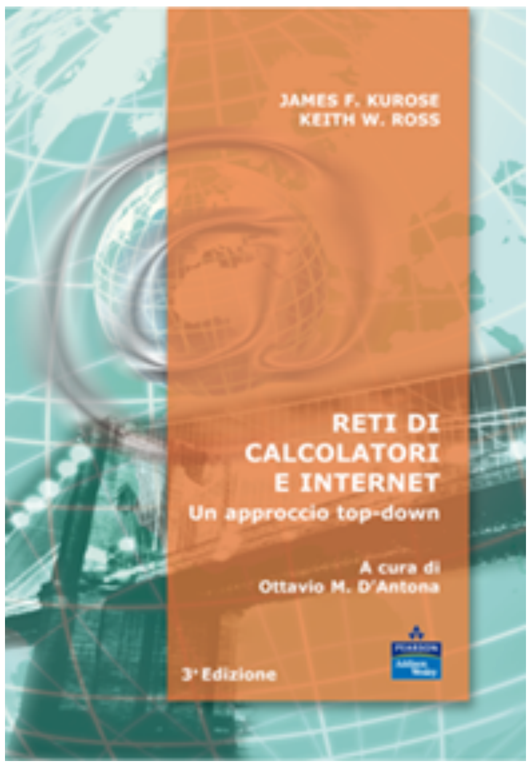
Reti di calcolatori: introduzione (Capitolo 1 Kurose-Ross)

Marco Roccetti

26 Febbraio 2024

(Capitolo 1 Kurose-Ross)



Reti di calcolatori e Internet:
Un approccio top-down

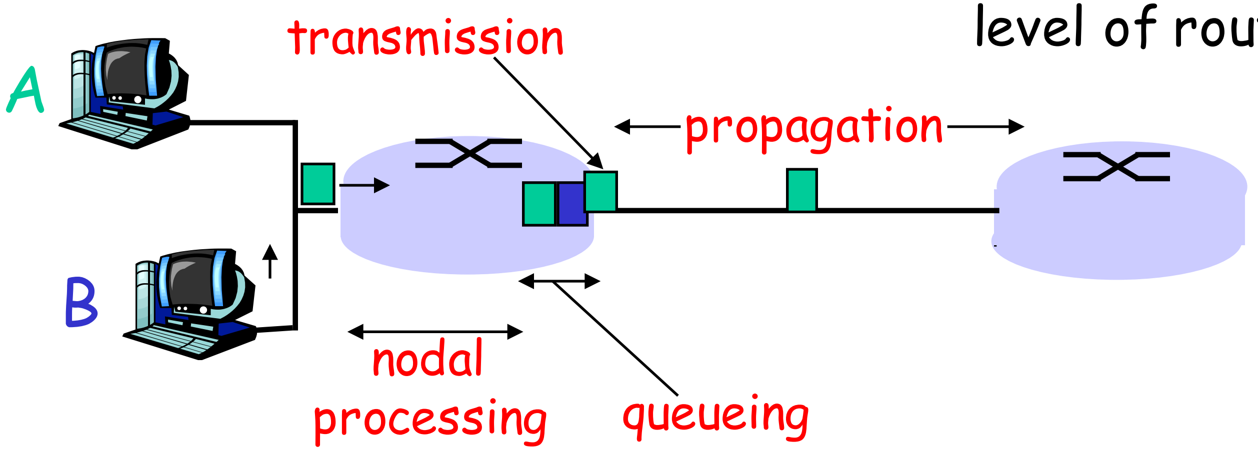
3ª edizione
Jim Kurose, Keith Ross
Pearson Education Italia
©2005

Delay in packet-switched networks

packets experience delay on end-to-end path

- four sources of delay at each hop

- nodal processing:
  - check bit errors
  - determine output link
- queueing
  - time waiting at output link for transmission
  - depends on congestion level of router



Delay in packet-switched networks

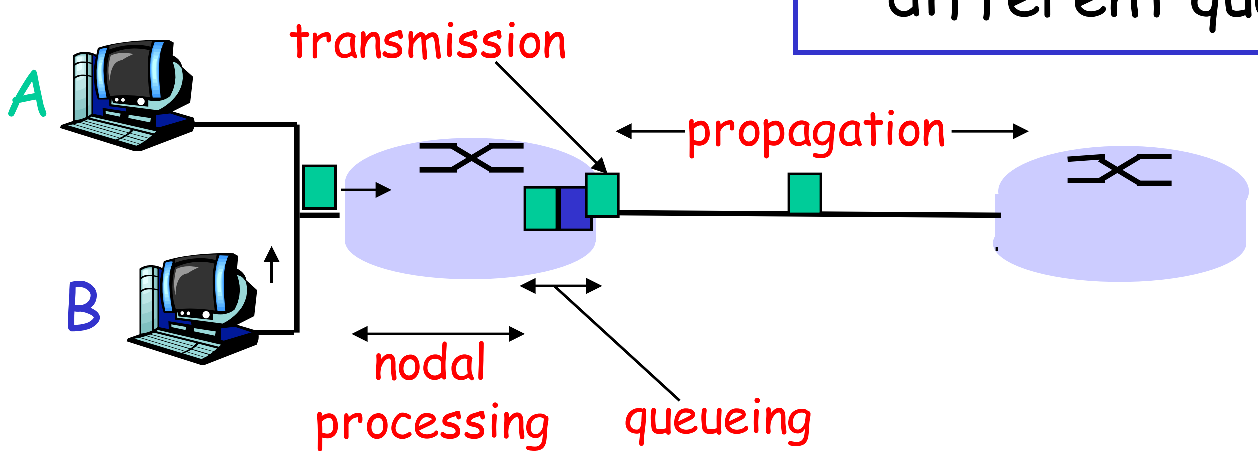
Transmission delay:

