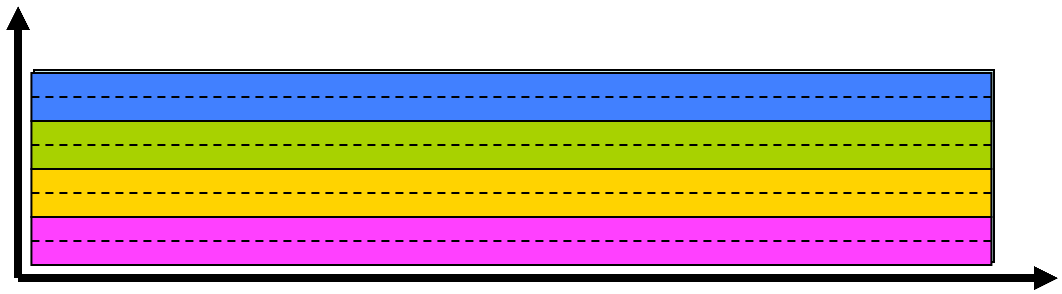
- Dividing link bandwidth across calls
  - frequency division
  - time division

FDM and TDM

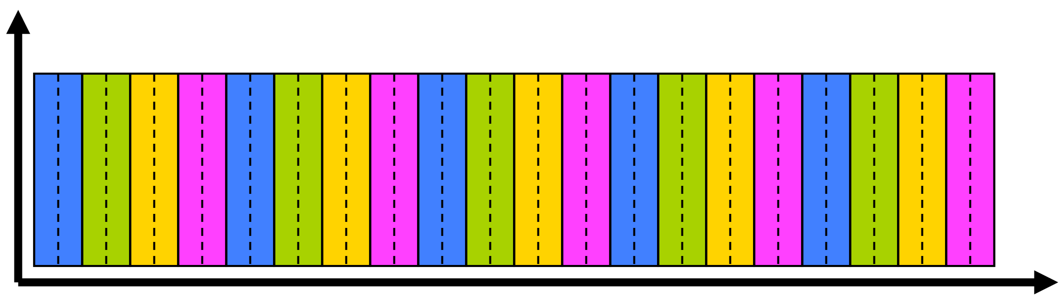
Example:

4 users

FDM



TDM



Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

Packet Switching

Each end-end data stream divided into *packets*

- Packets across flows *share* network resources
- Each packet uses full link bandwidth
- Resources used *as needed*

