Networking Refresher

Instructor: Khaled Diab

Rationale

- We previously studied:
  - vulnerabilities at one machine/endpoint

- What if a machine is connected to a “network”
  - What are possible vulnerabilities?

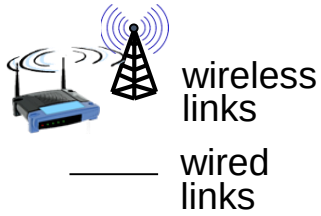
Outline

- Network Architecture
  - Components
  - Functionalities
  - Circuit switching vs. Packet switching
- Network Protocols
  - The need for layering
- Basics of Routing
- Network Security Roadmap

Network Architecture

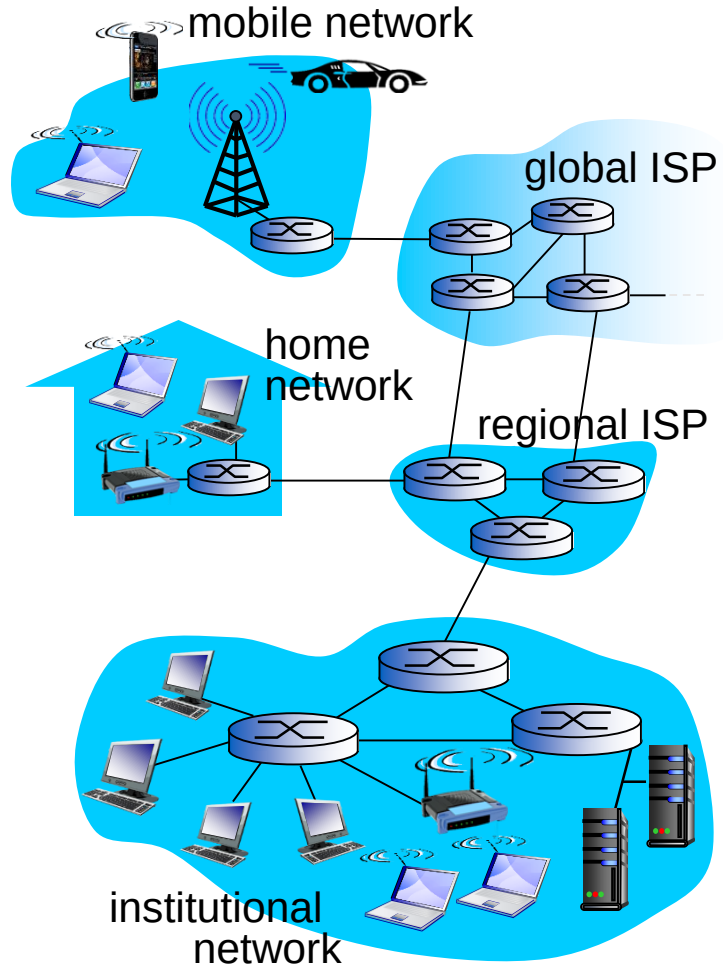
What is the Internet? "Nuts and bolts" View

- Millions of connected computing devices:
  - *hosts = end systems*
  - running *network apps*



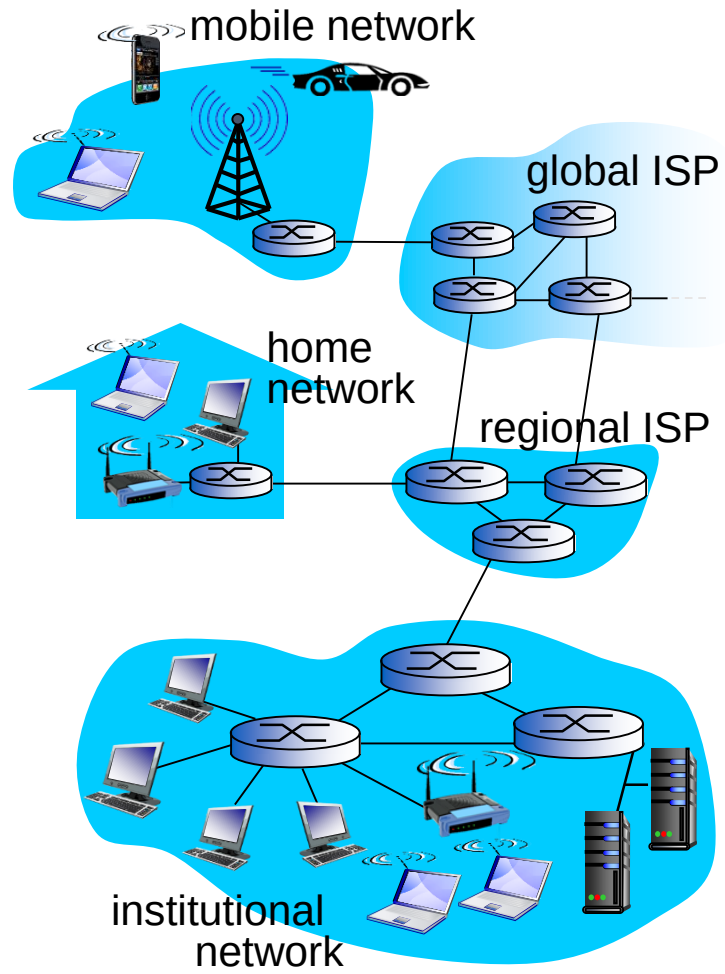
❖ *Communication links*
- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

❖ *Packet switches:* forward packets (chunks of data)
- *routers* and *switches*



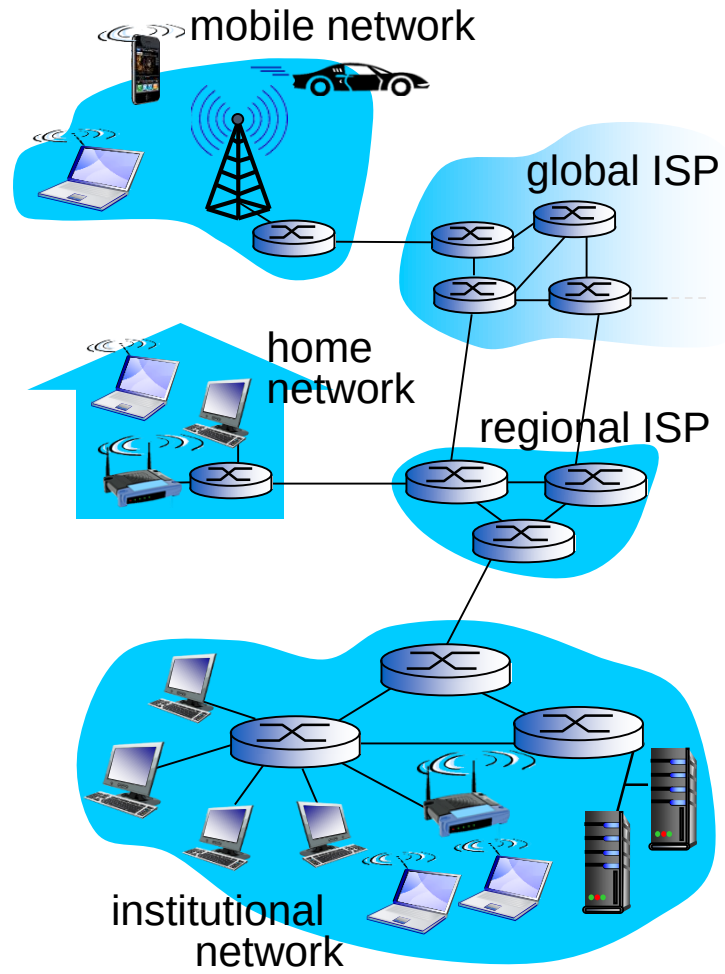
What is the Internet? “Nuts and bolts” View

- *Internet:* “network of networks”
  - Interconnected ISPs
- *Protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, 802.11
- *Internet standards*
  - IETF: Internet Engineering Task Force
  - RFC: Request for comments



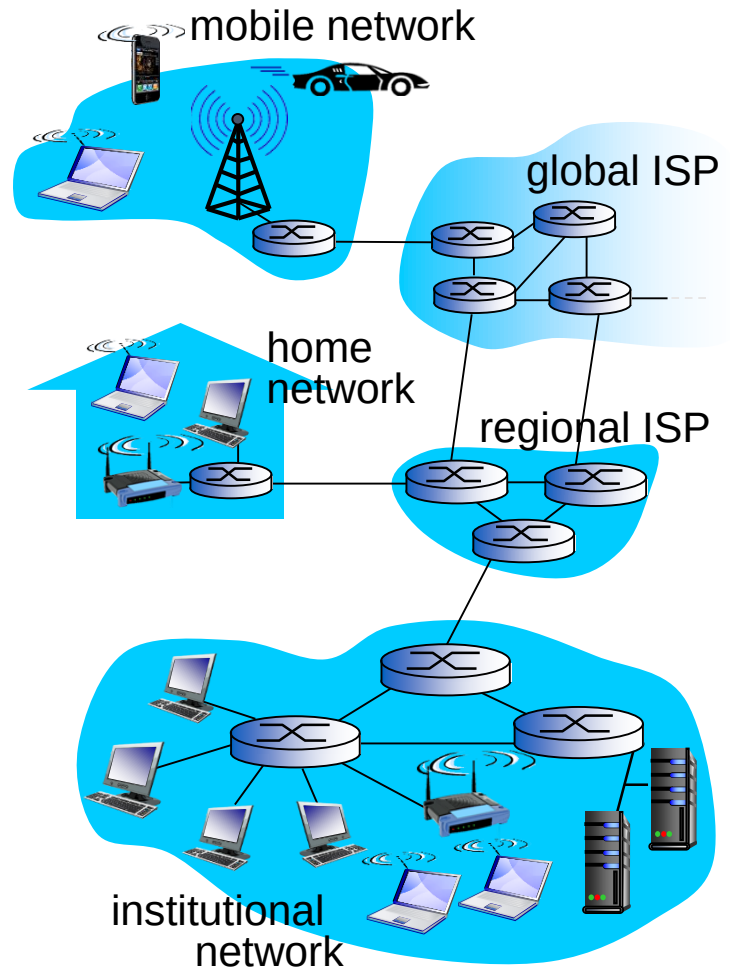
What is the Internet? A Service View

- *Infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *Provides programming interface to apps*
  - hooks that allow sending and receiving app programs to "connect" to Internet



A Closer Look at Network Structure

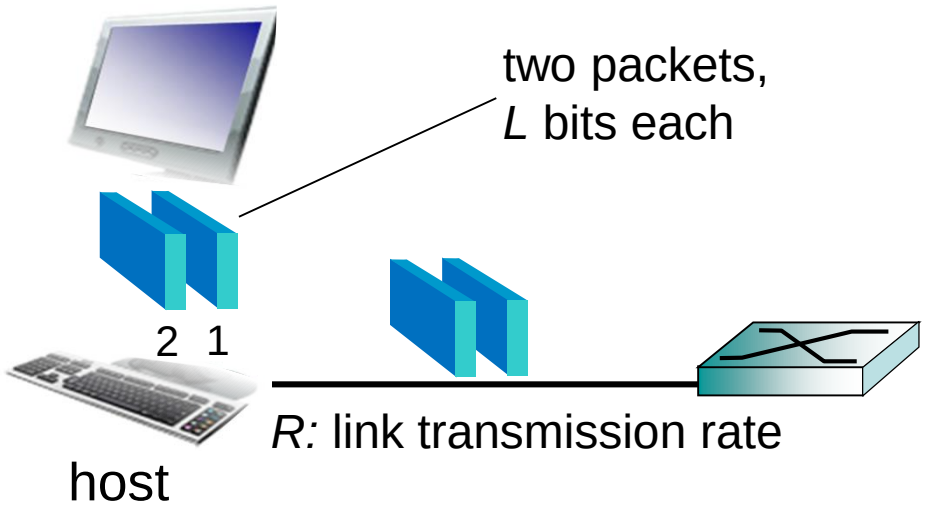
- *Network edge:*
  - hosts: clients and servers
  - servers often in data centers
- *Access networks, physical media:*
  - wired, wireless communication links
- *Network core:*
  - interconnected routers
  - network of networks