- R=link bandwidth (bps)
- L=packet length (bits)
- time to send bits into link = L/R

Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~2×10^8 m/sec)
- propagation delay = d/s

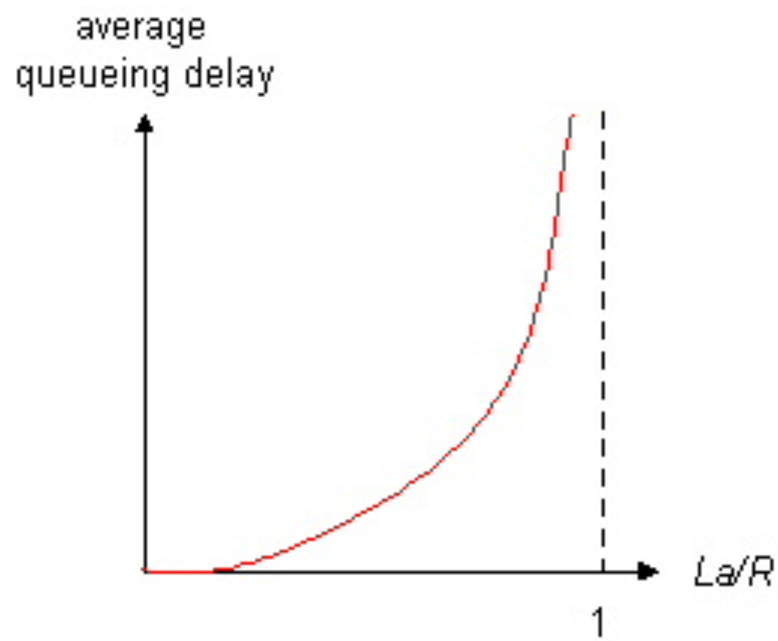
Note: s and R are *very* different quantitites!



Queueing delay (revisited)

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate

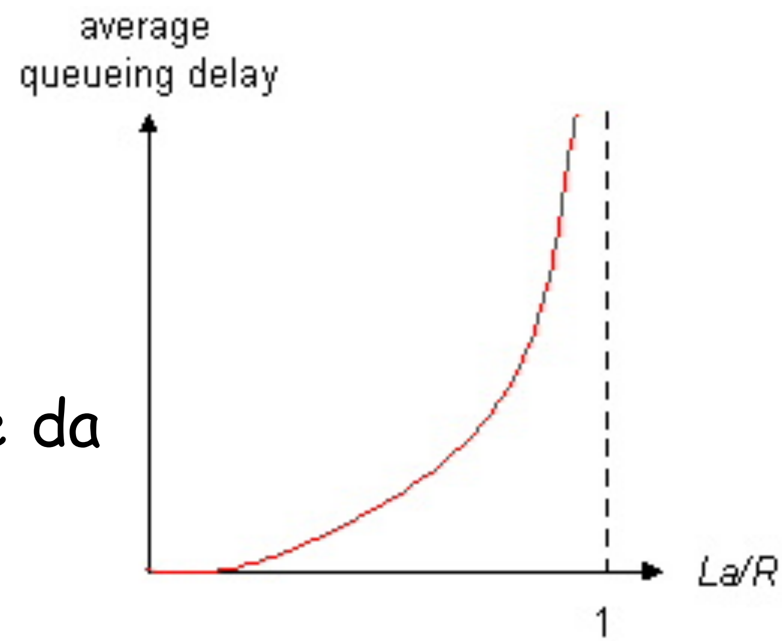
traffic intensity = L*a/R



- La/R ~ 0: average queueing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!

Packet Loss

- Tra l'altro: non si perderebbero pacchetti se la coda fosse di dimensione infinita!
- Purtroppo le code hanno dimensione finita, e dunque da un certo punto in poi i pacchetti possono essere persi: packet loss!
- Usate il programma trace route (RFC 1393) per fare un po' di prove su internet vera dove i pacchetti si accodano su N diversi router lungo una route



traffic intensity = L*a/R

Protocol "Layers"

Networks are complex!