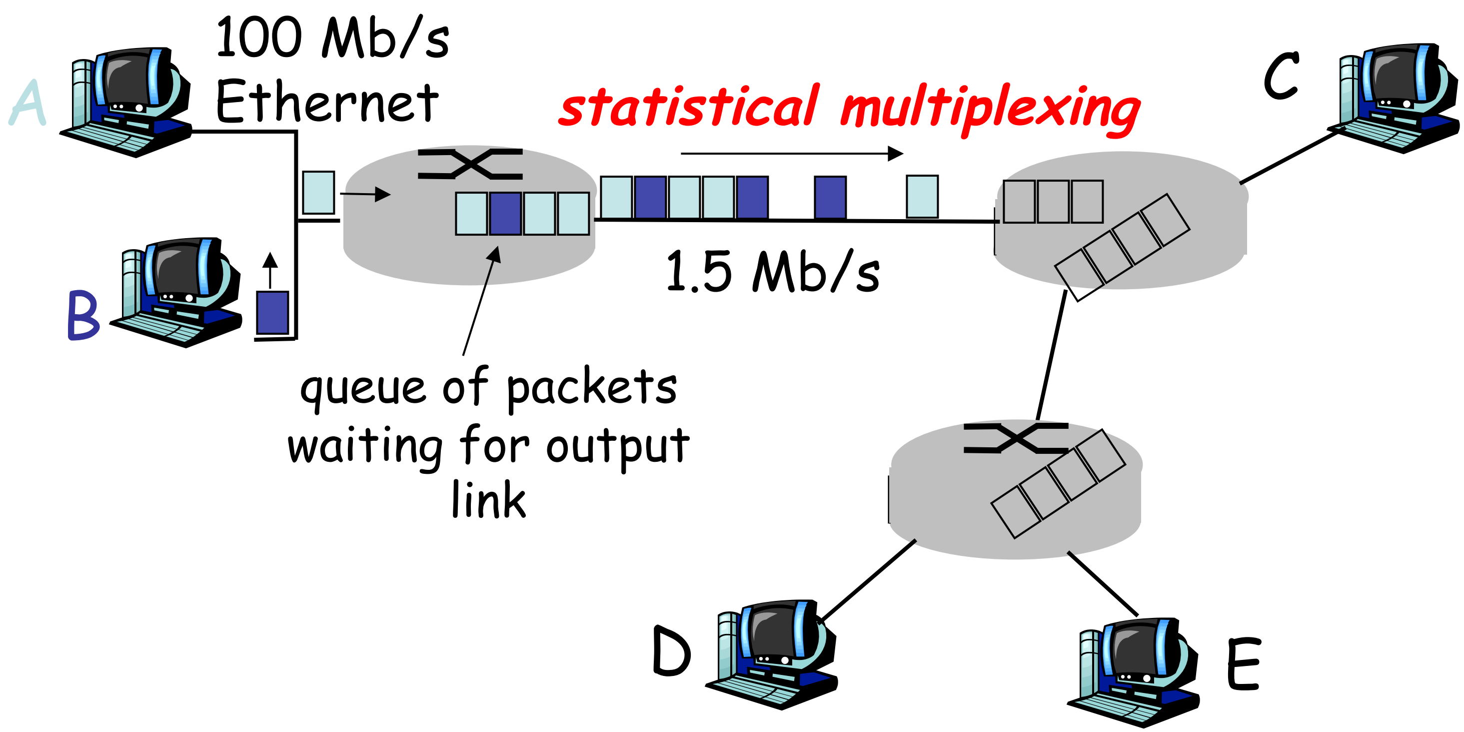
~~Dedicated allocation~~

~~Resource reservation~~

Resource contention:

- Aggregate resource demand can exceed amount available
- Congestion: packets queue, wait for link use
- Store and forward
  - packets move one hop at a time
  - node receives complete packet before forwarding

Packet Switching: Statistical Multiplexing

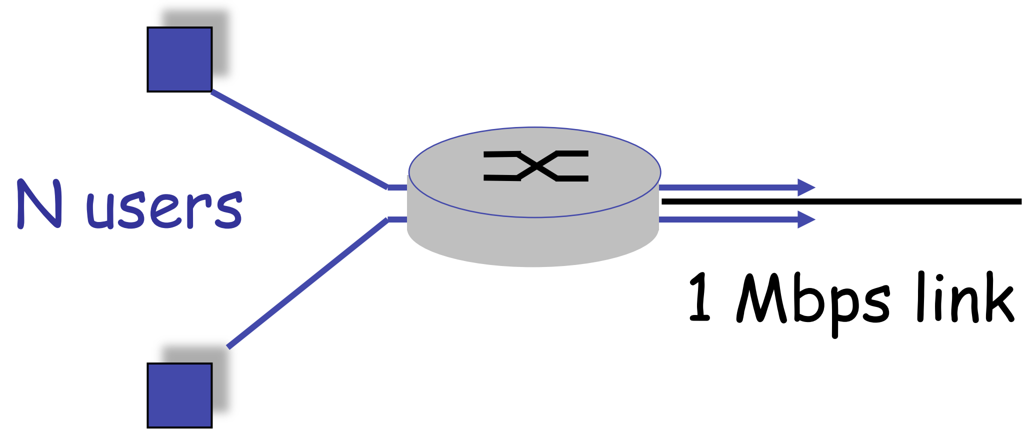


Sequence of A & B packets does not have fixed pattern, shared on demand ➡ ***statistical multiplexing***.

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- Each user:
  - 100 kb/s when "active"
  - active 10% of time
- Circuit-switching:
  - 10 users
- Packet switching:
  - with 35 users, probability > 10 active less than .0004



Q: how did we get value 0.0004?

Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion: packet delay and loss
  - protocols needed for reliable data transfer
  - congestion control needed
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still a research question

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Fundamental architectural questions

- Q1: How should the network allocate resources to users?
- Q2: How should functionality be divided between the edge and the core?

Edge vs. core functionality

- Telephone networks: “dumb” end-systems, complex network
- Internet: “smart” end-systems, simple network

Disclaimer: “Dumb”, “smart”, “simple”, “complex” are in the eyes of the beholder and difficult to quantify, eg read [MMZ02]

Roadmap

- A brief history of the Internet
- Circuits vs. packets
- Design philosophy of Internet protocols
- Placement of functionality
  - End-to-end principle
  - Layering

Design philosophy of the DARPA Internet [Clark88]

- Top level goal: “to develop an effective technique for multiplexed utilization of existing interconnected networks”

Umm.. so what exactly does this mean?