Host: Sends Packets of Data

Host sending function:

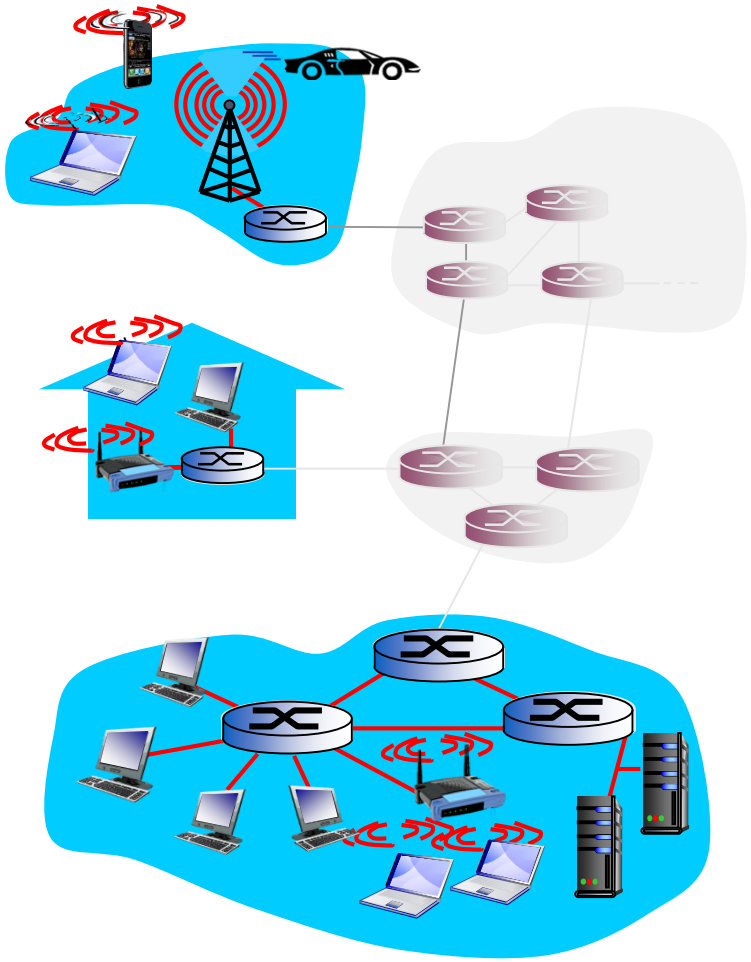
- takes application message
- breaks into smaller chunks, known as *packets*, of length *L* bits
- transmits packet into access network at *transmission rate R*
  - link transmission rate, aka link *capacity, aka link bandwidth*



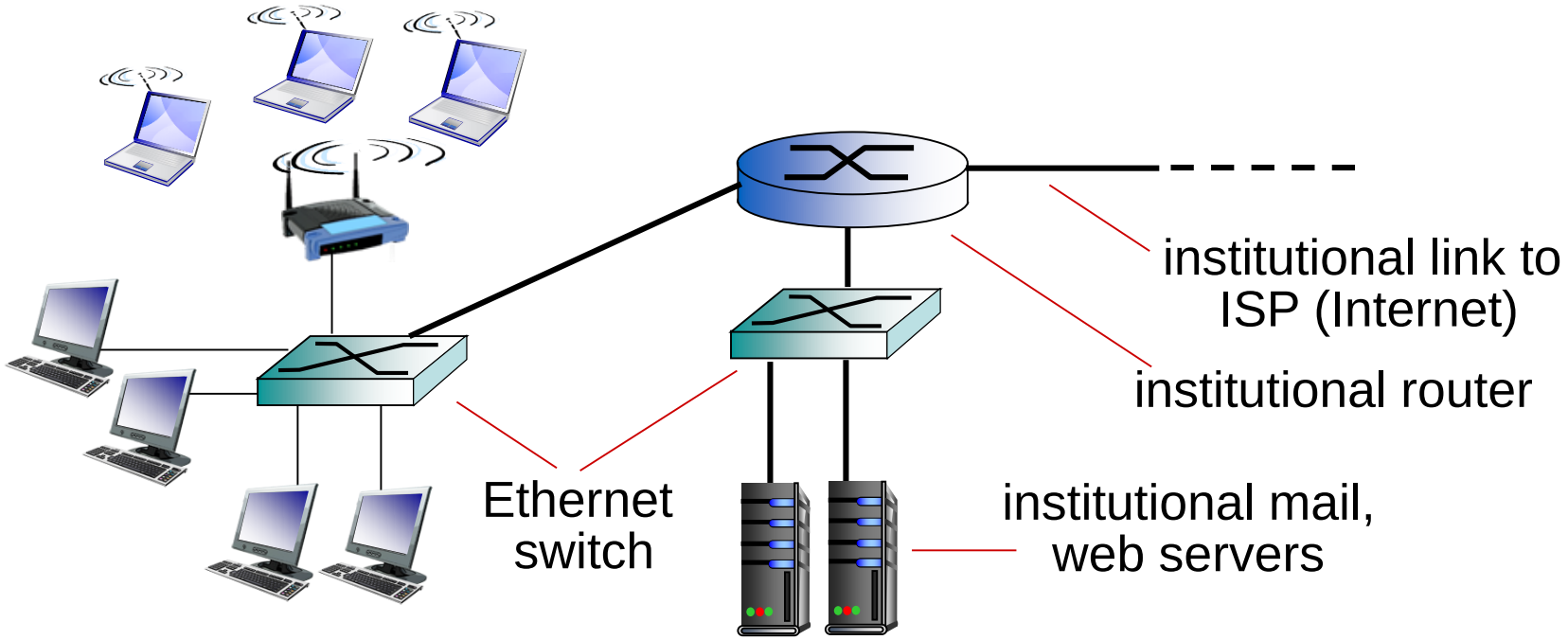
Access Networks

Connecting end systems to an edge router

- residential access nets
  - DSL
  - Cable network
- institutional access networks (school, company)
- mobile access networks



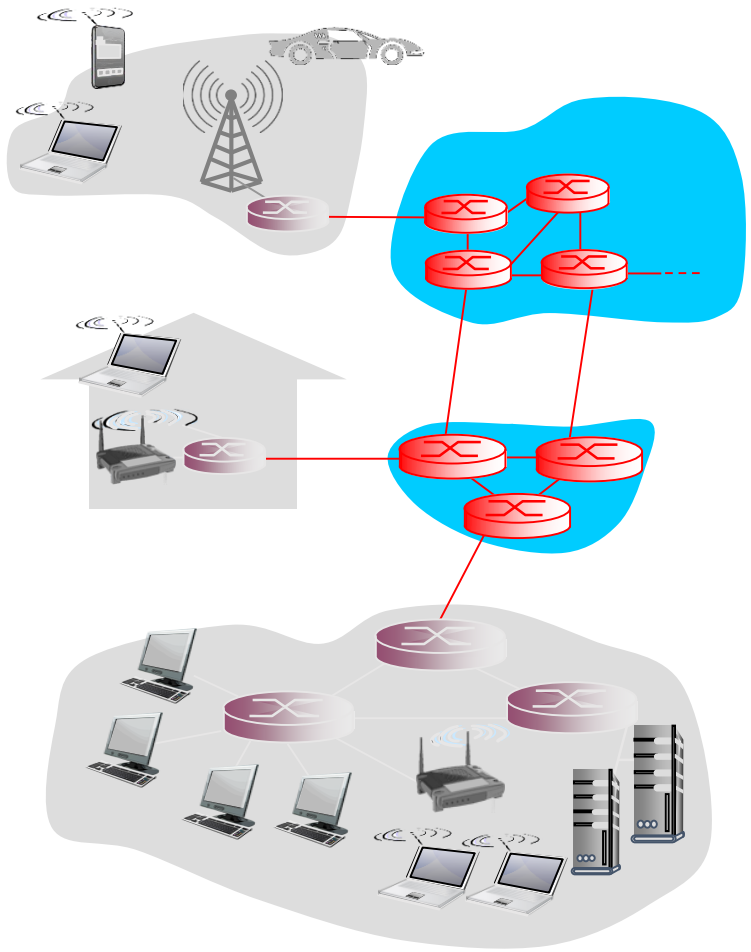
Enterprise Access Networks



- Typically used in companies, universities, etc
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- Today, end systems typically connect into Ethernet switch

The Network Core

- Mesh of interconnected routers
- Two approaches:
  - Circuit switching
  - Packet switching

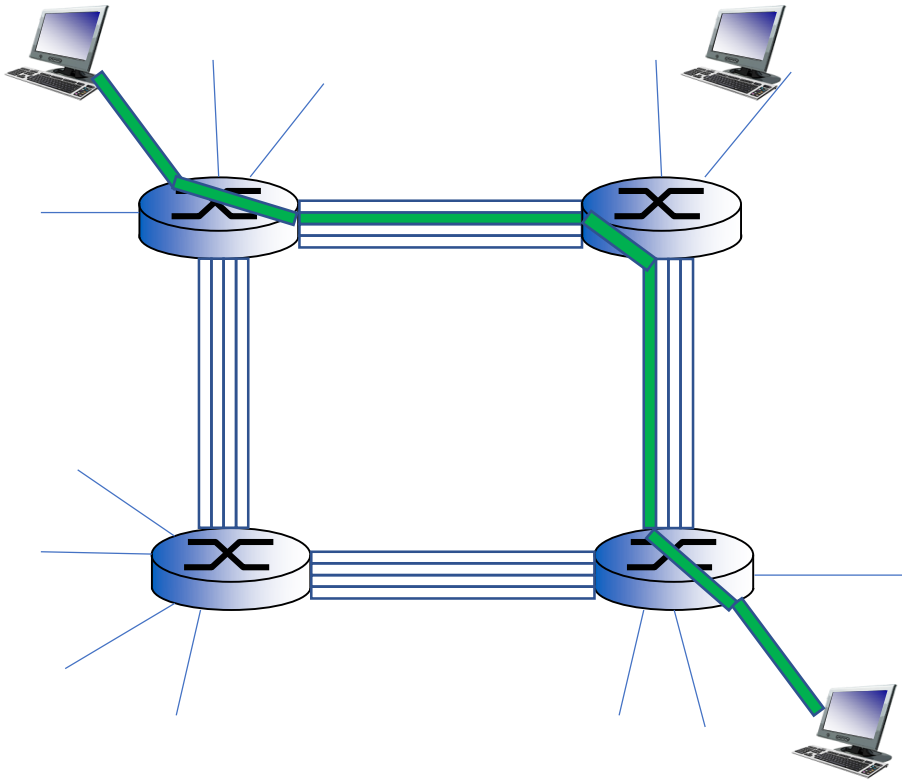


Circuit Switching

Circuit Switching

End-to-end resources allocated to, reserved for "call" between source & destination:

- In diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- *No sharing:* circuit segment idle if not used by call
- Commonly used in traditional telephone networks