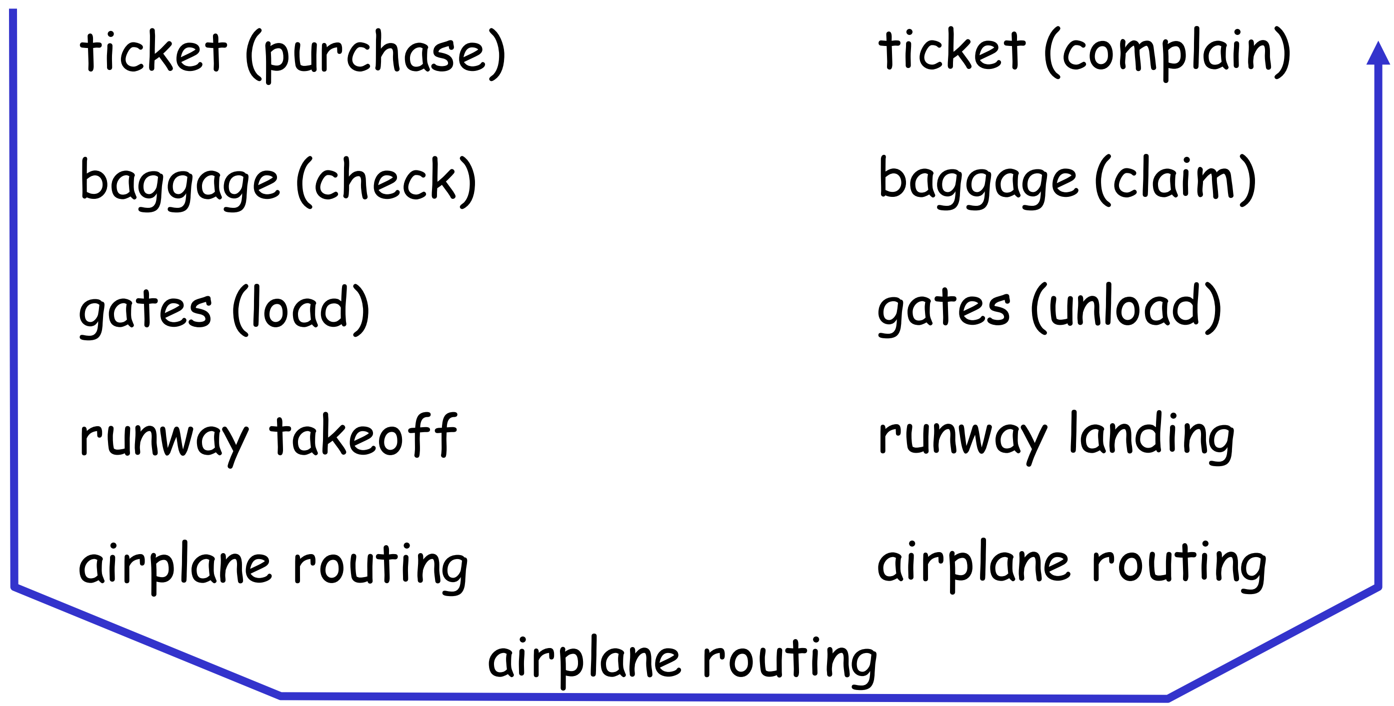
- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

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

Or at least our discussion of networks?