Seven second level goals

1. Internet communication must continue despite loss of networks or gateways.

- Assumptions/implications at end-host
  - Thou shalt communicate unless partitioned (no connection setup, route diversity exists, adaptive network-layer, end-host oblivious to network information)
  - Thou shalt not depend on a router (stateless)
  - Thou shalt not fail unless thou fails (fate sharing)

Seven second level goals

2. Internet must support multiple types of communications service

- Layering implications
  - TCP maybe inappropriate (eg, ping) or overkill (eg, real-time audio/video), so TCP must be in a separate layer from IP
  - TCP and UDP and many other protocols co-exist on top of IP

Seven second level goals

3. Internet must accommodate a variety of networks

- Implications
  - Thou shalt not expect anything of IP except a common addressing scheme
  - Minimal best-effort datagram service
  - Variety of networks underneath and variety of transport services above IP (“thin waist”)

Seven second level goals

Internet must

4. permit distributed management of resources
5. efficiently use resources
6. permit low-effort host attachment
7. enable accountable resource usage

Q: What are implications, strengths, and drawbacks of these design choices today?

Roadmap

- A brief history of the Internet
- Circuits vs. packets
- Design philosophy of Internet protocols
- Placement of functionality
  - End-to-end principle
  - Layering

End-to-end principle

Avoid complex functionality at lower layers unless critical for performance

- Implications/examples?
  - End-to-end reliability and error-recovery
  - End-to-end encryption, integrity check, and authentication
  - Avoid in-network duplicate suppression
  - End-to-end FIFO message delivery

Protocol "Layers"