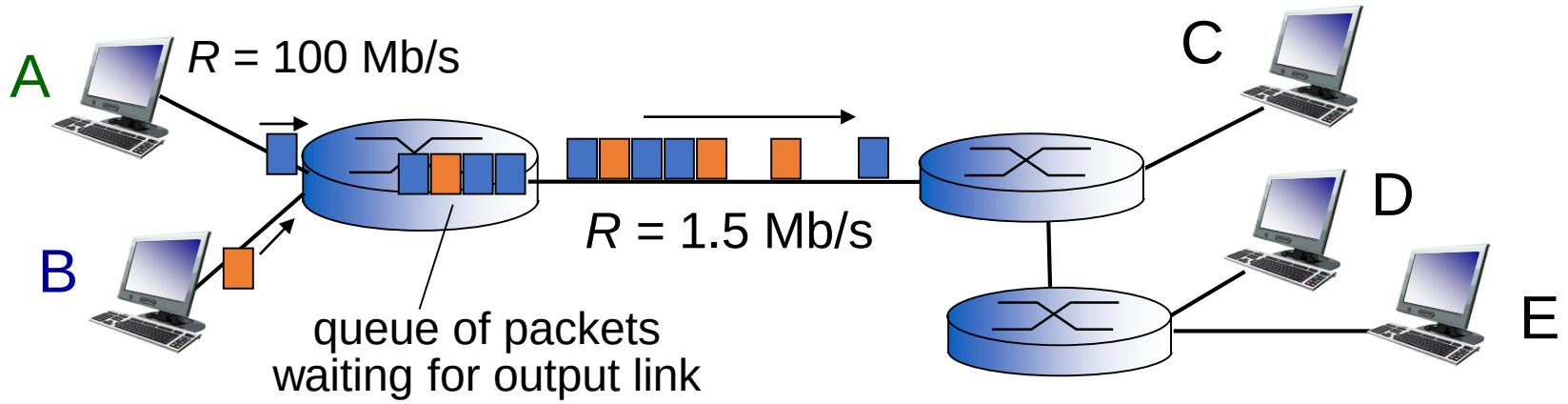
Packet Switching

- Packet Switching: Hosts break application-layer messages into packets
  - Forward packets from one router to the next, across links on path from source to destination
  - Each packet is transmitted at full link capacity (no reservation)
- The header of each packet carries necessary information
  - Routers examine the header and make forwarding decisions

Packet Switching: Store-and-forward

- No end-to-end connection is established

  → entire packet must arrive at router before it can be transmitted on next link (i.e., *store-and-forward*)



As a result, packet switching may result in queuing delay and packet loss:

- If arrival rate (in bits) to link exceeds transmission rate of link for a period:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

Circuit Switching vs Packet Switching

- Packet switching
  - Good for bursty traffic
  - Resource sharing → more users
  - Simple implementation without call setup
  - No guaranteed bandwidth
    - it may result in congestion (packet delay and loss)
- Circuit switching
  - Good for constant data rates
  - Guaranteed bandwidth
  - Circuit establishment and maintenance is expensive

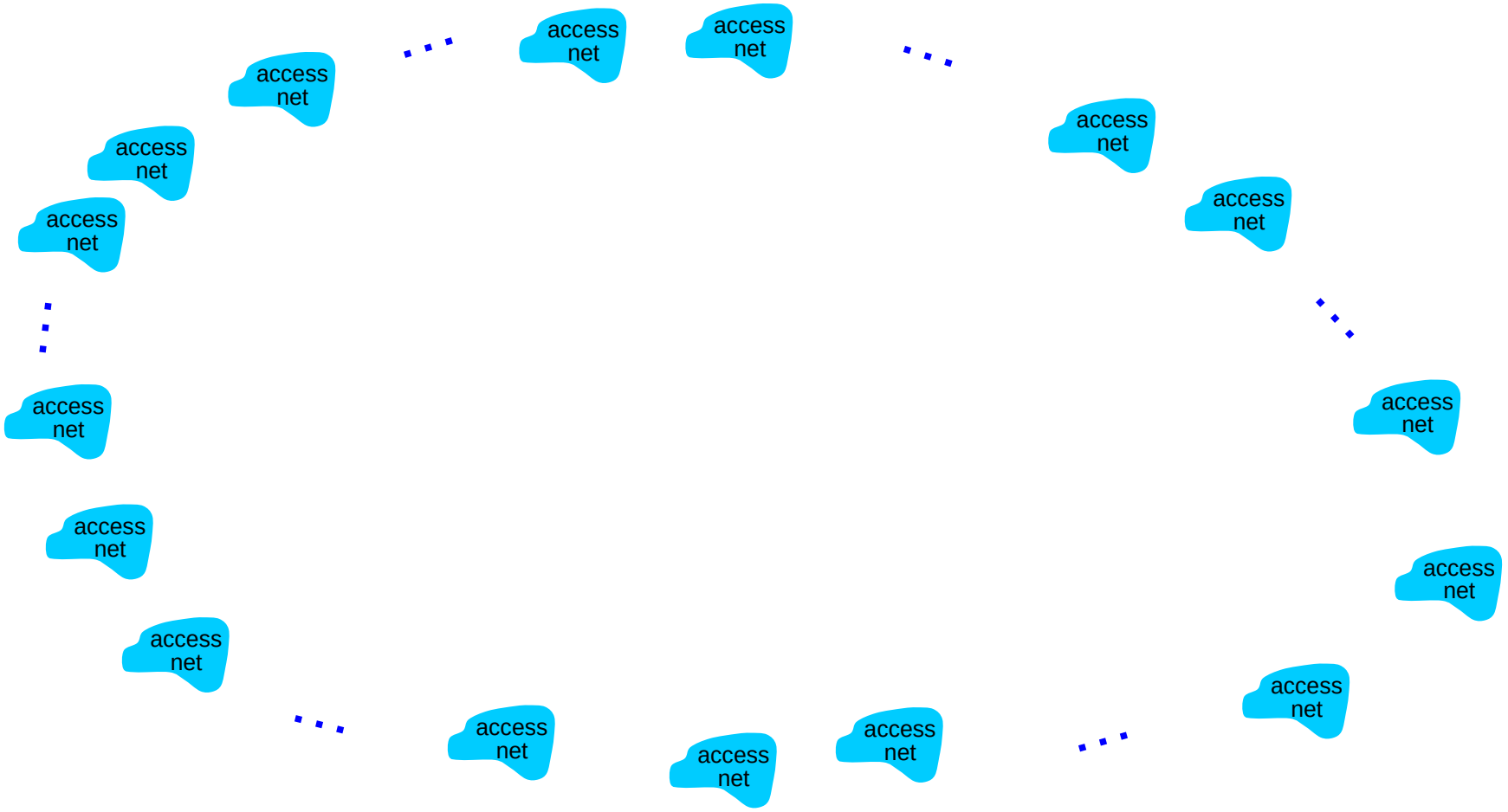
Which one is used in today's Internet?

Internet Structure: Network of networks!

- End systems connect to Internet via access ISPs (Internet Service Providers)
  - Residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - So that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - Evolution was driven by economics and national policies

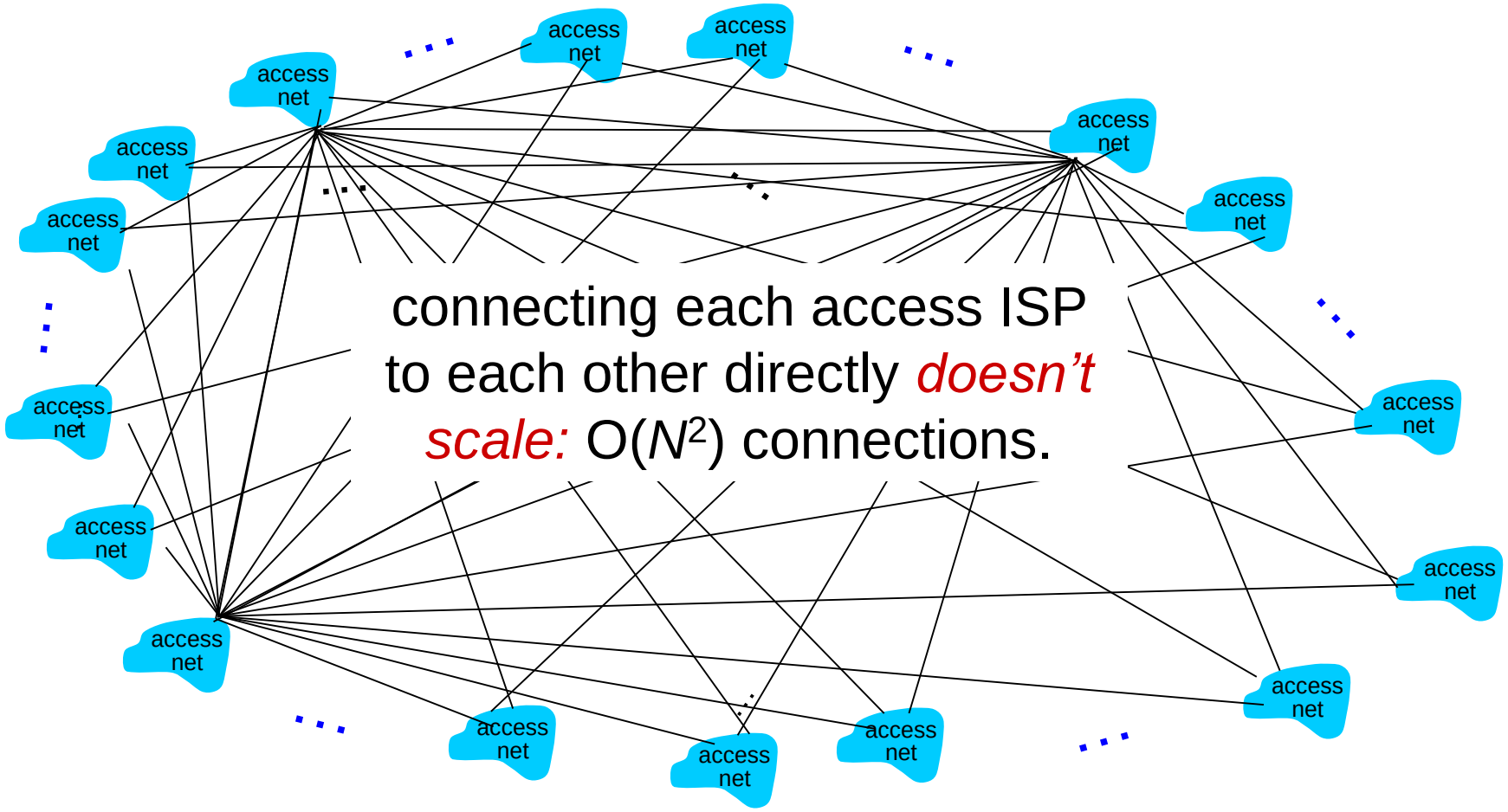
Internet Structure: Network of networks

- given *millions* of access ISPs, how to connect them together?