Organization of air travel



- a series of steps

Organization of air travel: a different view

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing
airplane routing	

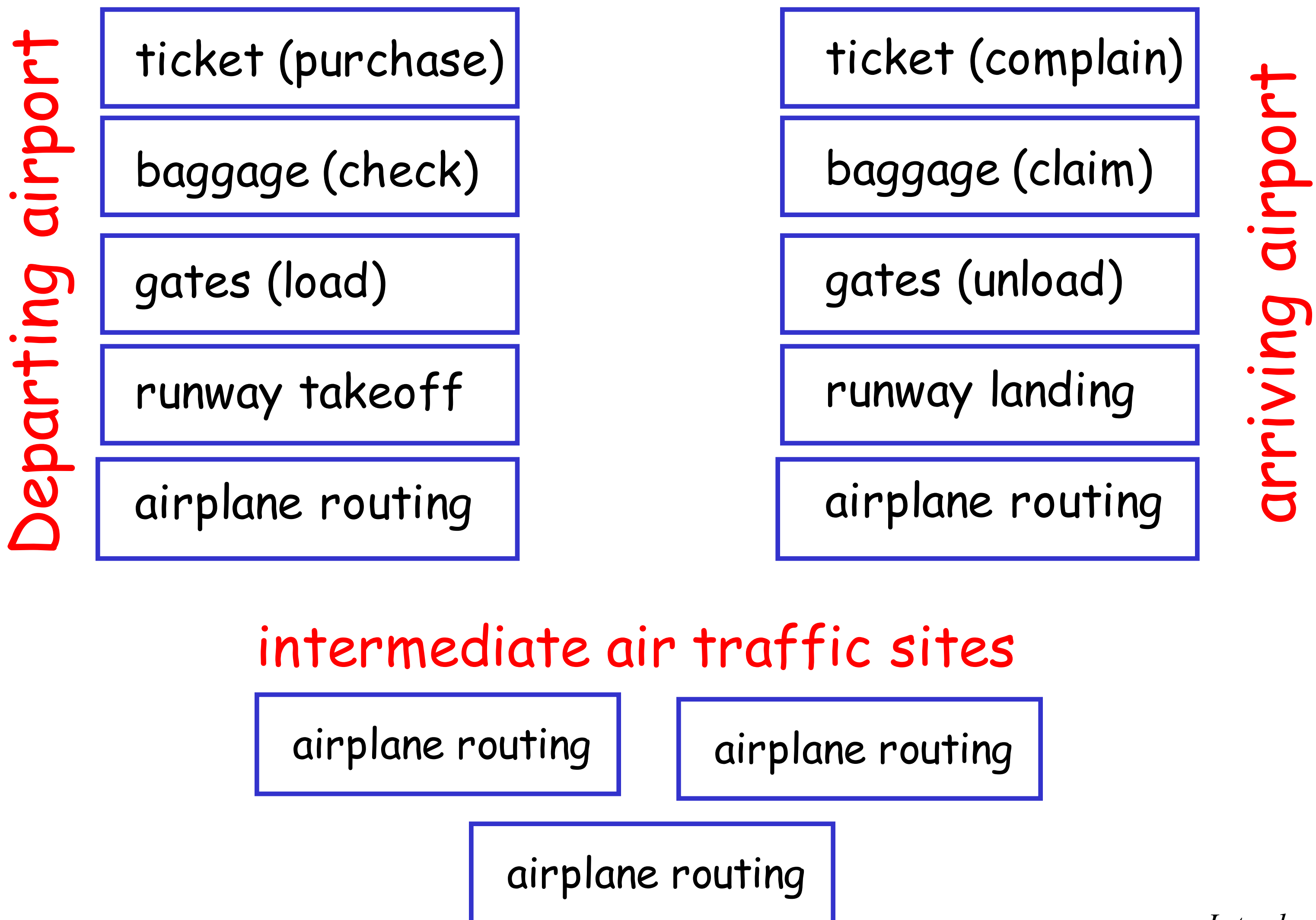
Layers: each layer implements a service

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

Layered air travel: services

Counter-to-counter delivery of person+bags
baggage-claim-to-baggage-claim delivery
people transfer: loading gate to arrival gate
runway-to-runway delivery of plane
airplane routing from source to destination

Distributed implementation of layer functionality



Why layering?

Dealing with complex systems:

- 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
- layering considered harmful? Beware of duplications

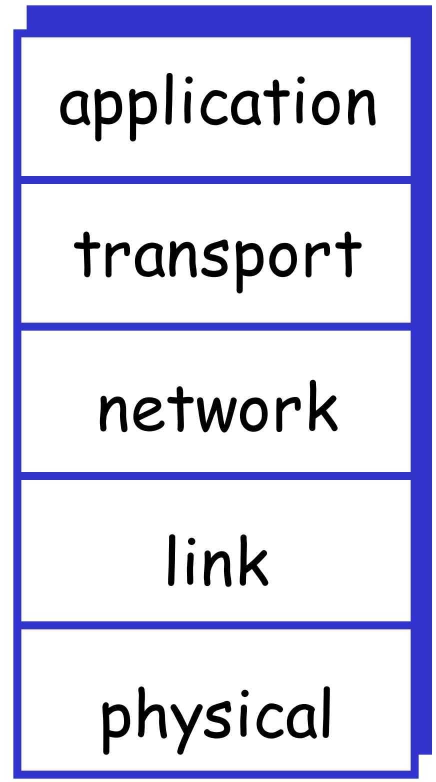
How to place functions among layers?

- End-to-end argument: functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level.
- moving functions upward in a layered system, closer to the application that uses the function

- Higher-level layers, more specific to an application, are free to (and thus expected to) organize lower-level network resources to achieve application-specific design goals efficiently. (application autonomy)
- Lower-level layers, which support many independent applications, should provide only resources of broad utility across applications, while providing to applications usable means for effective sharing of resources and resolution of resource conflicts. (network transparency)