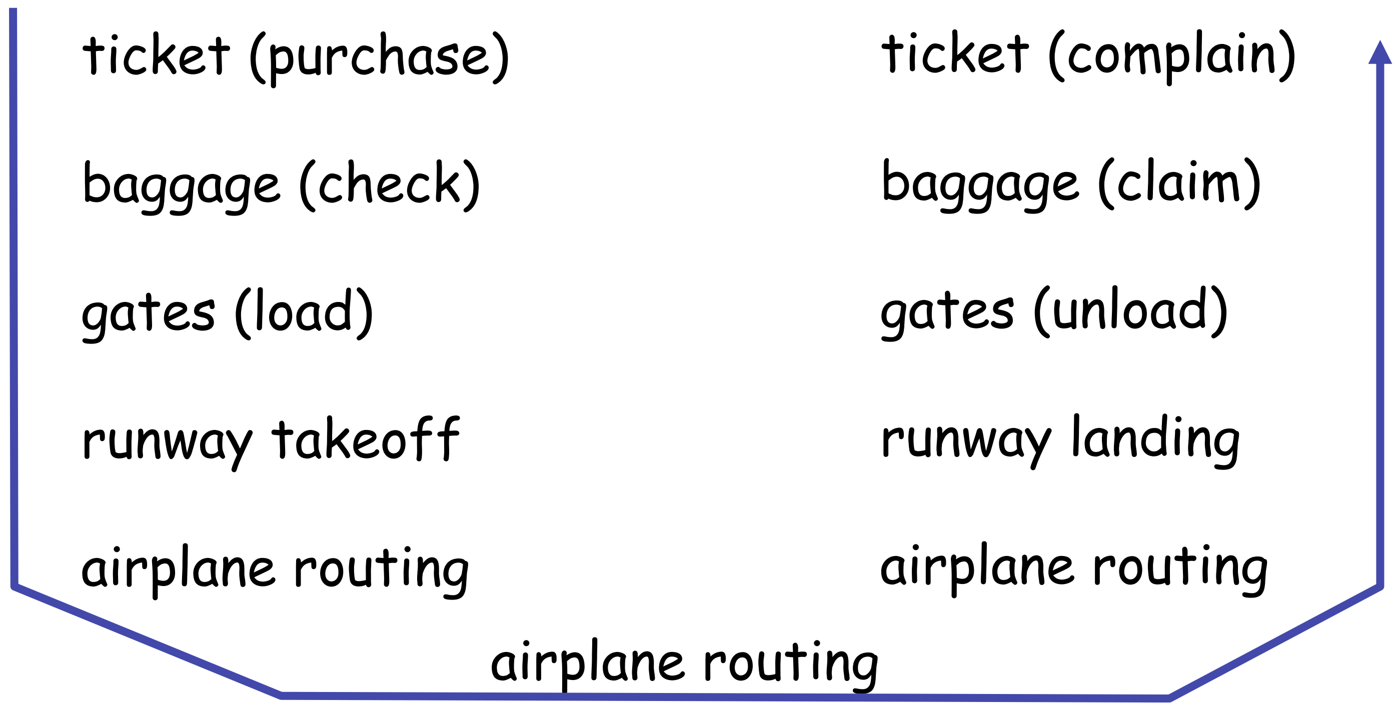
<u>Networks are complex!</u>

- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

<u>Question:</u>

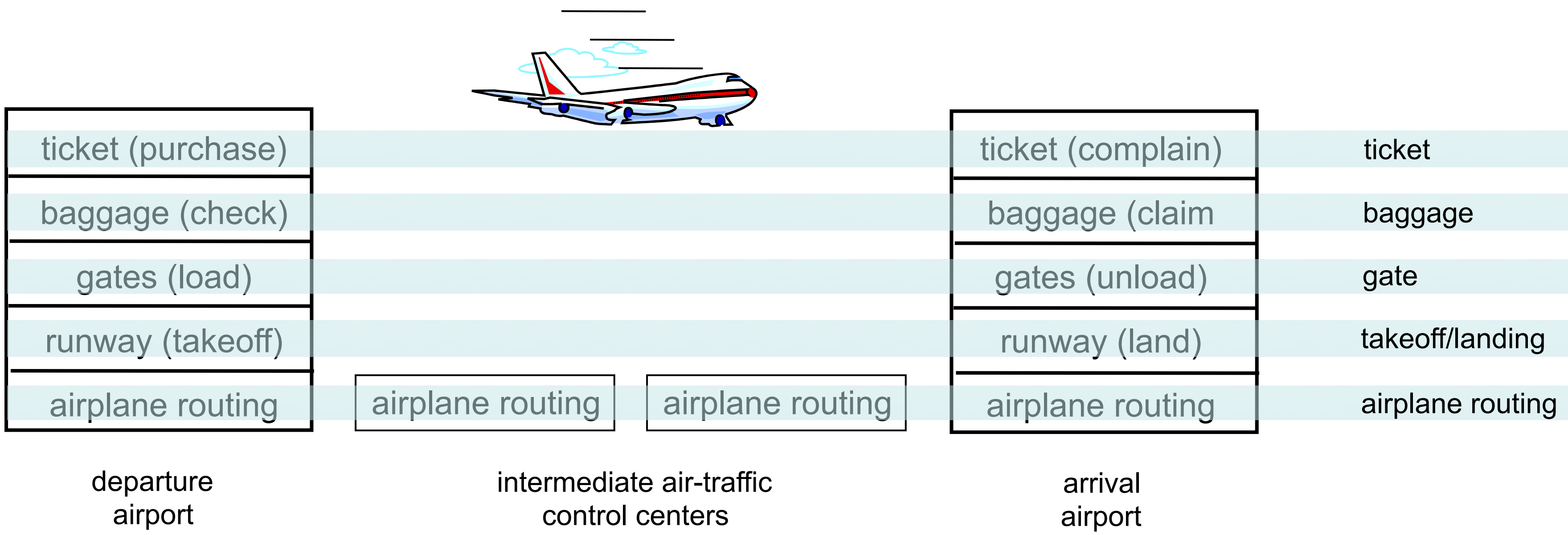
Is there any hope of *organizing* structure of network?

Organization of air travel



- a series of steps

Layering of airline functionality



Layers: each layer implements a service

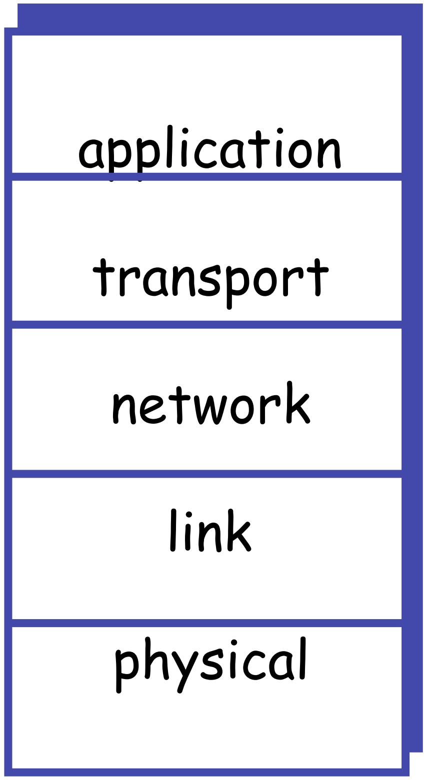
- via its own internal-layer actions
- relying on services provided by layer below

