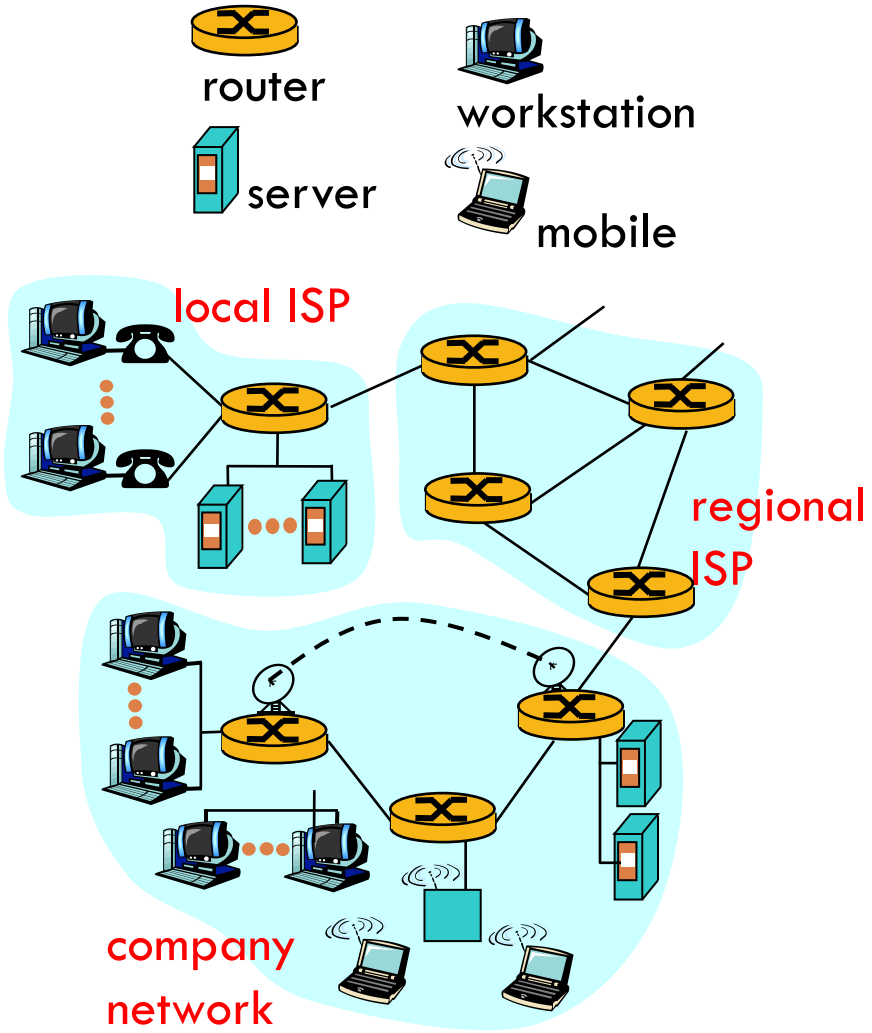
LECTURE 2

Networking Primer

Visualizing the Internet

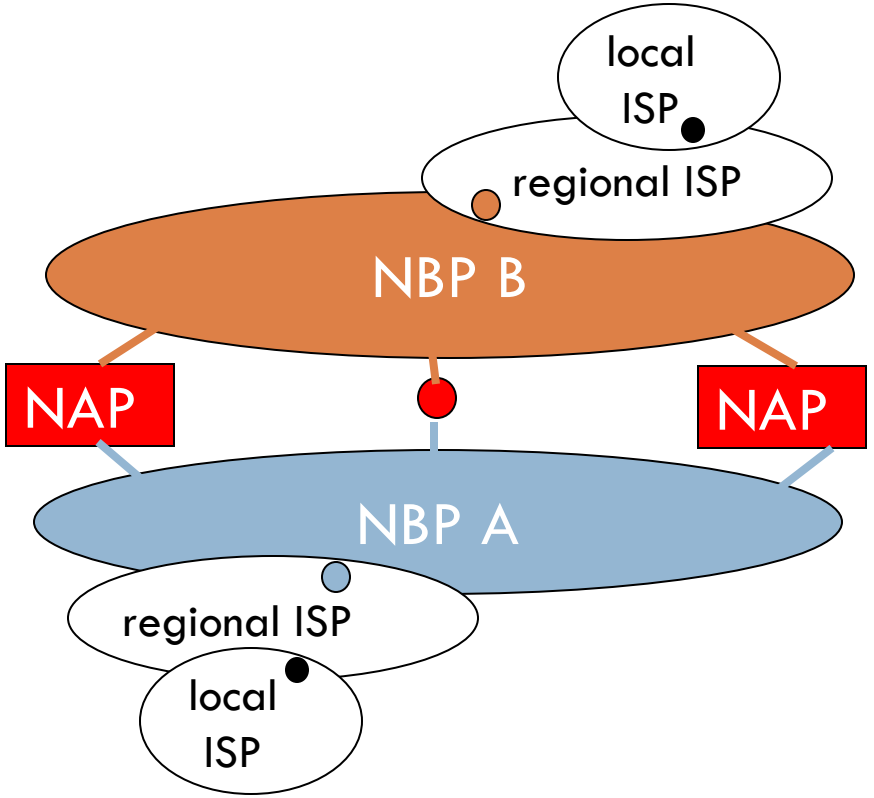
- millions of connected computing devices: *hosts, end-systems*
  - pc's workstations, servers
  - Tablets, smartphones, toasters

  running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
- *routers:* forward packets (chunks) of data thru network