Internet protocol stack

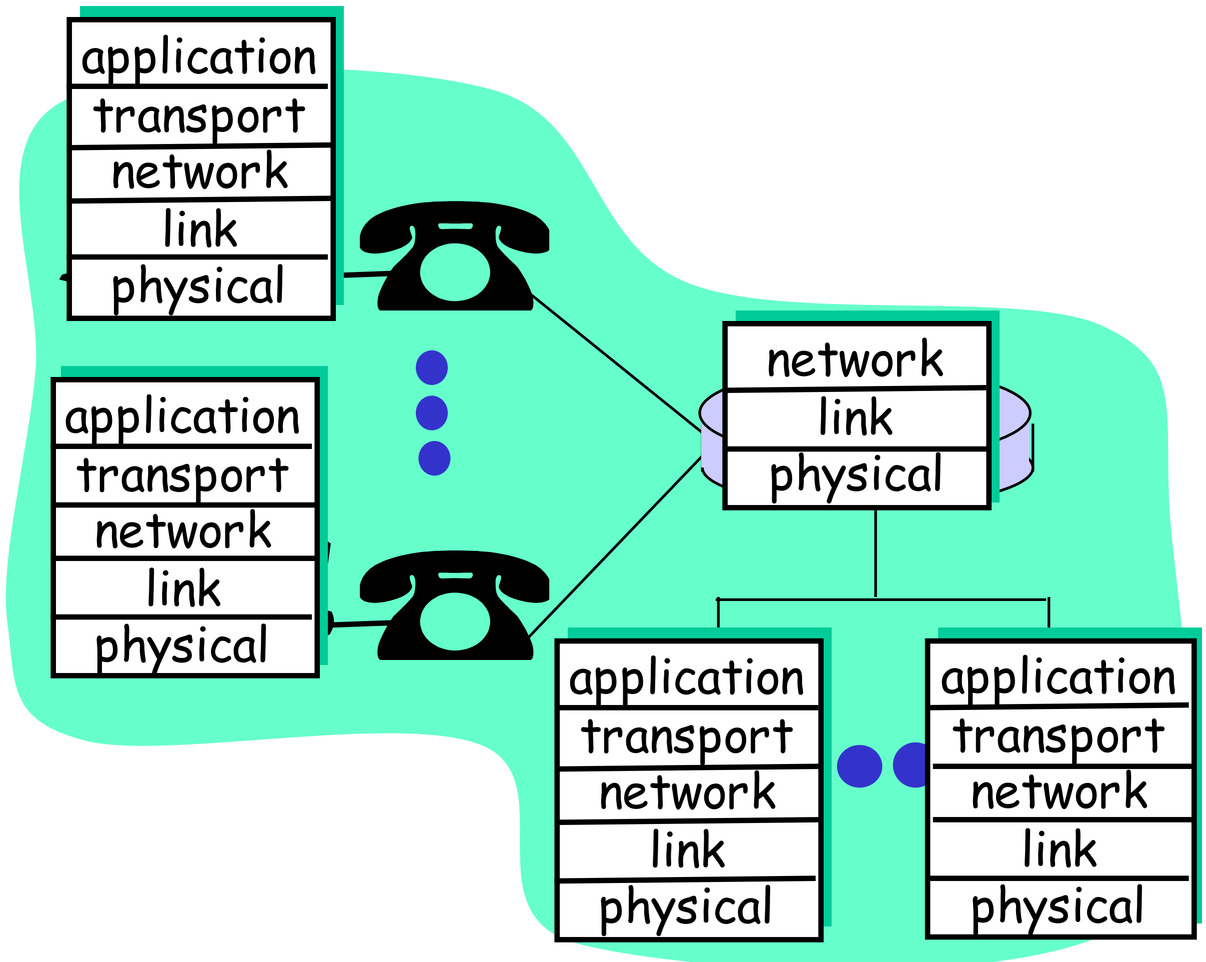
- application: supporting network applications (data)
  - ftp, smtp, http
- transport: host-host data transfer (segment)
  - tcp, udp
- network: routing of datagrams from source to destination
  - ip, routing protocols
- link: data (frame) transfer between neighboring network elements
  - ppp, ethernet
- physical: bits "on the wire"



Layering: logical communication

Each layer:

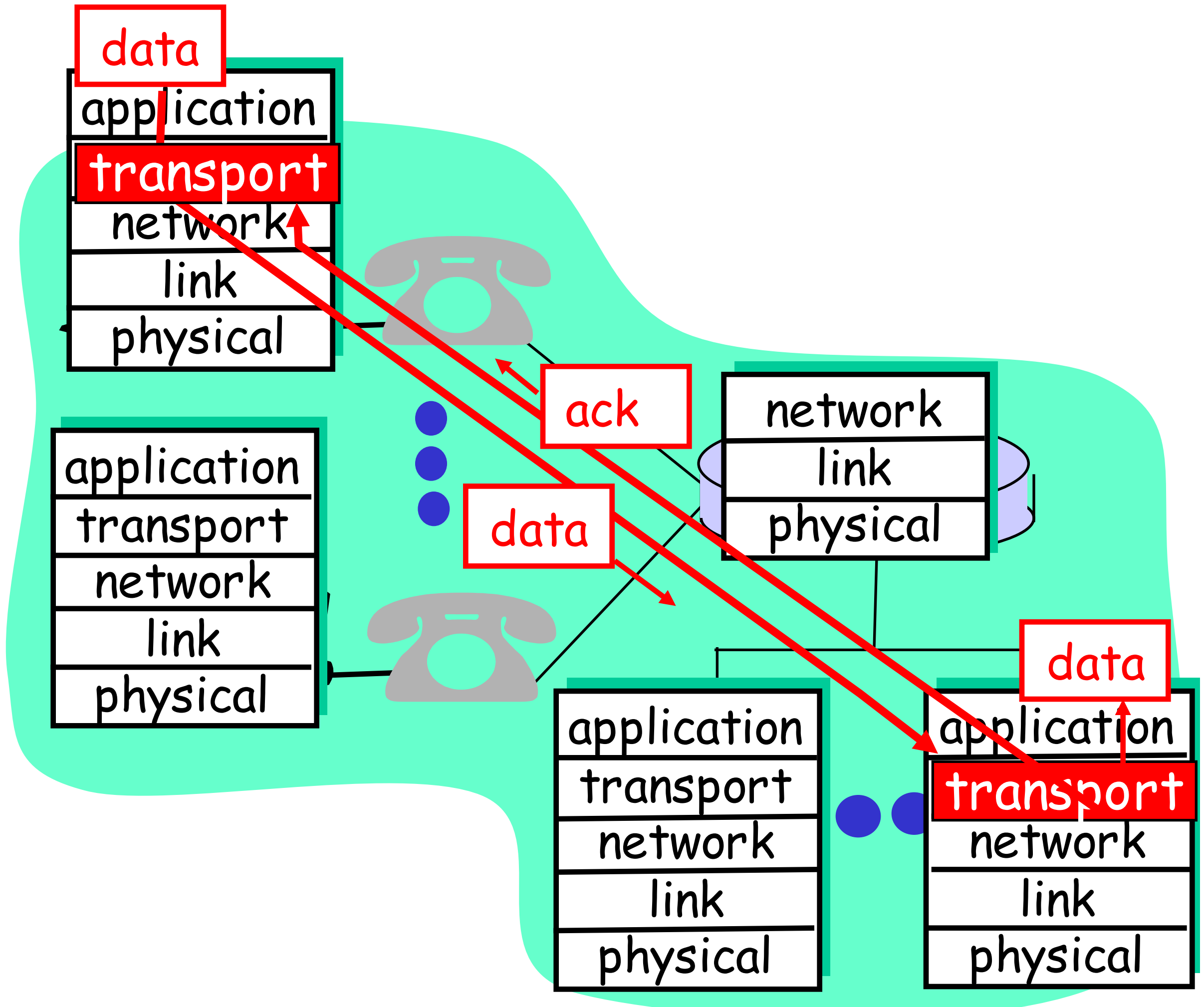
- distributed "entities" implement layer functions at each node
- entities perform actions, exchange messages with peers