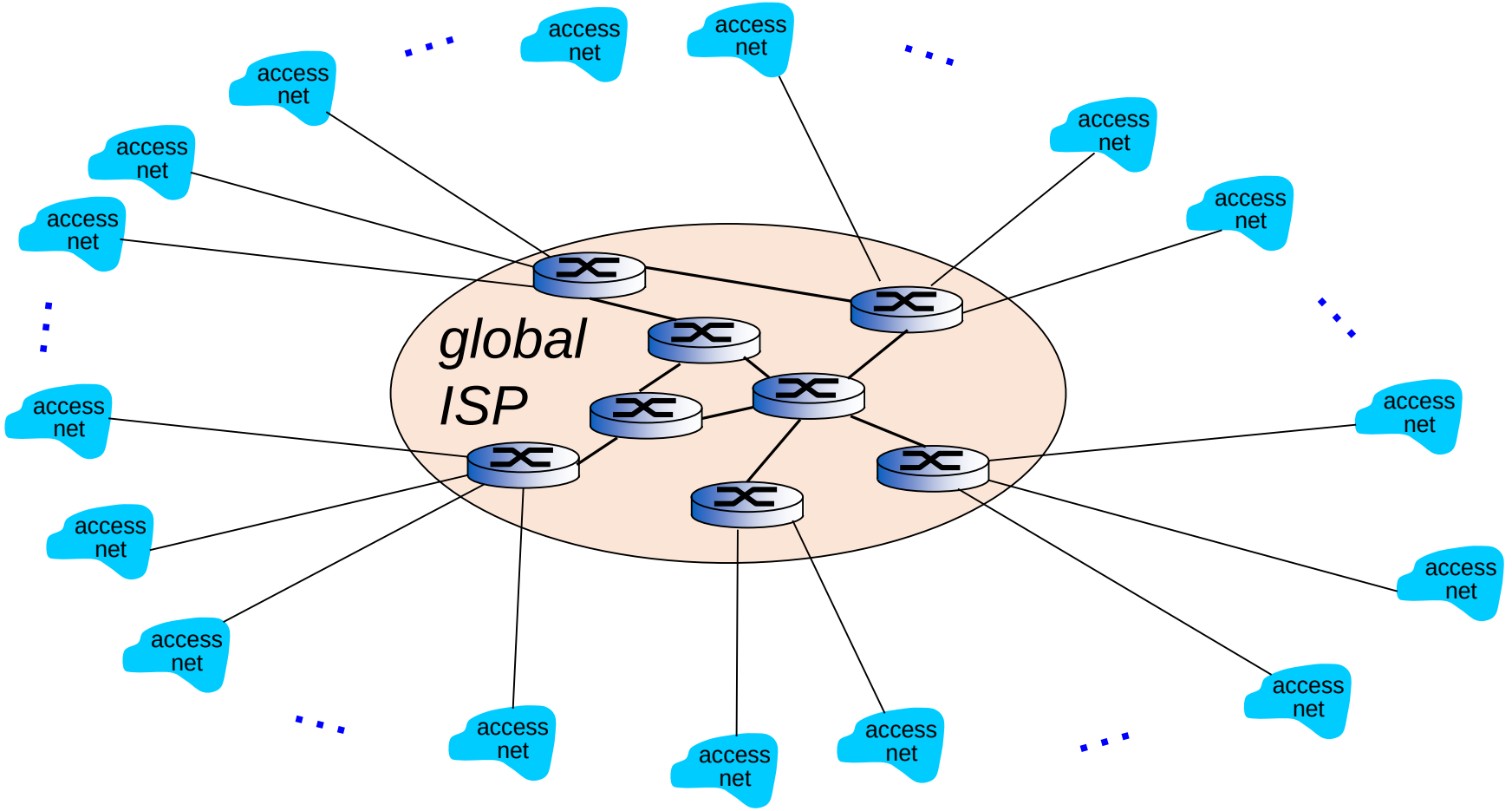
Internet Structure: Network of networks

Option: connect each access ISP to every other access ISP?



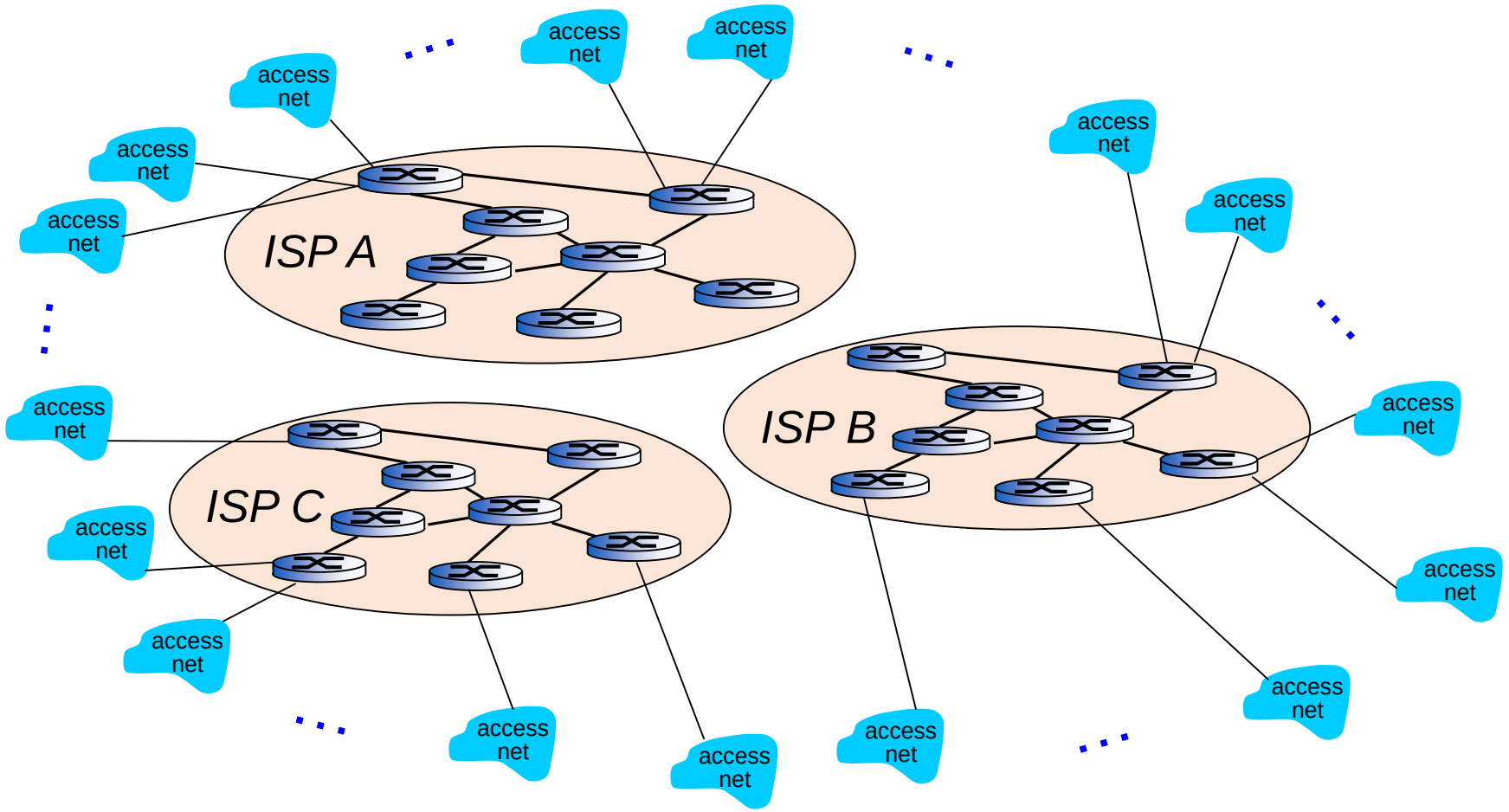
Internet Structure: Network of networks

- *Option: connect each access ISP to a global transit ISP?*
  - *Customer and provider ISPs have economic agreement.*



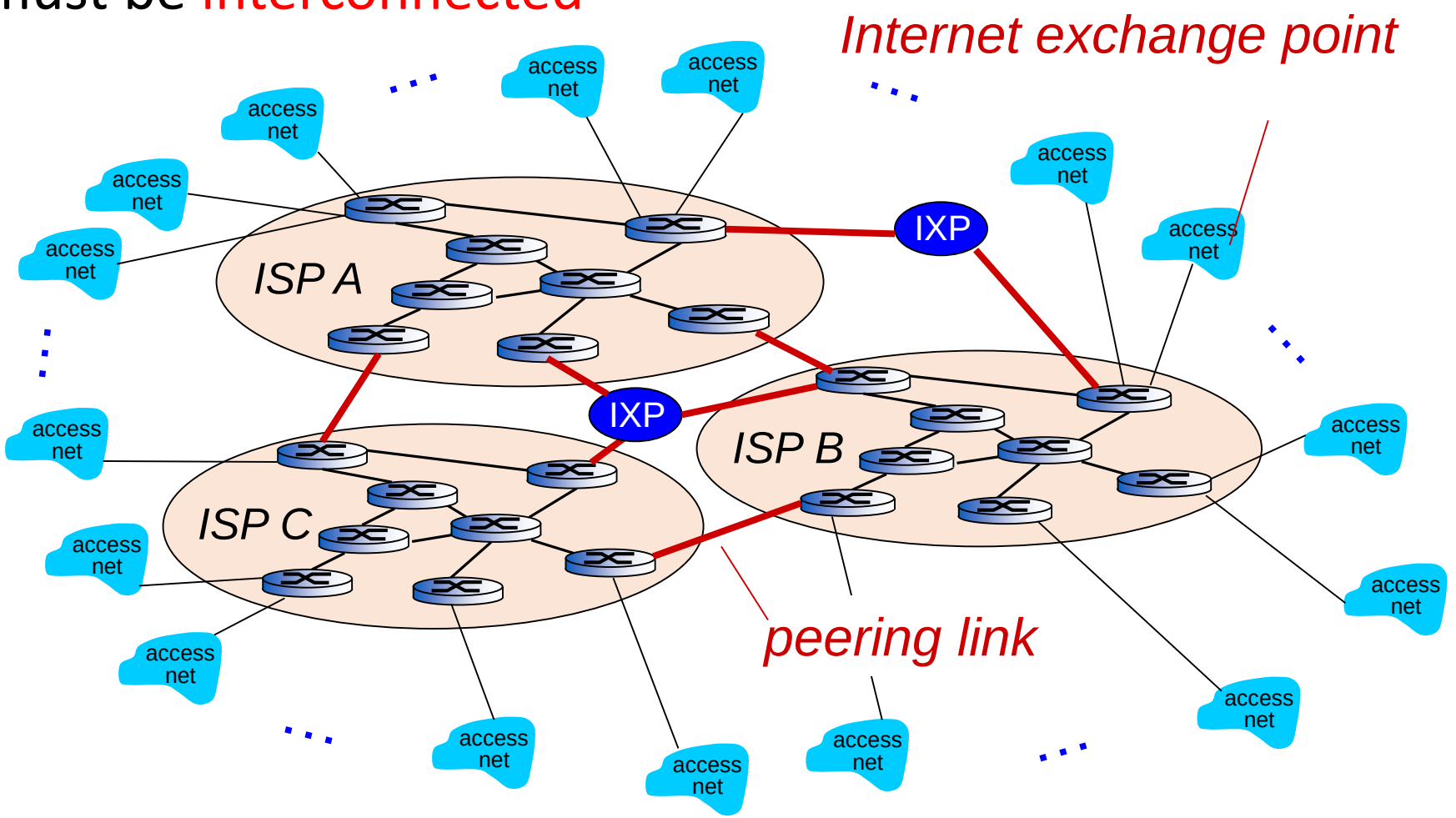
Internet Structure: Network of networks

- But if one global ISP is viable business, there will be competitors ….