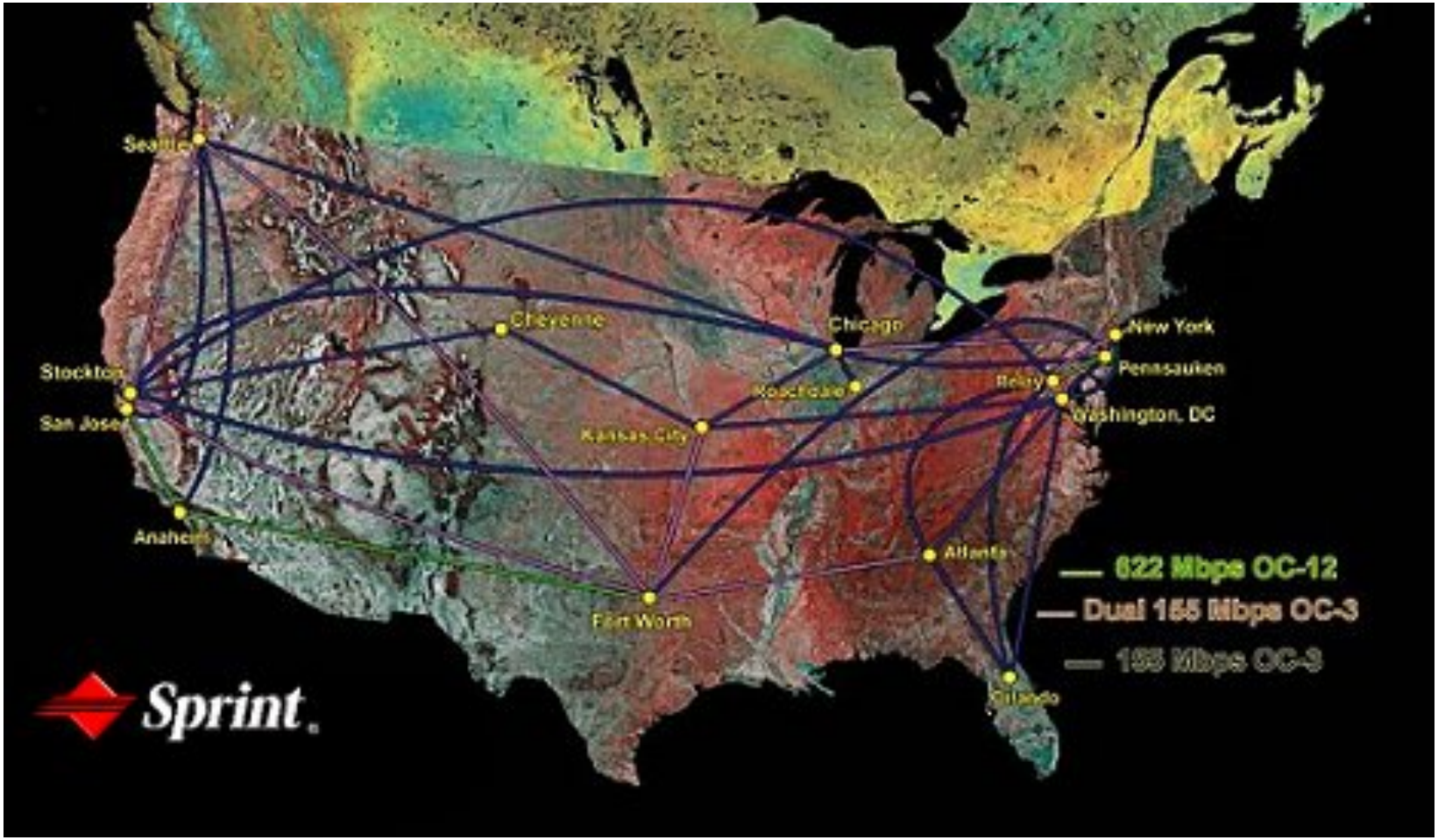
Internet structure: network of networks

- roughly hierarchical
- national/international backbone providers (NBPs)
  - e.g. BBN/GTE, Sprint, AT&T, IBM, UUNet
  - interconnect (peer) with each other privately, or at public Network Access Point (NAPs)
- regional ISPs
  - connect into NBPs
- local ISP, company
  - connect into regional ISPs



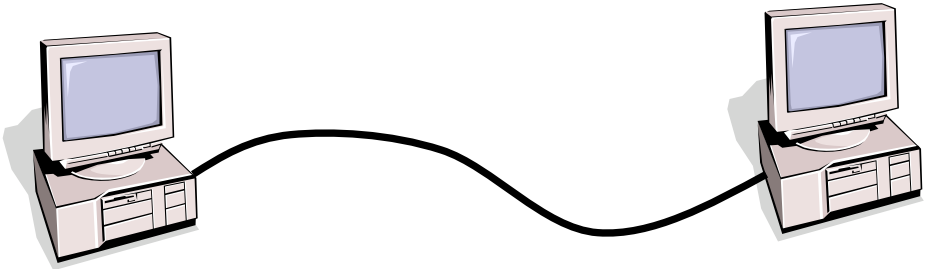
National Backbone Provider

e.g. Sprint US backbone network



Links between computers

- Point to point link -- a direct cable or wire that is dedicated for use between the computers.