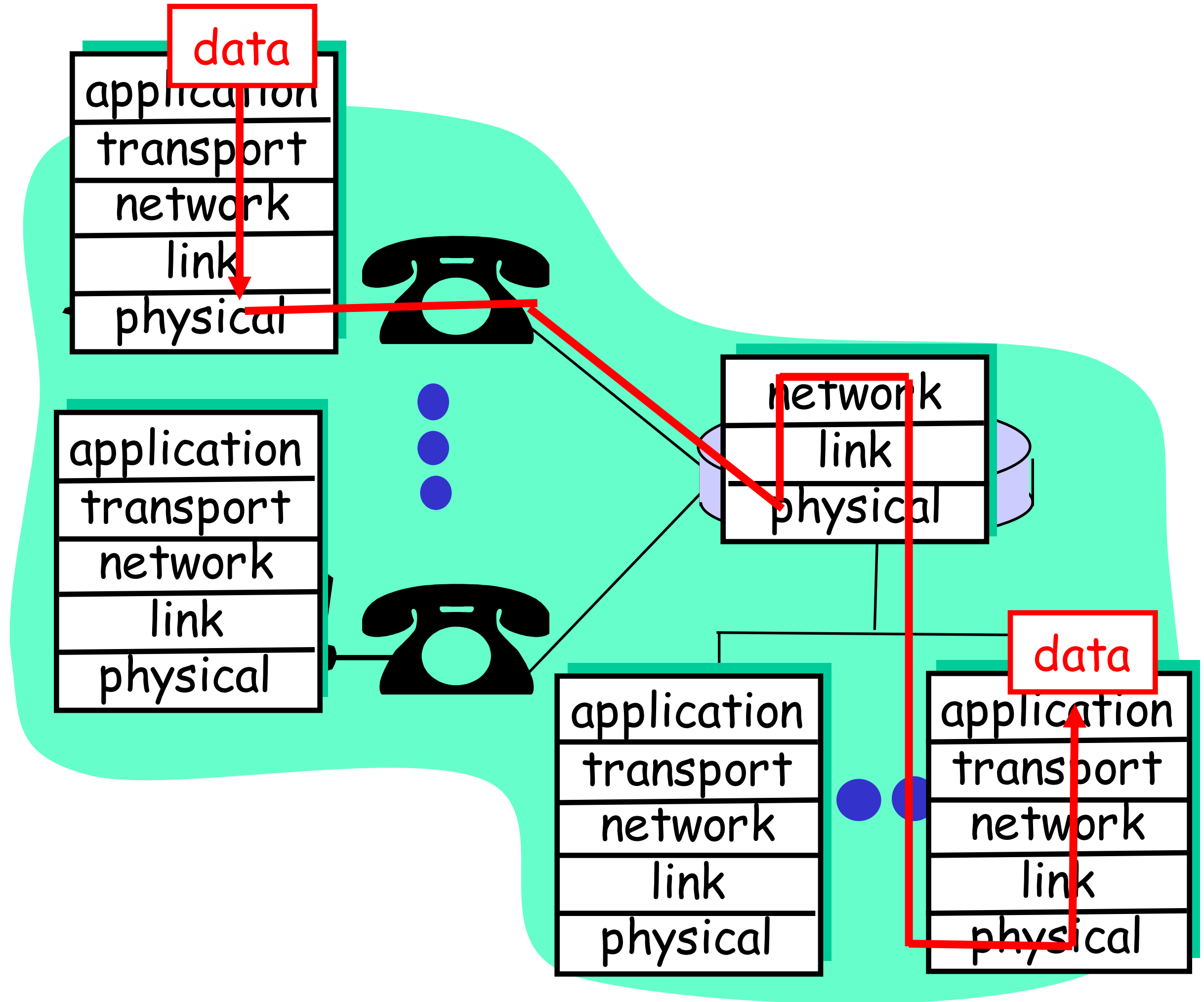
Layering: *logical* communication

E.g.: transport

- take data from app
- add addressing, reliability check info to form "datagram"
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



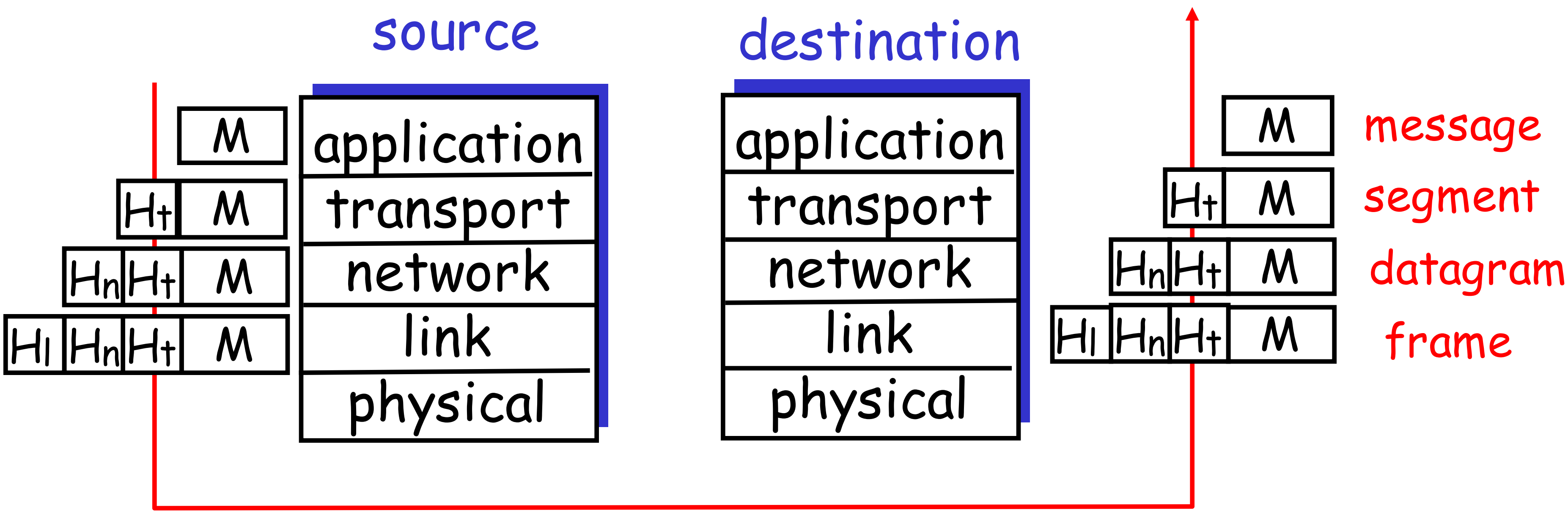
Layering: physical communication



Protocol layering and data encapsulation

Each layer takes data from above

- adds header information to create new data unit
- passes new data unit to layer below



Internet History

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 Reearch Projects Agency
- 1969: first ARPAnet node operational, Labor day
- Supervisionato da Kleinrock, 4 nodi UCLA, S. Barbara, SRI e Utah, primo messaggio LO

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

Internet History

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1973: Metcalfe's PhD thesis proposes Ethernet
- 1974: Cerf and Kahn - architecture for interconnecting networks (TCP)
- 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

Internet History

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

Internet History

1990's: commercialization, the WWW

- Early 1990's: ARPAnet decomissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: WWW
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, http: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the WWW