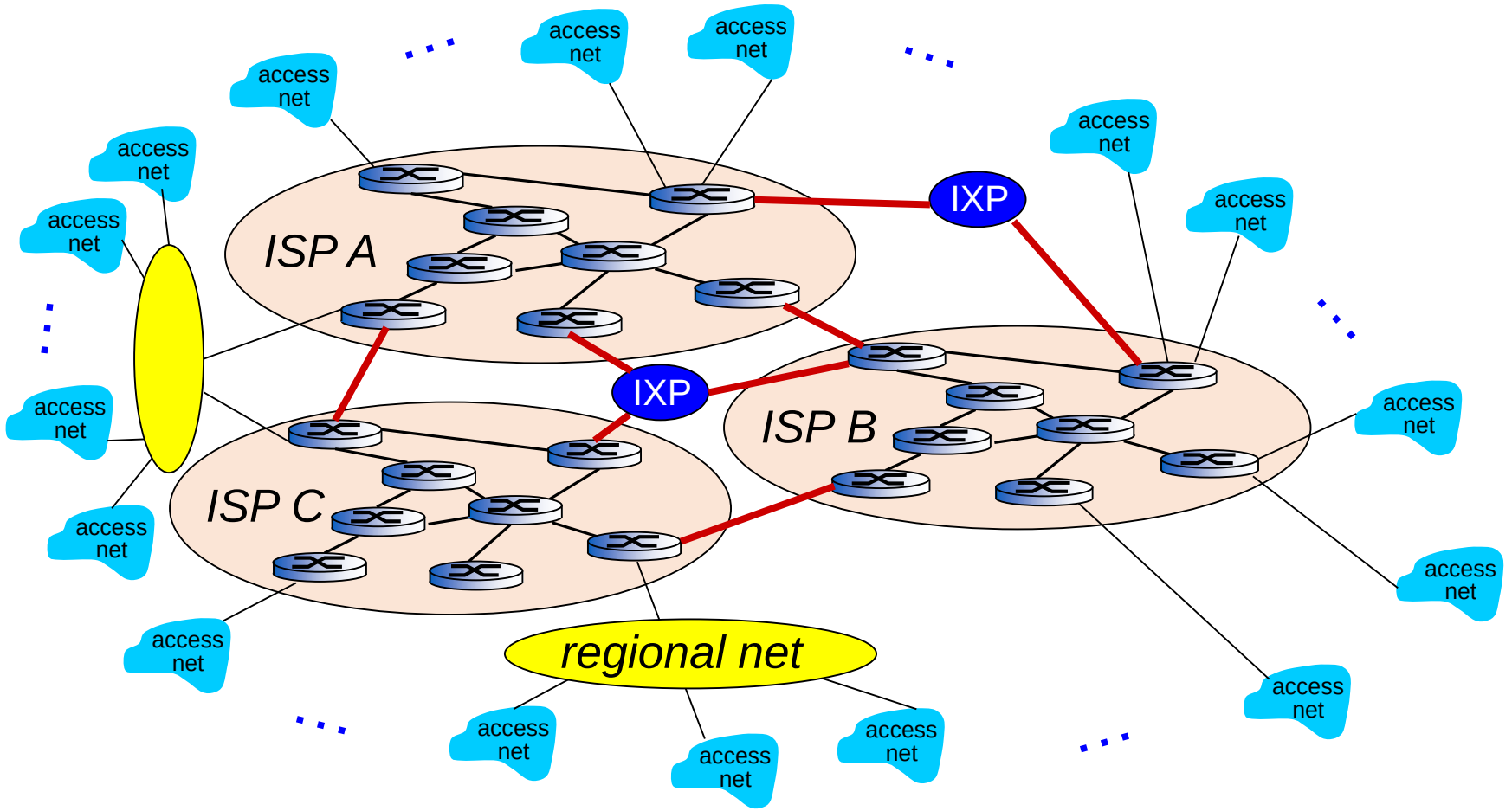
Internet Structure: Network of networks

- But if one global ISP is viable business, there will be competitors …. which must be interconnected



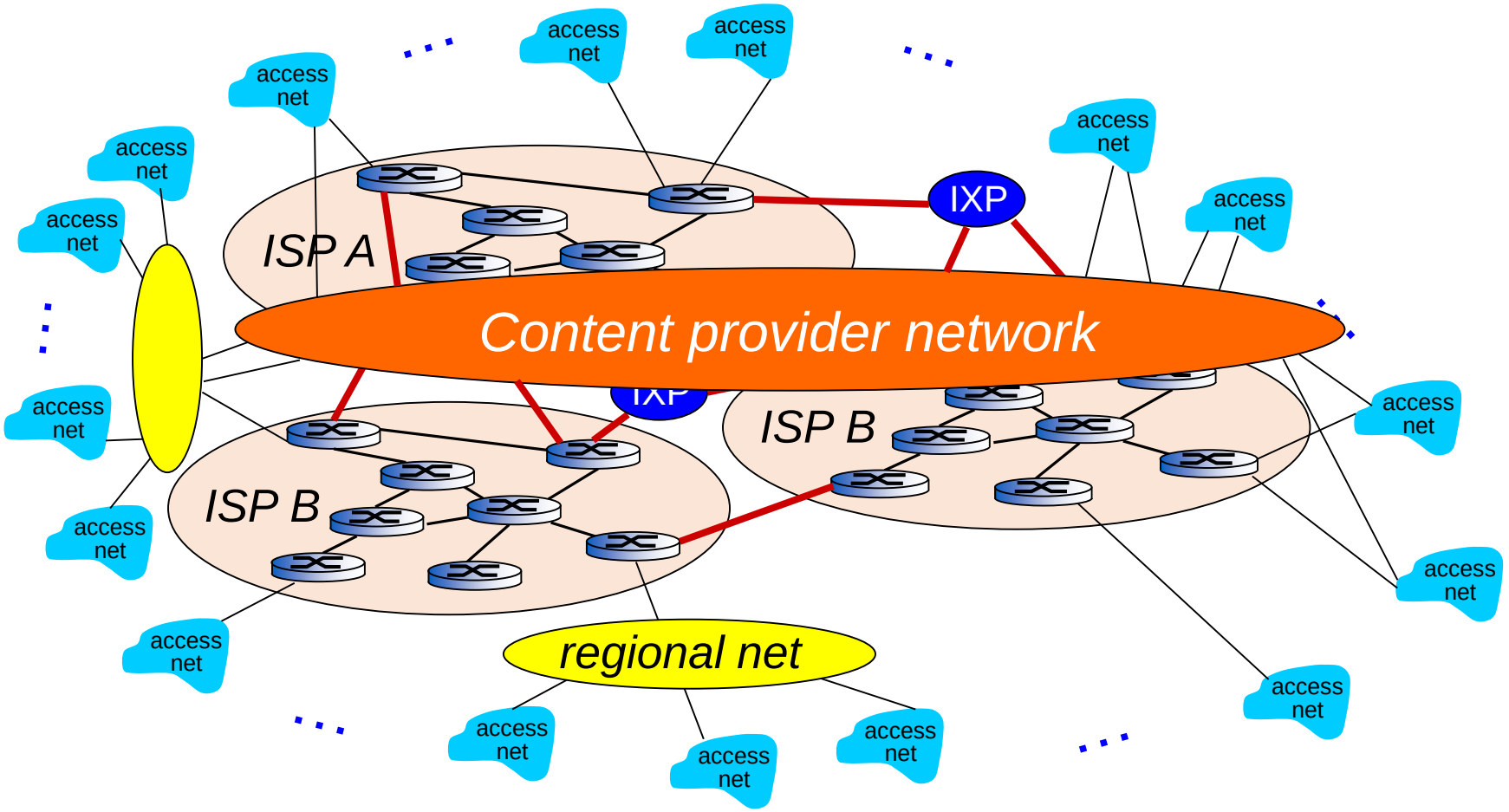
Internet Structure: Network of networks

- … and regional networks may arise to connect access nets to ISPS

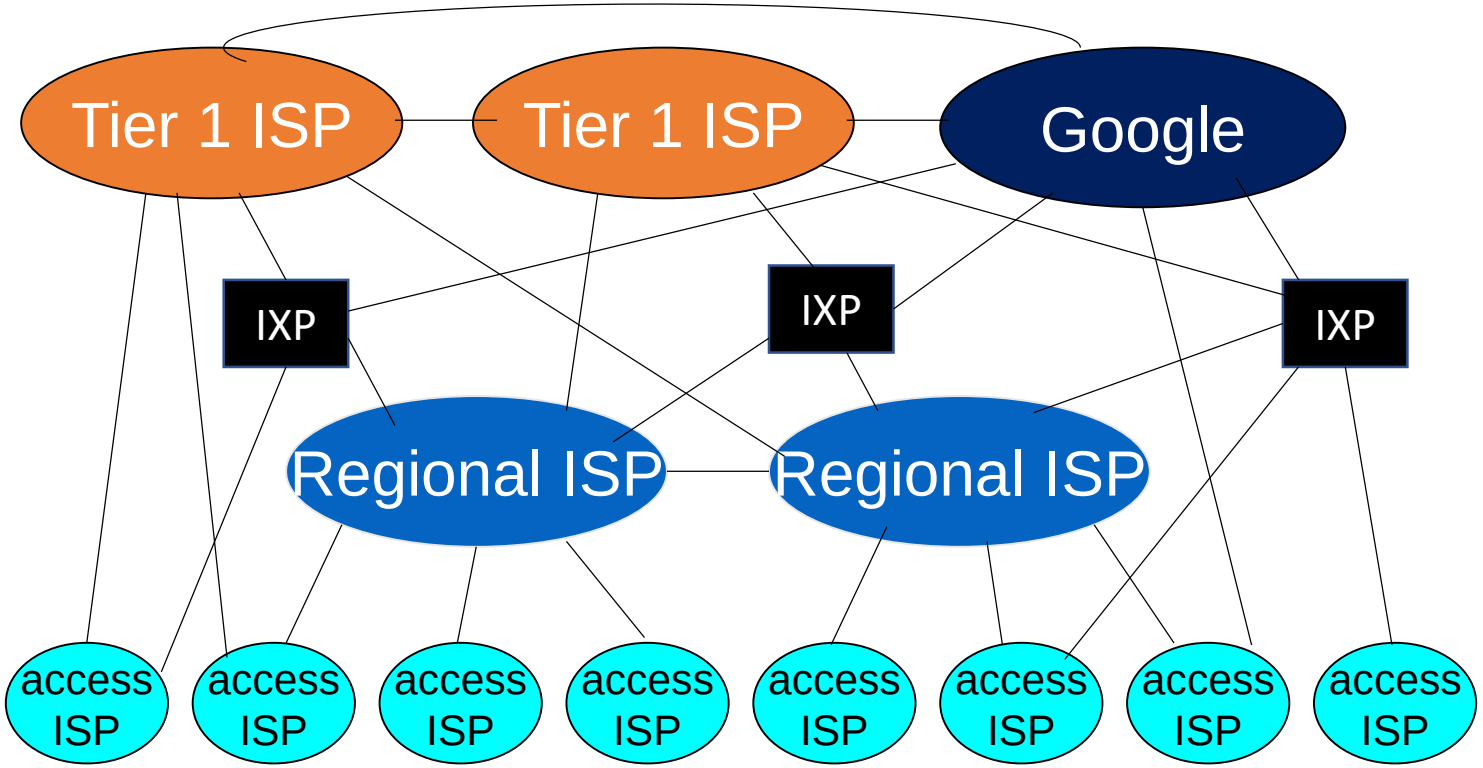


Internet Structure: Network of networks

- … and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Internet Structure: Network of networks



- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g, Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

Network Protocols

What is a Protocol?