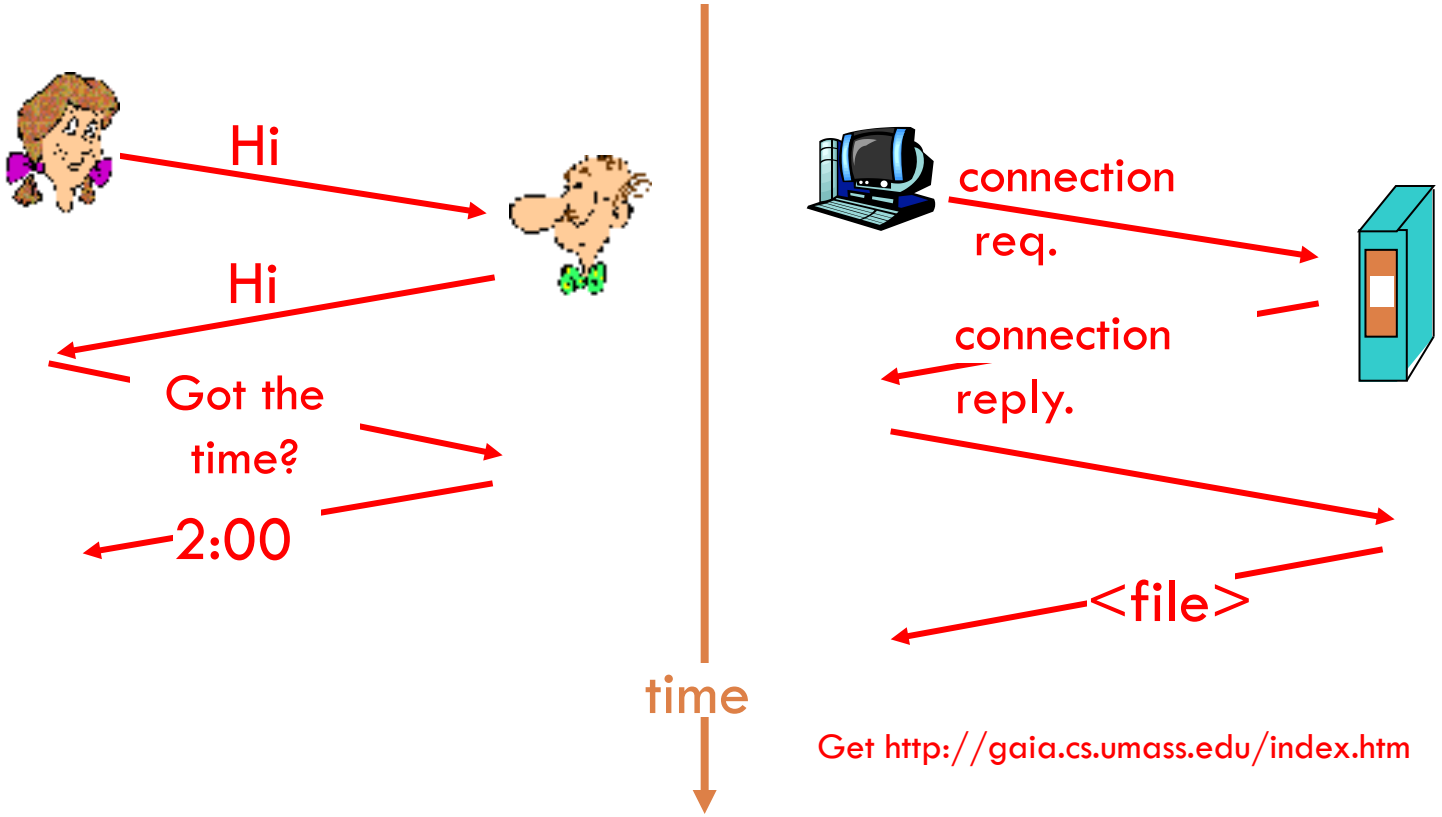
- Multiple access link -- shared between many nodes -- example a bus, wireless medium etc.

Why structure ?

- Direct connectivity does not scale -- all computers cannot be directly connected to each other.
- Need organization
  - End hosts -- clients and servers -- usually house information
  - Routers and switches -- nodes that are primarily used for relaying information -- sending information where it needs to go.

What's a protocol?