Late 1990's:

- est. 50 million computers on Internet
- est. 100 million+ users
- backbone links runnning at 1 Gbps

ATM: Asynchronous Transfer Mode nets

Internet:

- today's *de facto* standard for global data networking

1980's:

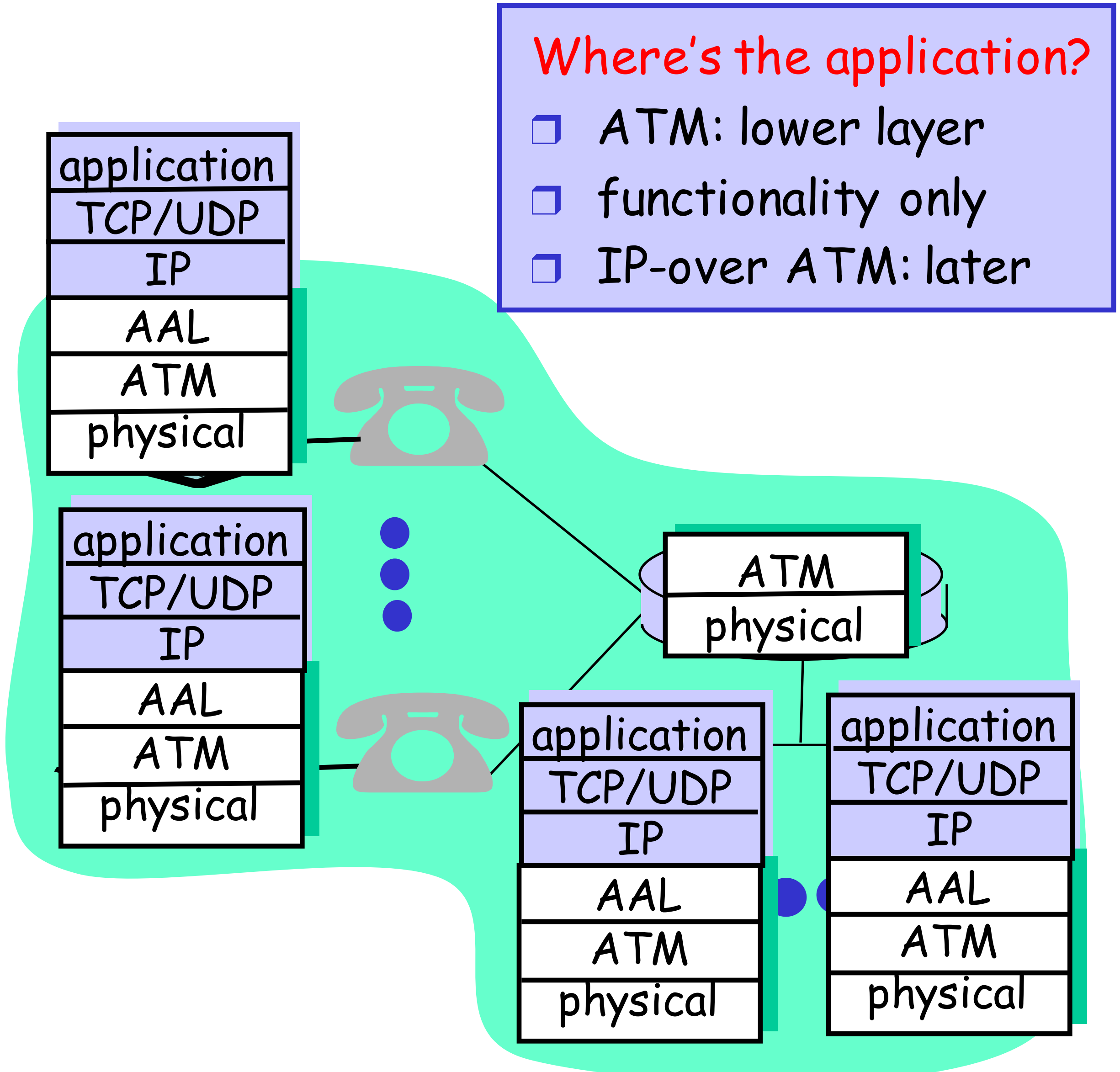
- telco's develop ATM: competing network standard for carrying high-speed voice/data
- standards bodies:
  - ATM Forum
  - ITU

ATM principles:

- small (48 byte payload, 5 byte header) fixed length *cells* (like packets)
  - fast switching
  - small size good for voice
- virtual-circuit network: switches maintain state for each "call"
- well-defined interface between "network" and "user" (think of telephone company)

ATM layers

- ATM Adaptation Layer (AAL): interface to upper layers
  - end-system
  - segmentation/reassembly
- ATM Layer: cell switching
- Physical



Chapter 1: Summary

Covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
- performance: loss, delay
- layering and service models
- backbones, NAPs, ISPs
- history
- ATM network

You now hopefully have:

- context, overview, "feel" of networking
- more depth, detail *later* in course