Human protocols:

- "what's the time?"
- "I have a question"
- introductions

- specific msgs sent
- specific actions taken when msgs received, or other events

Network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

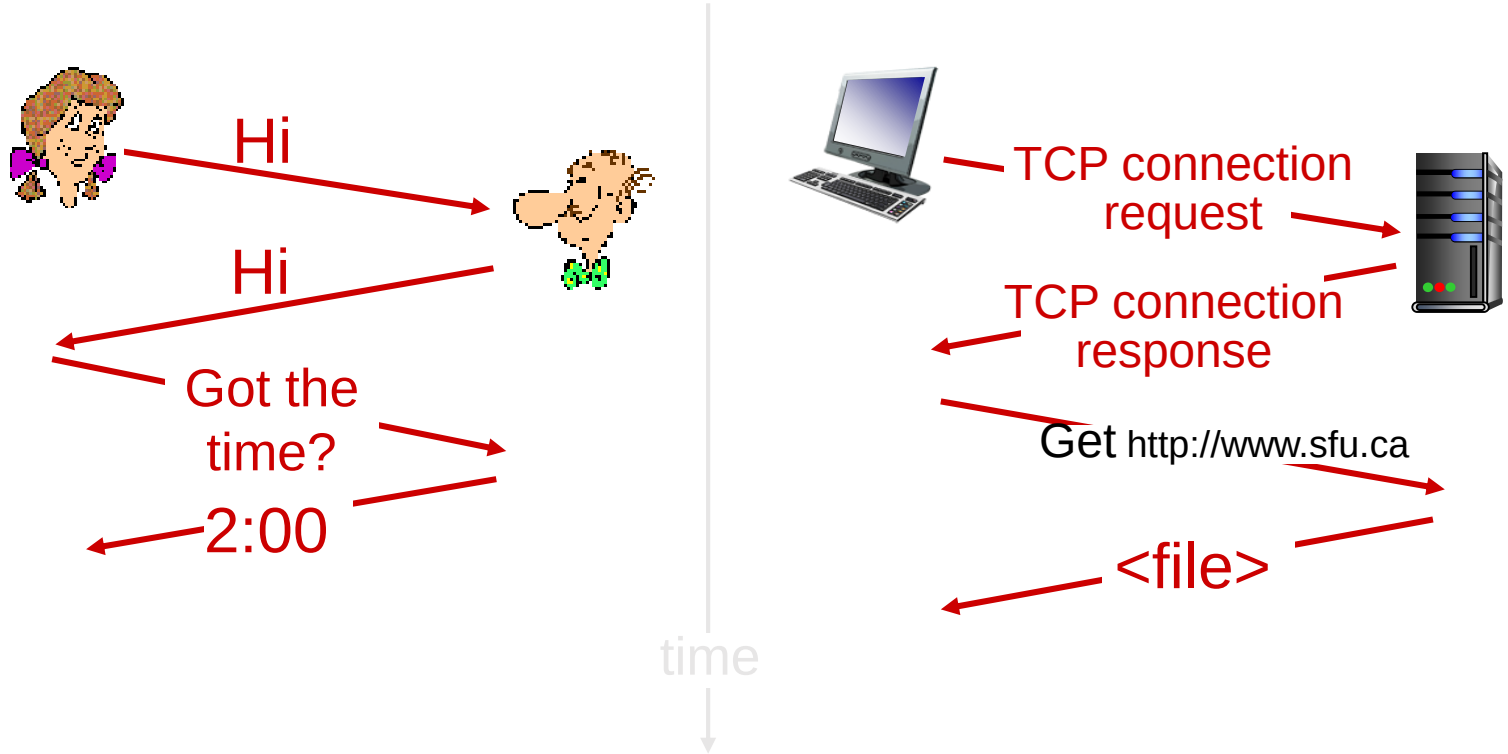
Protocols define:

(1) format, order of msgs sent and received among network entities, and

(2) actions taken on msg transmission, receipt

What is a Protocol?

a human protocol and a computer network protocol:

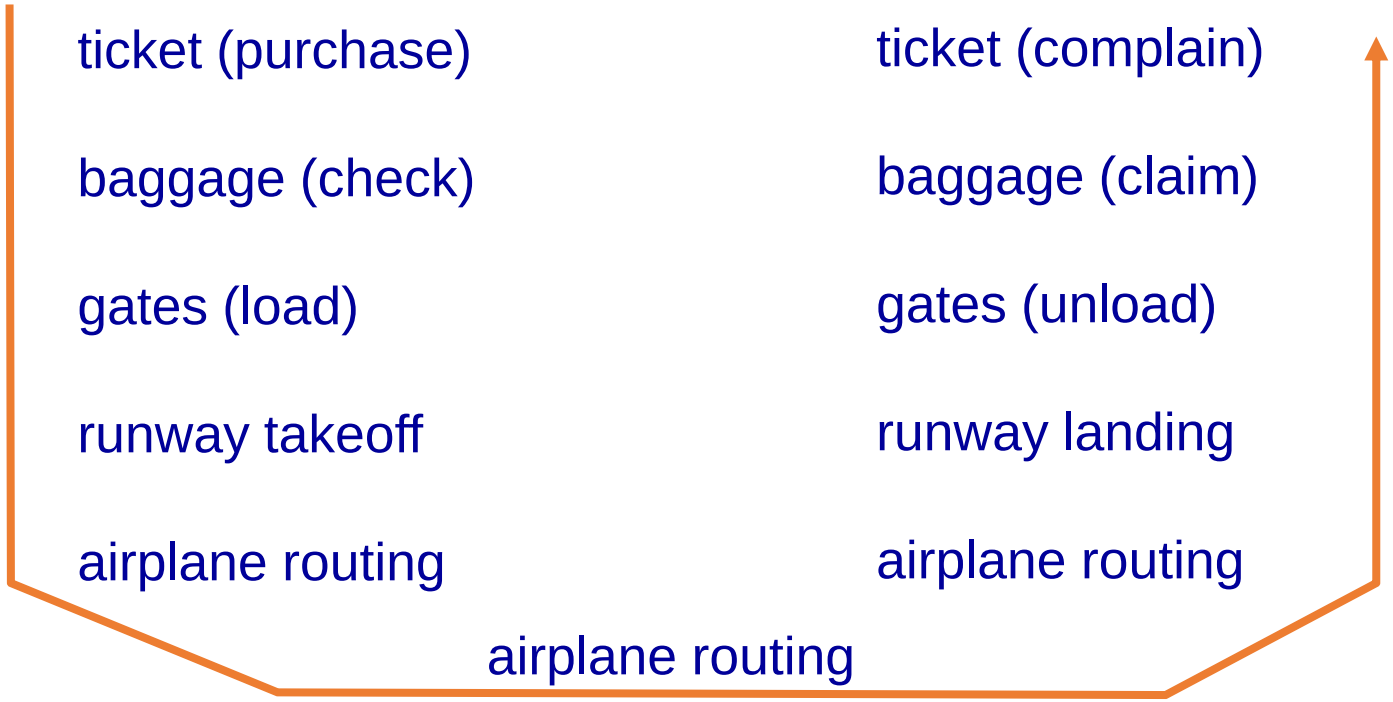


Protocol Layers

- *Networks are complex, with many "components":*
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

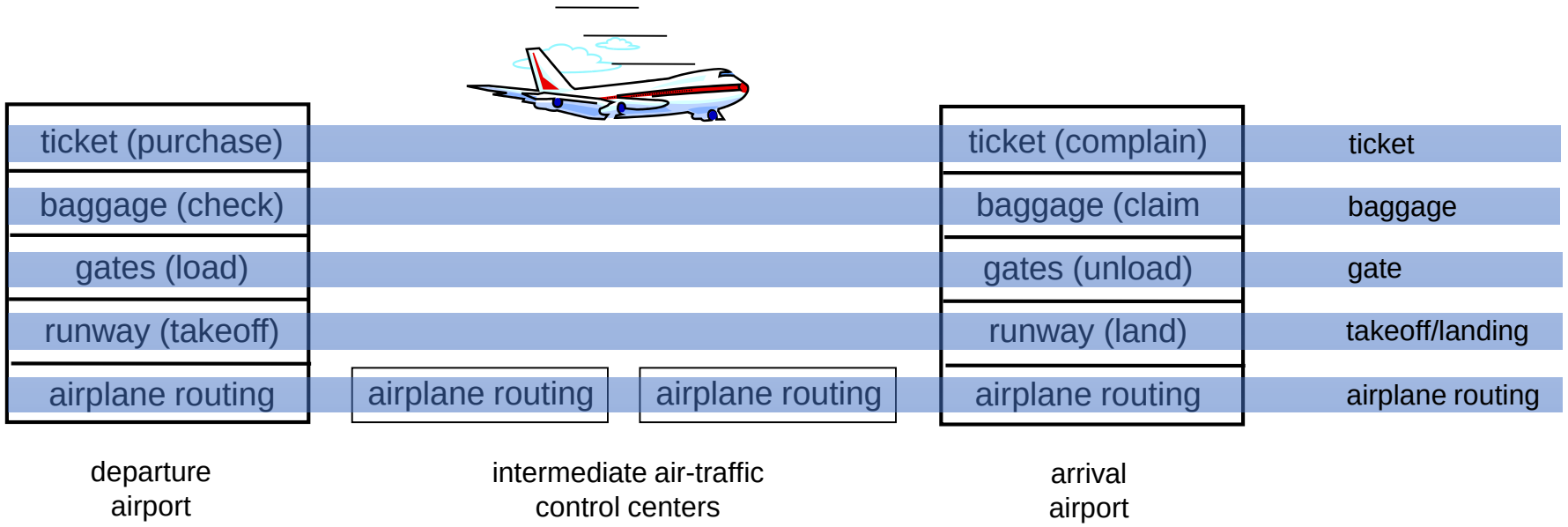
- How can we organize this structure?

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

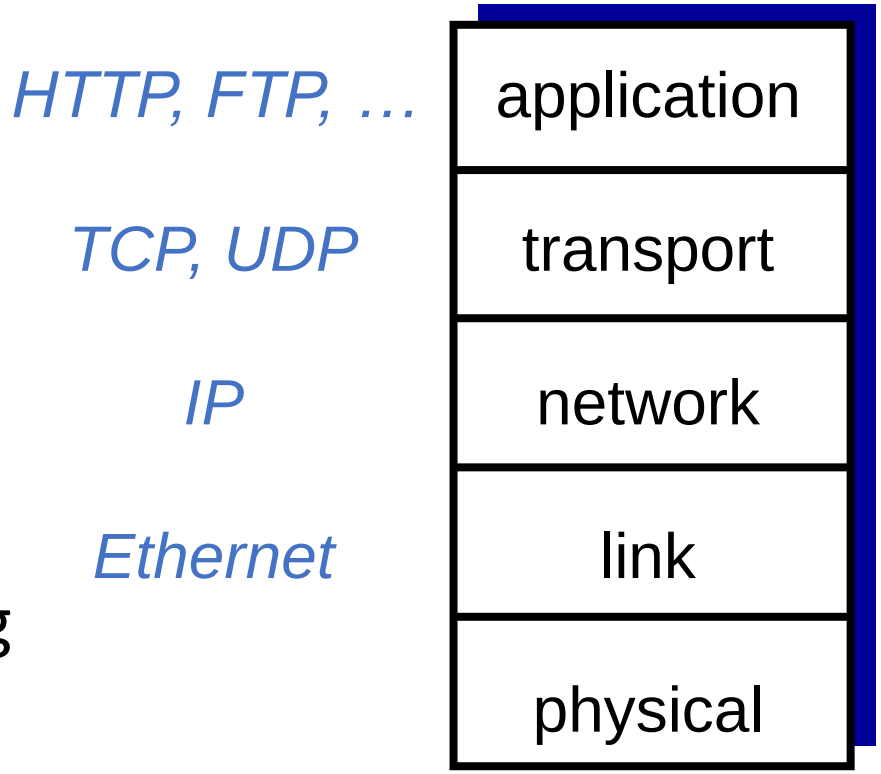
Why layering?