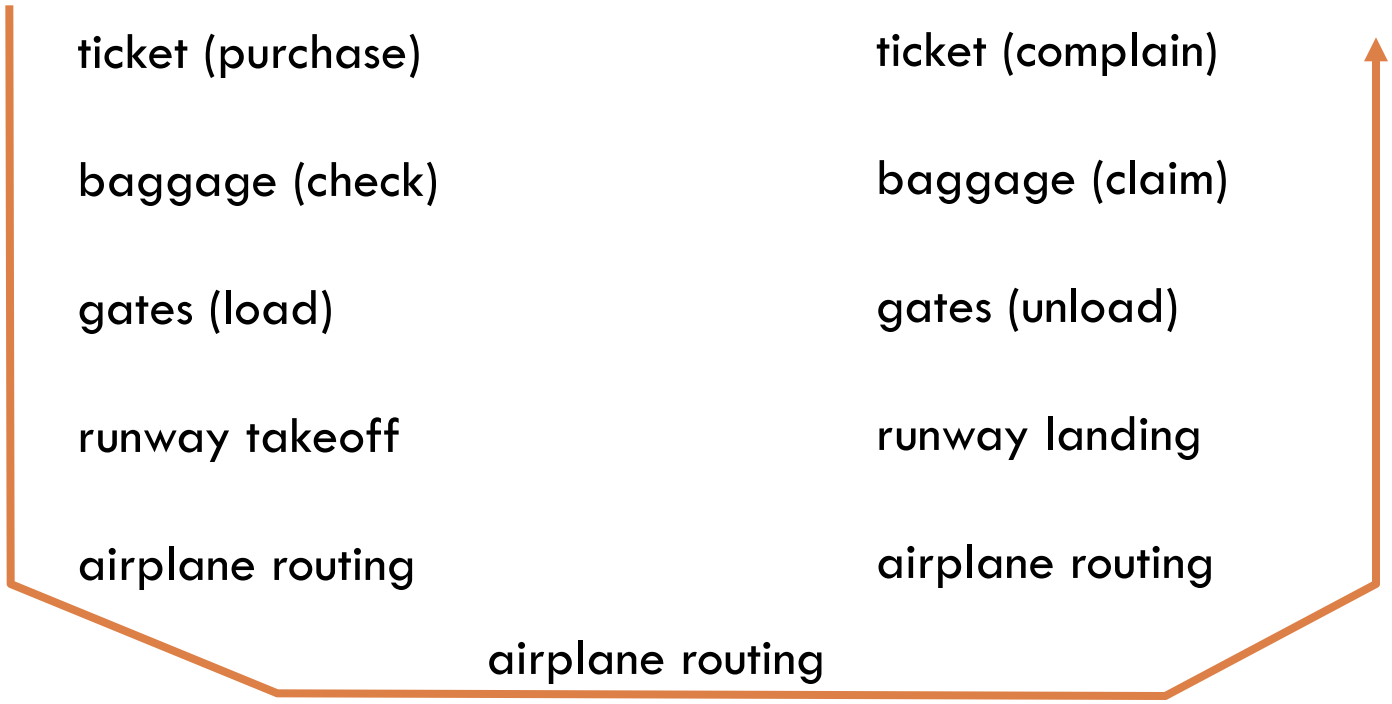
- **The definition of a behavior --set of rules**
- **Here: the format of a communication exchange:**
- **Sequence of actions, format of information, predefined interpretation**



Packets

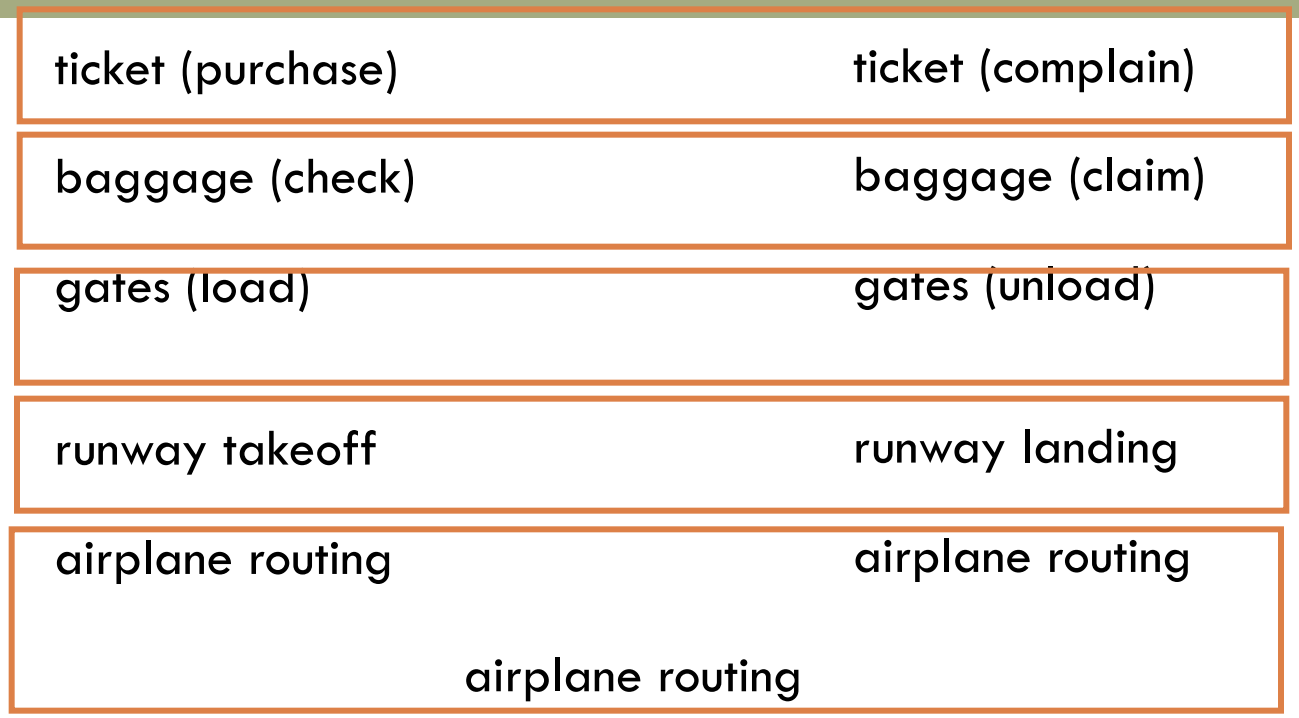
- Packets are similar to postal letters -basic units of information
  - From, to, content
  - Postman handles all packets similarly
  - Addressing is hierarchical.
- The protocol that defines how packets are to be “routed” is the Internet Protocol or IP.
  - Hierarchical addresses like in the postal world.