Layering to combat complexity

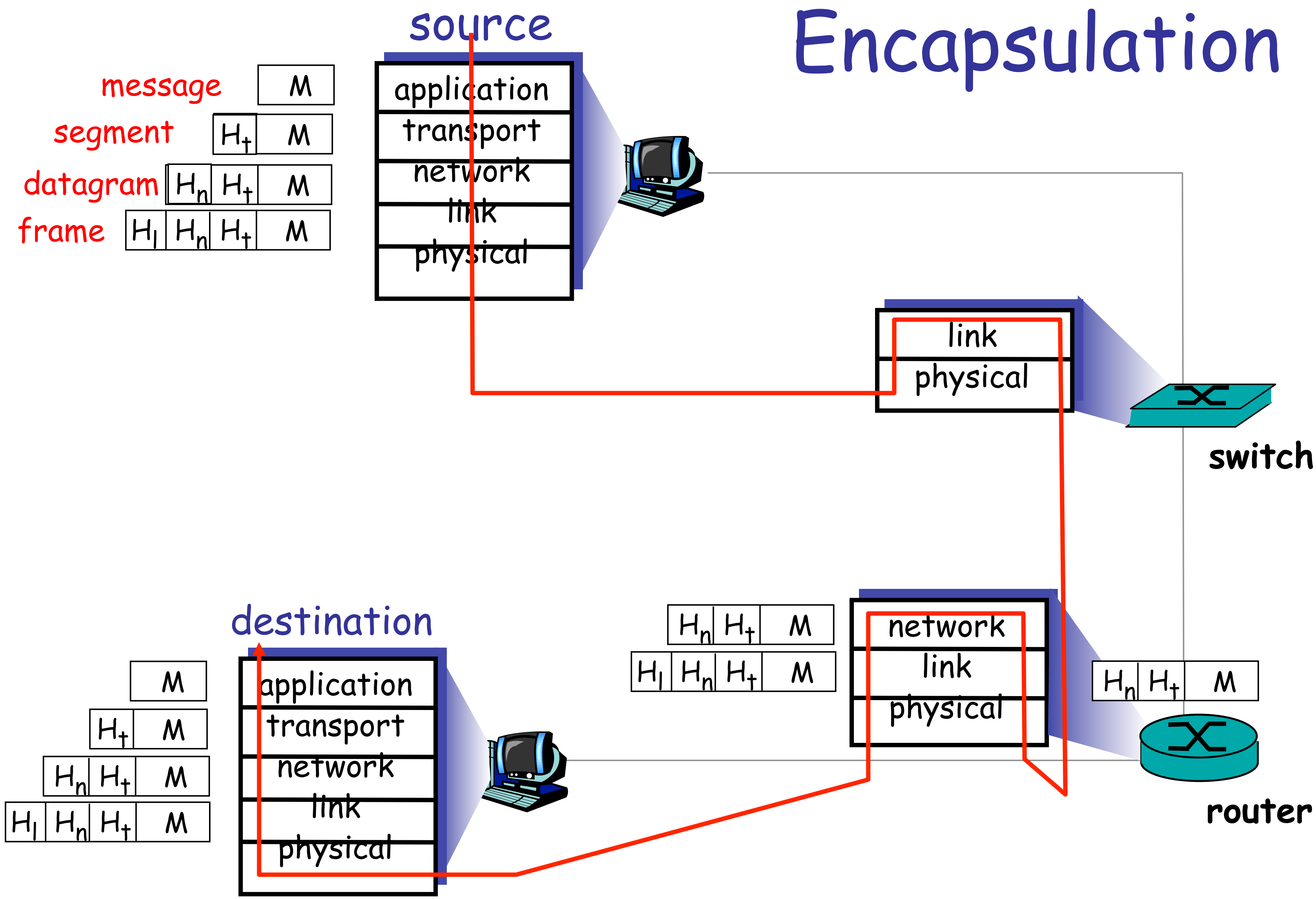
- Explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- Modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system in an airline system
- Layering considered harmful?

Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits "on the wire"



Encapsulation



Application layer framing (ALF)

- Motivation: what if an app wants selective reliability, eg multimedia reference frames?
  - TCP is reliable, UDP is unreliable
  - Not easy to modify TCP for selective reliability
- Problem: no common identification scheme between application and transport
- Solution: ALF proposes application data units (ADUs) to address this problem [CT90]
  - Inevitably(?) violates clean layering