- explicit structure allows identification, relationship of complex system components
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer service transparent to rest of system
  - e.g., change in gate procedure does not affect rest of system

- Two layering models:
  - TCP/IP protocol suite
  - ISO/OSI reference model

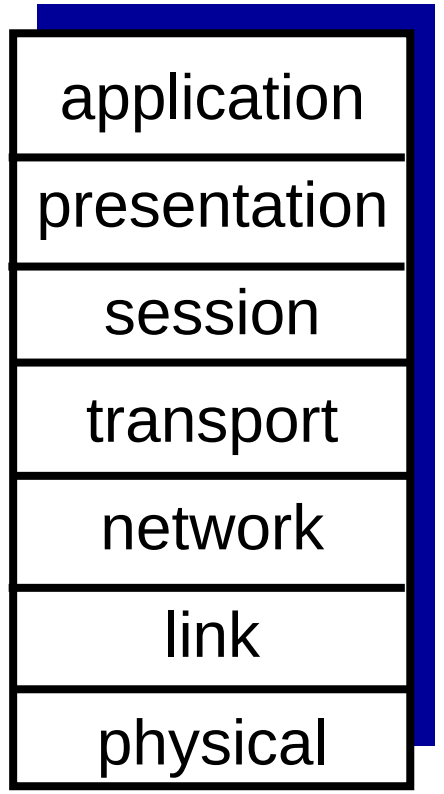
TCP/IP Protocol Suite

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-to-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits “on the wire”

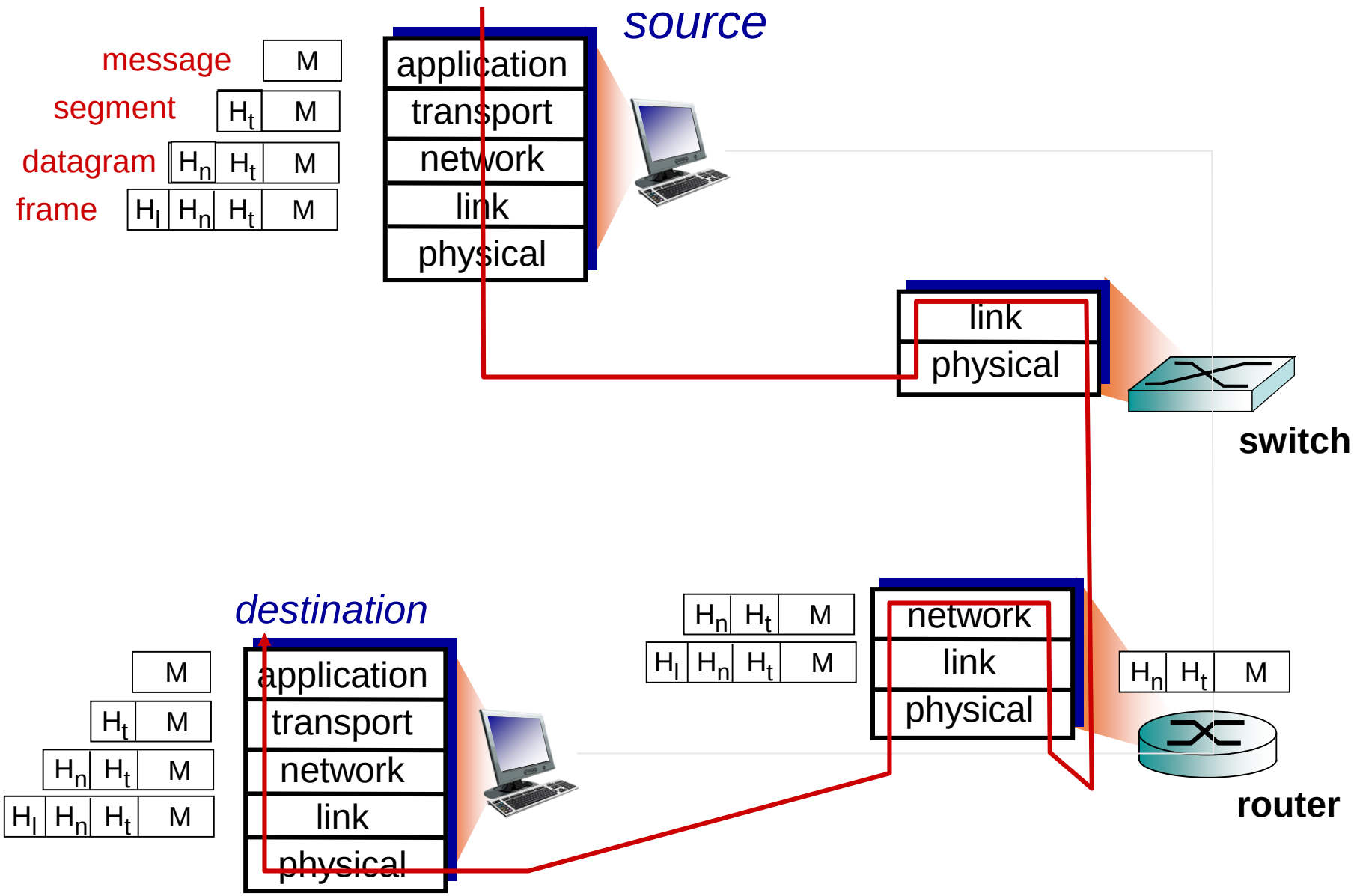


ISO/OSI Reference Model

- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- TCP/IP stack "missing" these layers!
  - these services, *if needed,* must be implemented in application

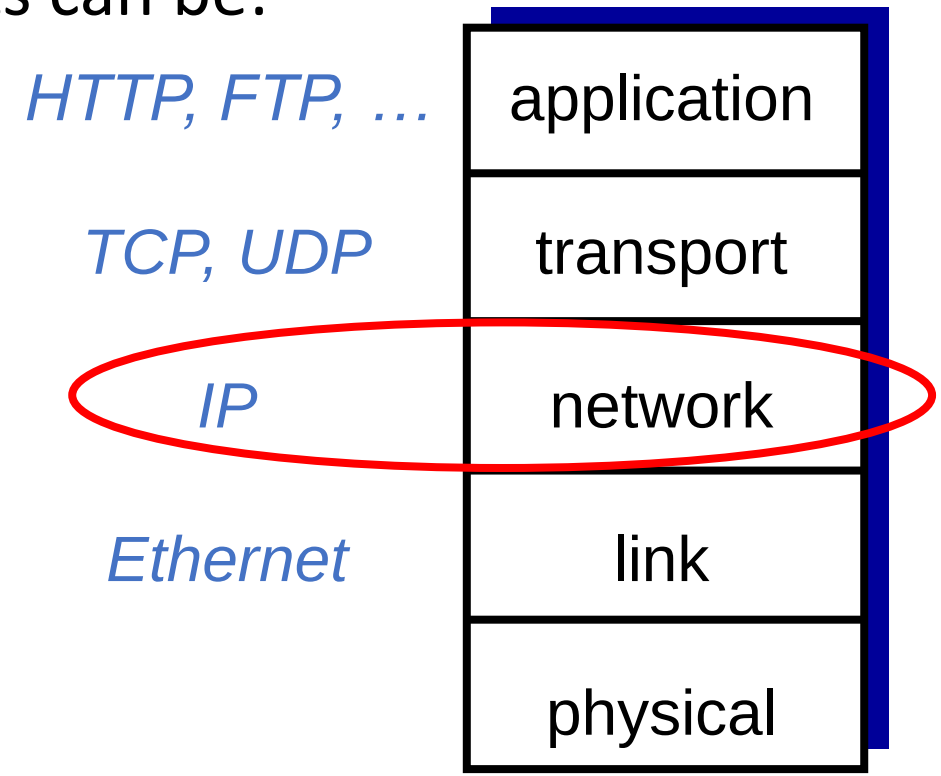


Encapsulation



Network Layer: Internet Protocol (IP)

- IP is a connectionless protocol, and provides no end-to-end control
  - A datagram service
- Each packet is treated separately, so packets can be:
  - received out of order
  - dropped
  - duplicated



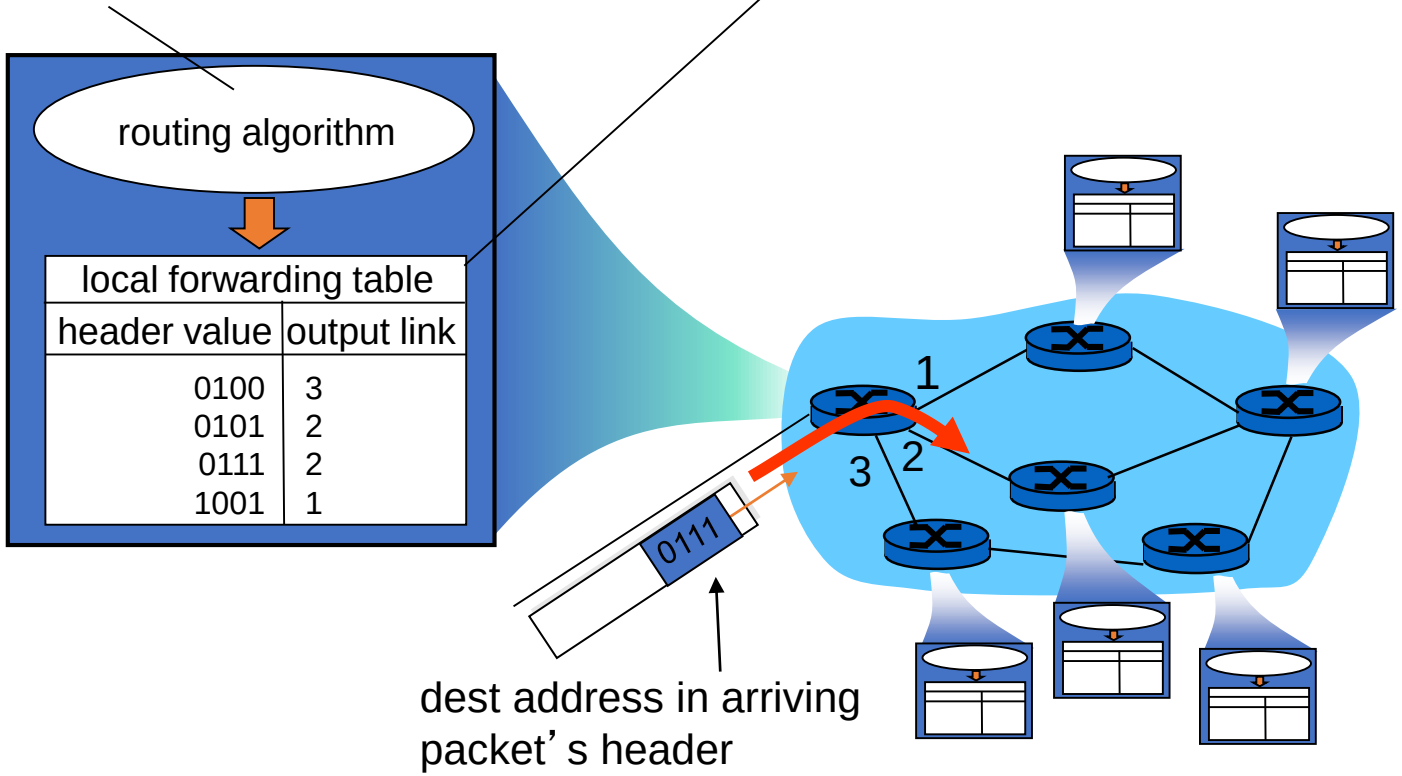
Network Layer: Internet Protocol (IP)

- Recall Packet switching at the network core.