Organization of air travel



- a series of steps

Organization of air travel: a different view

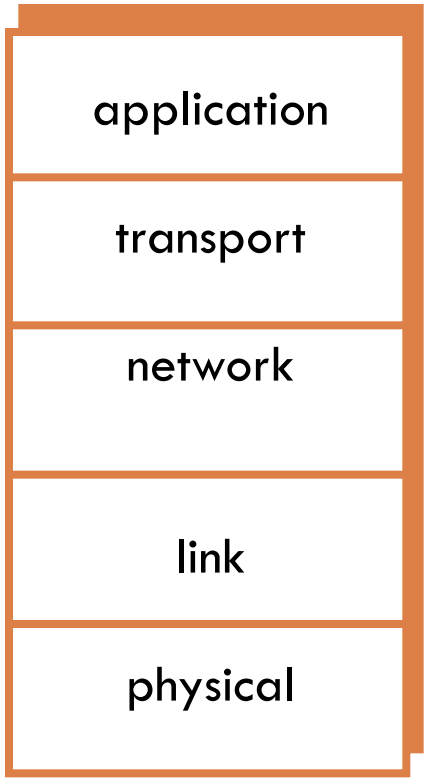


Layers: each layer implements a service

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

Internet protocol stack

- application: supporting network applications
  - ftp, smtp, http
- transport: host-host data transfer
  - tcp, udp
- network: routing of datagrams from source to destination
  - ip, routing protocols
- link: data transfer between neighboring network elements
  - ppp, ethernet, wireless link
- physical: bits "on the wire or wireless" -- representation, signal modulation.



Why layering?

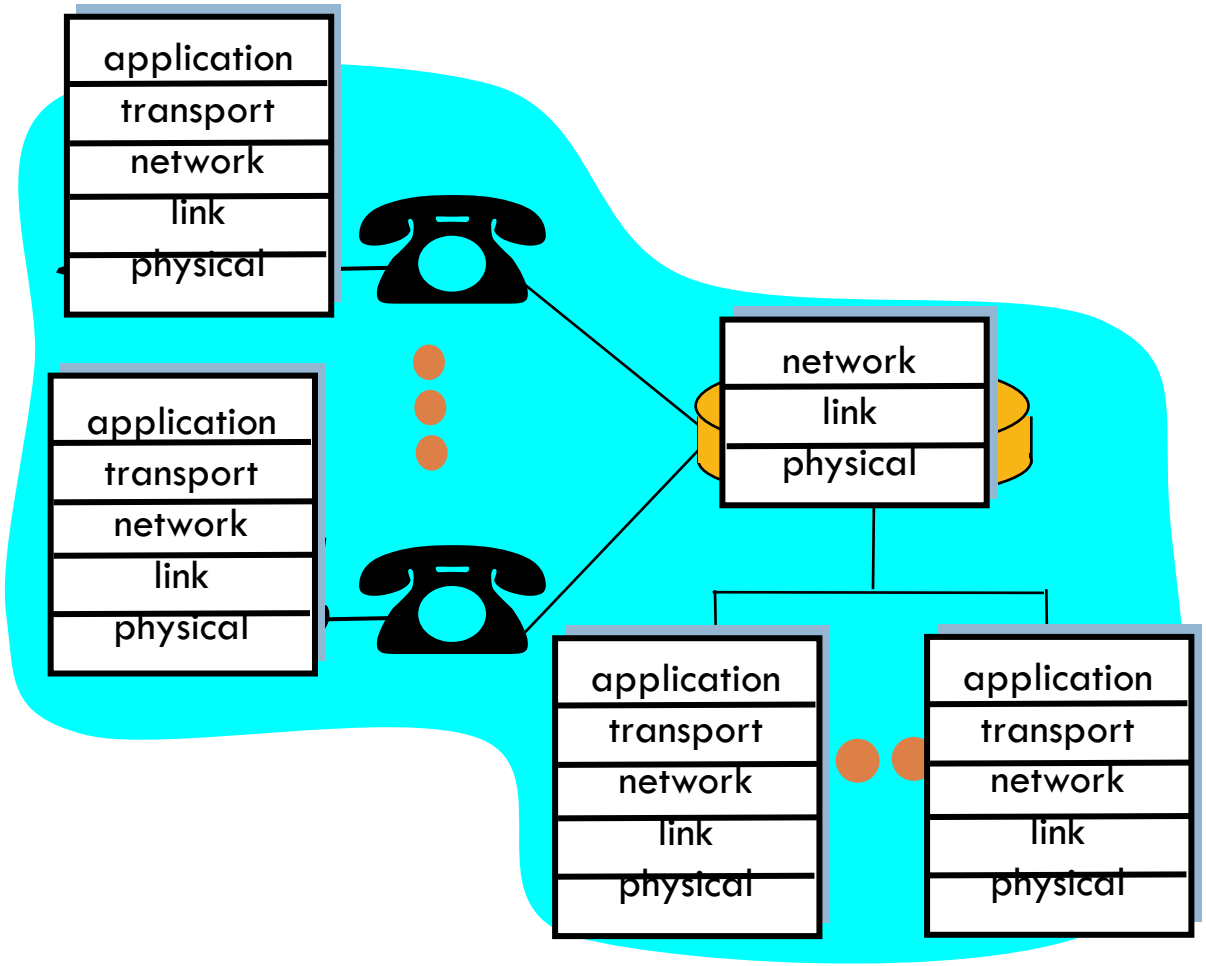
Dealing with complex systems:

- 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
- Isolating "functions" and interactions components
  - layered reference model for discussion

Layering: logical communication

Each layer:

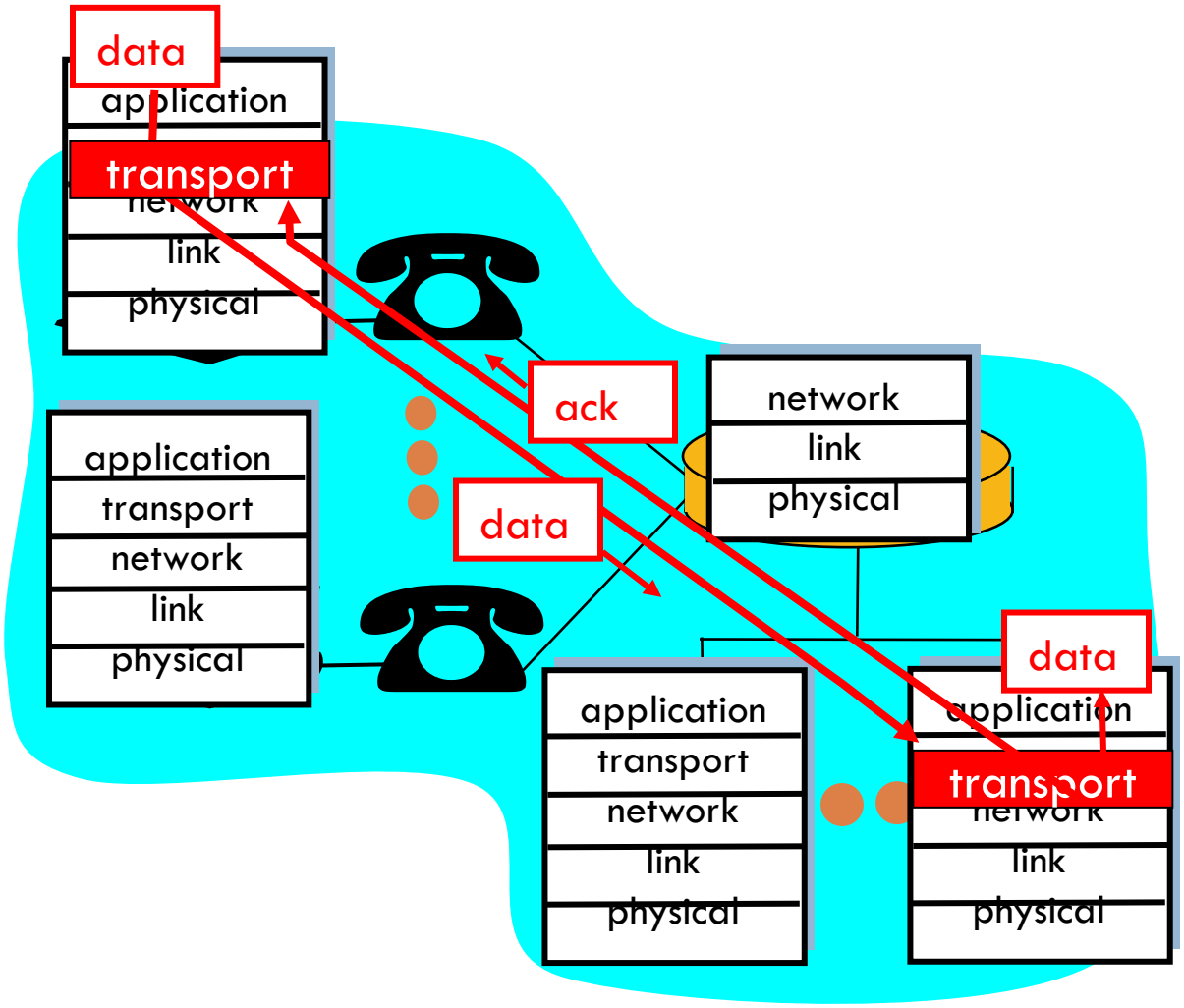
- distributed
- "entities" implement layer functions at each node
- entities perform actions, exchange messages with peers



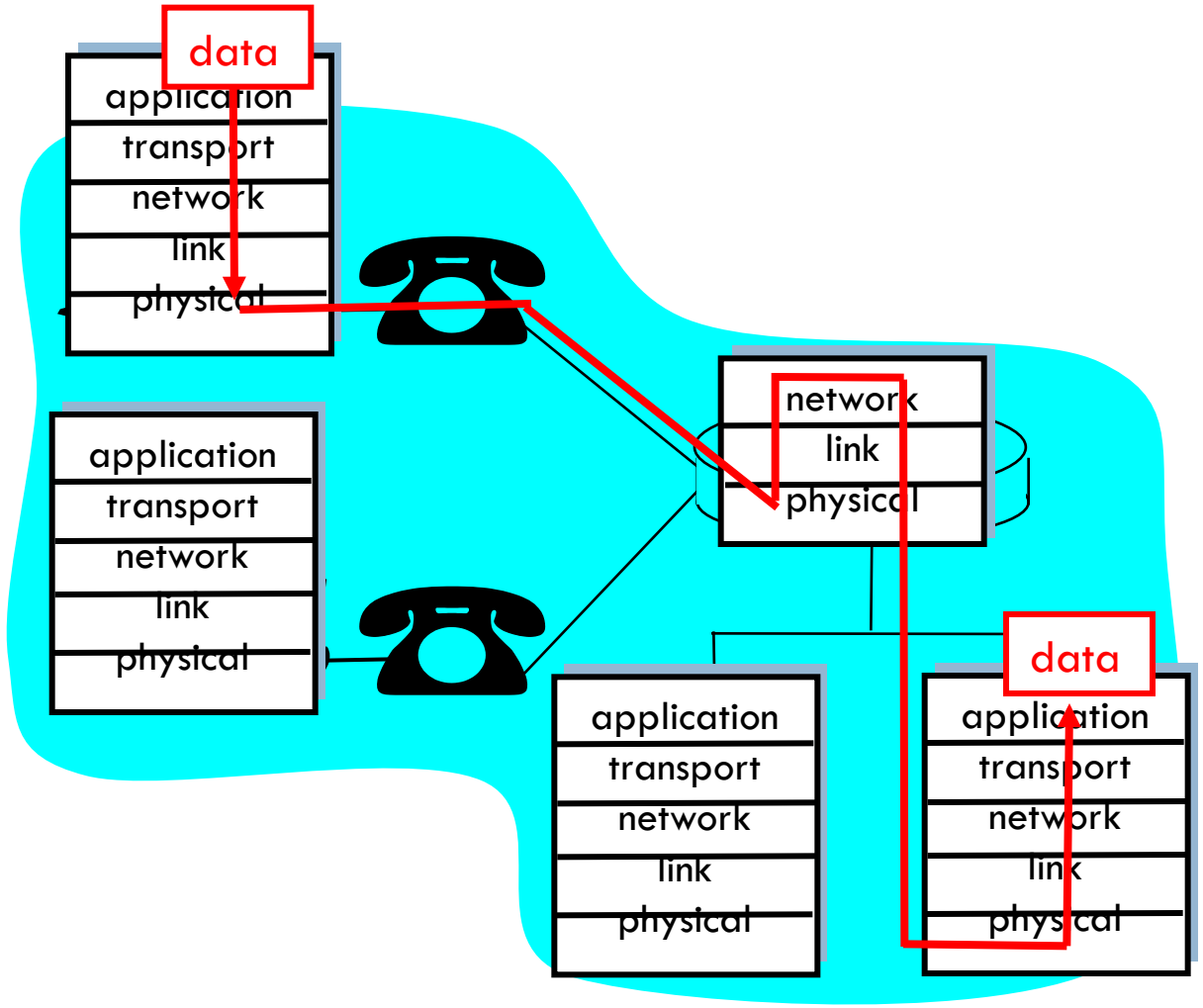
Layering: *logical* communication

E.g.: transport

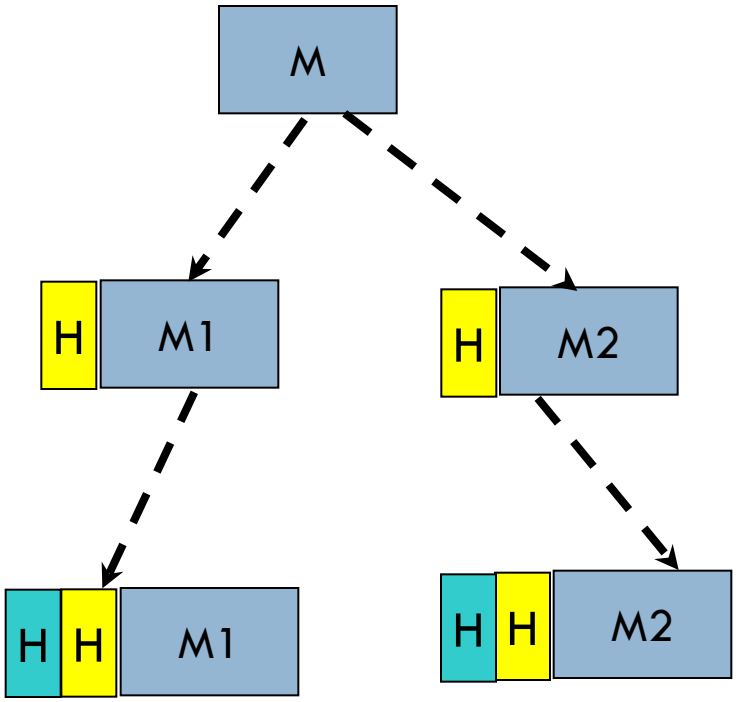
- take data from app
- add addressing, reliability check info to form "datagram"
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