routing: determines source-destination route taken by packets

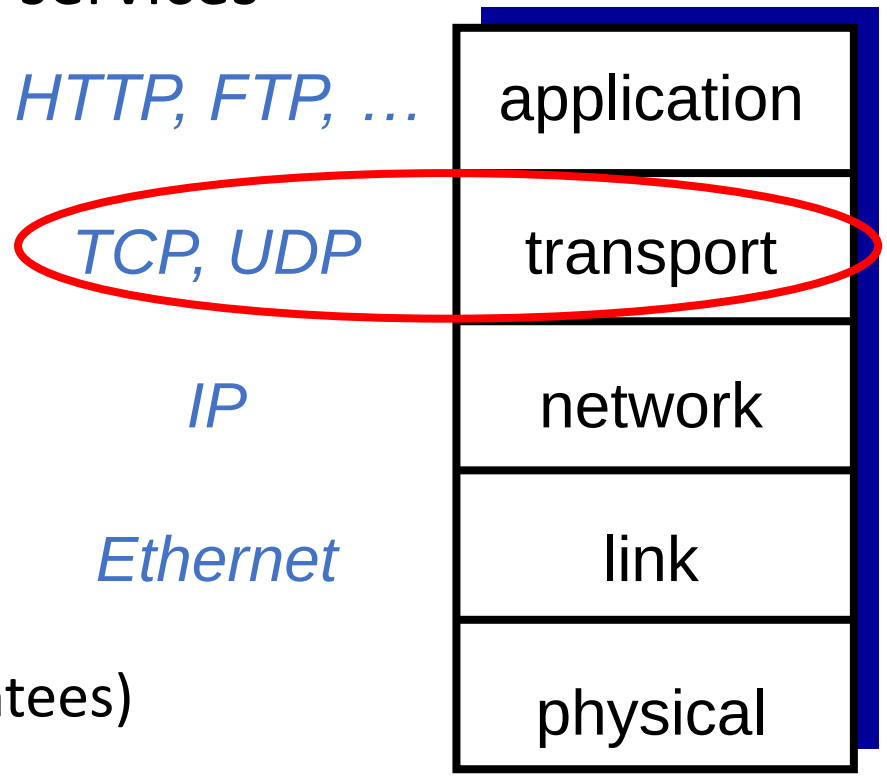
- *routing algorithms*

forwarding: move packets from router's input to appropriate router output



Transport Layer

- Provides process-to-process communication services
- User Datagram Protocol (UDP)
  - Connectionless protocol
  - No delivery guarantees
  - Low overhead
- Transmission Control Protocol (TCP)
  - Connection-oriented
  - Reliable transmission (but no bandwidth guarantees)
  - More overheads



Basics of Routing

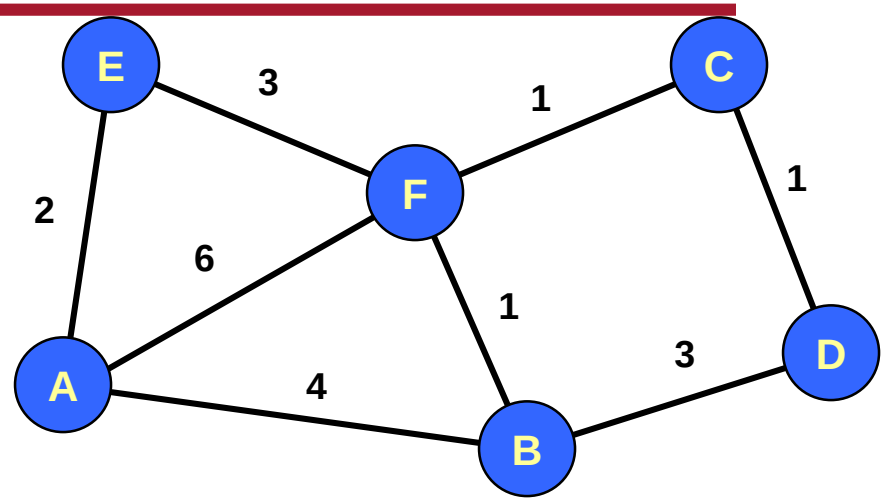
Edited from Vyas Sekar's slides

Intra-domain Routing

- Routing within the network
  - No hierarchy is needed
- The connectivity layer in the ISP network
- Examples: Distance-Vector and Link State routing

Distance-Vector Routing

- Idea
  - At any time, have cost/next hop of best-known path to destination
  - cost = ∞ when no path known
- Initially
  - Only have entries for directly connected nodes



Initial Table for A		
Dest	Cost	Next Hop
A	0	A
B	4	B
C	∞	–
D	∞	–
E	2	E
F	6	F

DV Routing Update

- Update(x,y,z)

  d ← c(x,z) + d(z,y) # Cost of path from x to y with first hop z

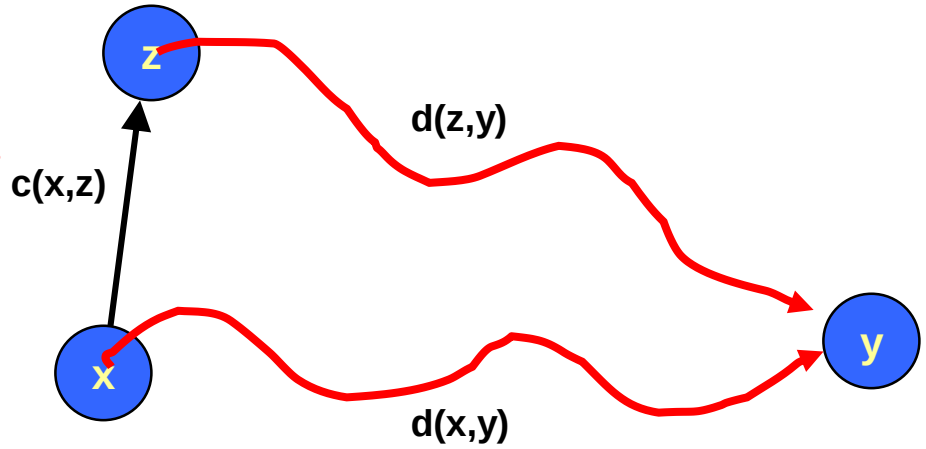
  if d < d(x,y)

    # Found better path

    return d,z # Updated cost / next hop

  else

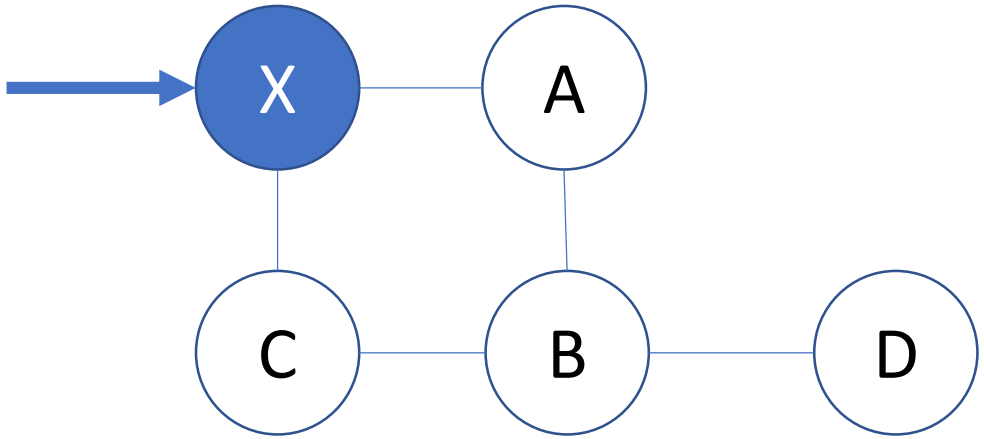
    return d(x,y), nexthop(x,y) # Existing cost / next hop



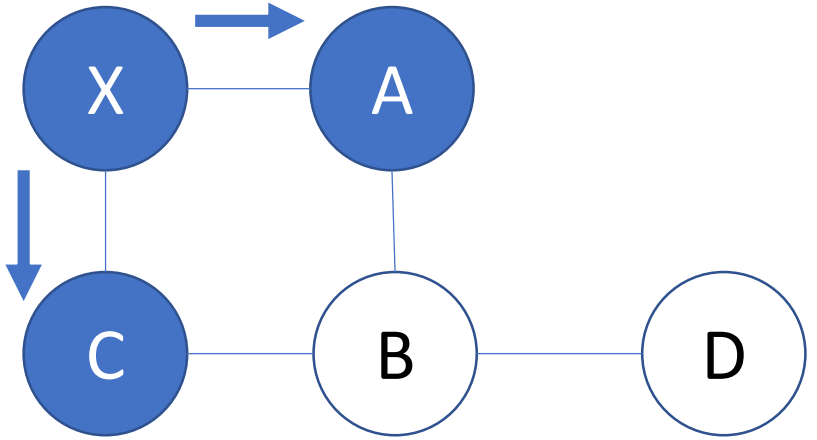
Link State Protocol

- Every node gets complete copy of graph
  - Every node “floods” network with data about its outgoing links
- Every node computes routes to every other node
  - Using single-source, shortest-path algorithm (Example?)
- Process performed whenever needed
  - When connections die / reappear

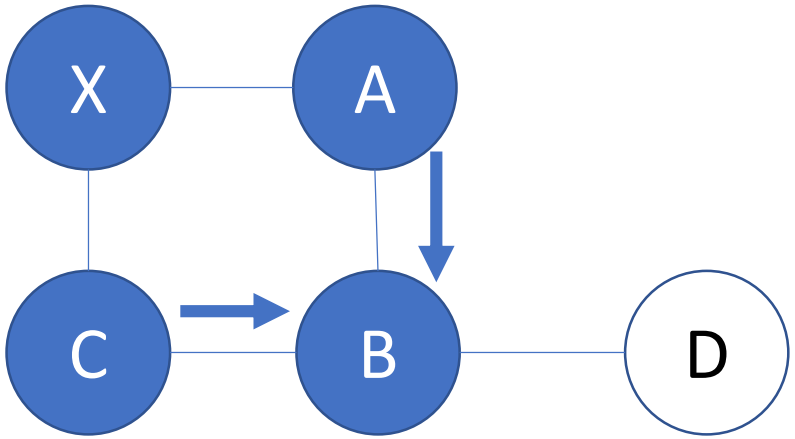
LS Flooding



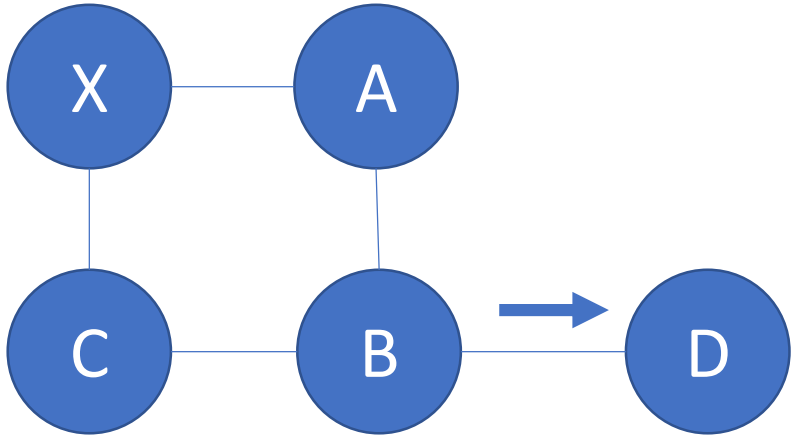
(1)



(2)



(3)



(4)

Inter-domain Routing

- "Flat" routing not suited for the Internet
  - Scalability (as the network size increases)
    - Space complexity → Each node cannot be expected to store routes to every destination (or destination network)
    - Convergence times increase
    - Communication → Total message count increases
  - Administrative autonomy
    - Each internetwork may want to run its network independently
      - E.g., hide topology information from competitors
- Solution: Hierarchy via autonomous systems

Today's Internet

- Uses hierarchy of AS's
- Each AS:
  - A set of routers under a single technical administration
  - Use an *interior gateway protocol (IGP)* and common metrics to route packets within the AS
  - Connect to other ASes using *gateway routers*
  - Use an *exterior gateway protocol (EGP)* to route packets to other AS's
- IGP: OSPF, RIP
- Today's EGP: BGP version 4

Recall: Security Goals

- CIA

Observation

- Lots of things designed for “working” and “internetworking”

- Security is missing or left as “out-of-band”

Sources of Network Vulnerabilities

- Protocol-level vulnerabilities
  - Implicit trust assumptions in design
- Implementation vulnerabilities
  - Both on routers and end-hosts
- Incomplete specifications
  - Often left to the programmers

Network Security Roadmap

- Packet sniffing and spoofing
- TCP/IP attacks
- DoS and DDoS
- DNS attacks
- BGP attacks
- Firewalls and IDS
- VPN
- SDN Security