Layering: physical communication



Message flow through stack

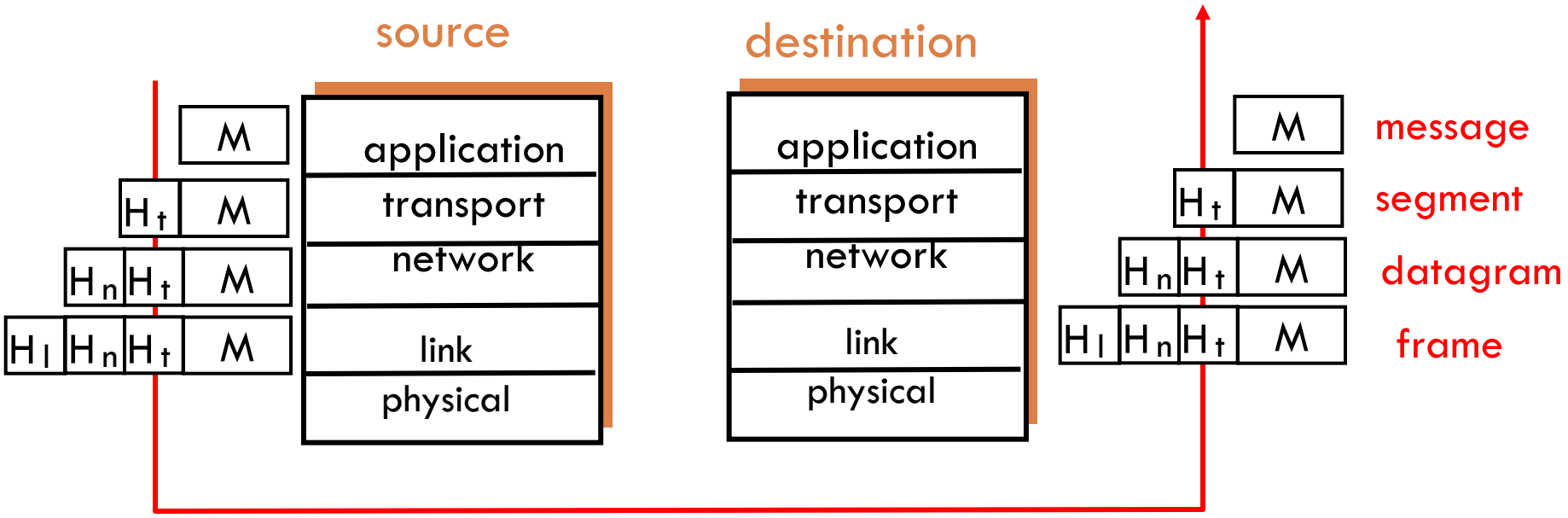


- At each layer, headers added.
- A protocol defines maximum packet size -- might require higher layer packet to be fragmented.

Protocol layering and data

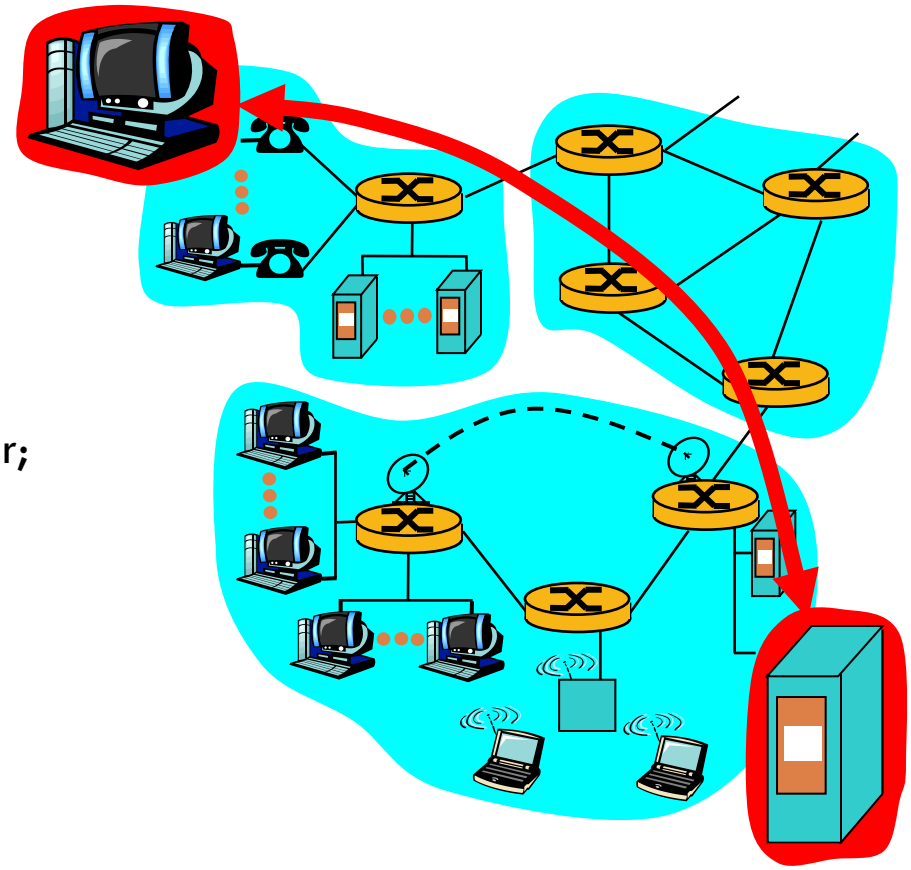
Each layer takes data from above

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



The network edge:

- end systems (hosts):
  - run application programs
  - e.g., WWW, email
  - at "edge of network"
- client/server model
  - client host requests, receives service from server
  - e.g., WWW client (browser)/ server; email client/server
- peer-peer model:
  - host interaction symmetric
  - e.g.: Gnutella, KaZaA



Network edge: connection-oriented service

Goal: data transfer between end sys.

- *handshaking:* setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - *set up "state"* in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet's connection-oriented service

TCP service [RFC 793]

- *reliable, in-order* byte-stream data transfer
  - loss: acknowledgements and retransmissions
- *flow control:*
  - sender won't overwhelm receiver
- *congestion control:*
  - senders "slow down sending rate" when network congested

Network edge: connectionless service

Goal: data transfer between end systems

- same as before!

- UDP - User Datagram Protocol [RFC 768]: Internet's connectionless service
  - unreliable data transfer
  - no flow control
  - no congestion control