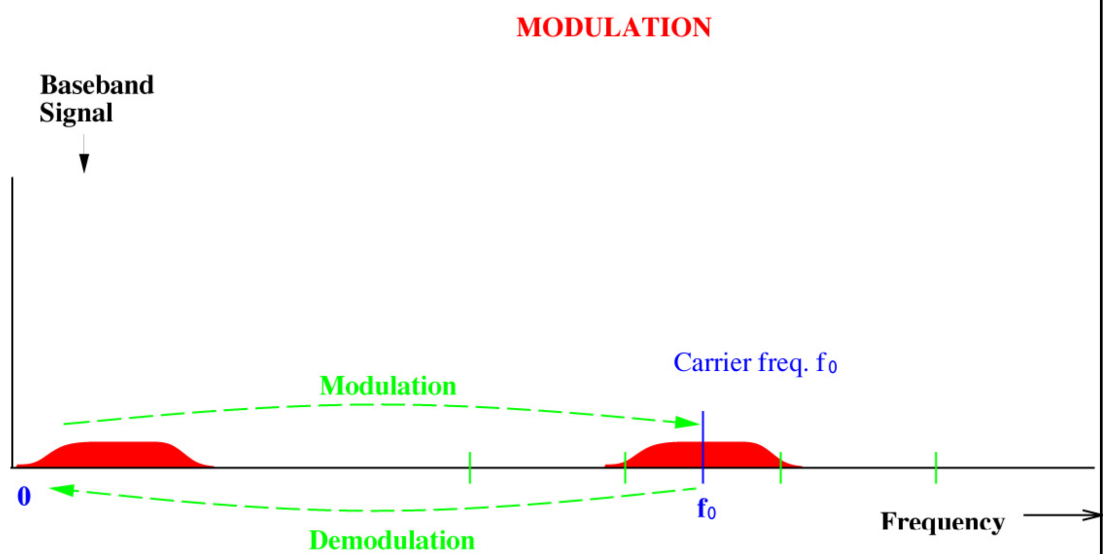
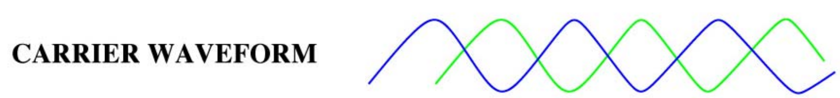
Introduction to Networks

Modulation and Demodulation

- Common examples: radio, television channels for analog signals
  - Bandwidth in hertz



- Can also be used for digital signals (encoding binary data)



$$A\cos(2\pi f_0 t+\theta)$$

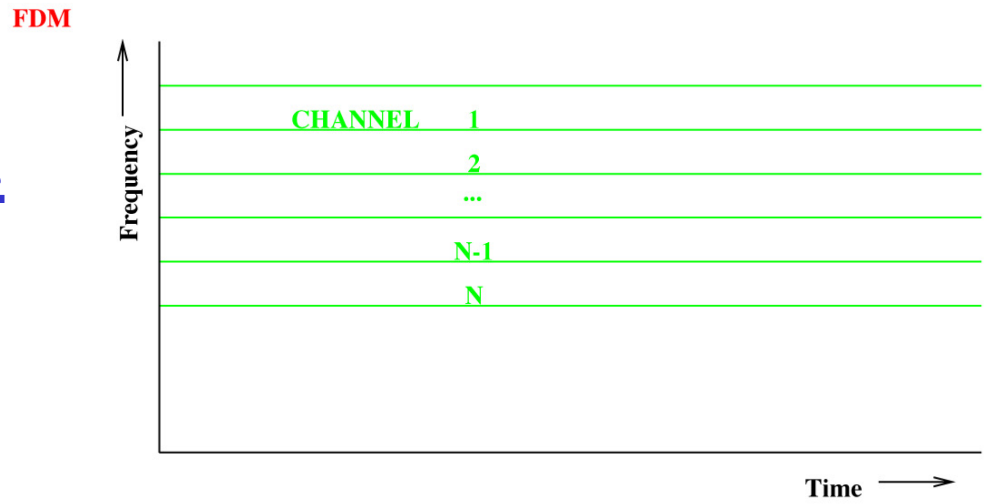
Shannon's Theorem

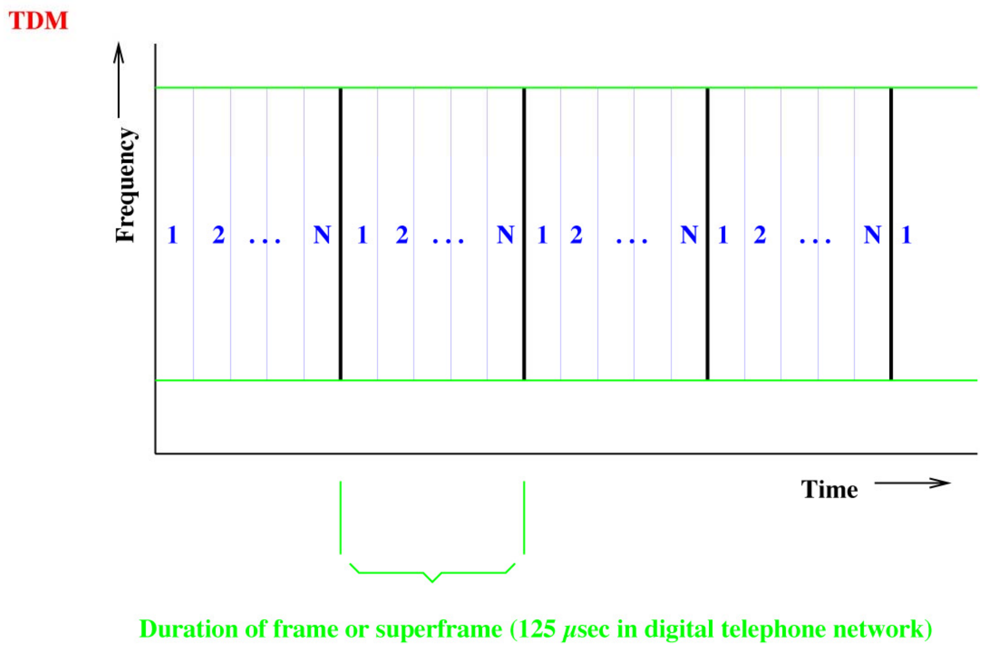
$$C = B \log_2 (1 + S/N)$$

where

C	max capacity in bits/sec
B	bandwidth in hertz
S/N	signal to noise ratio

FDM vs. TDM





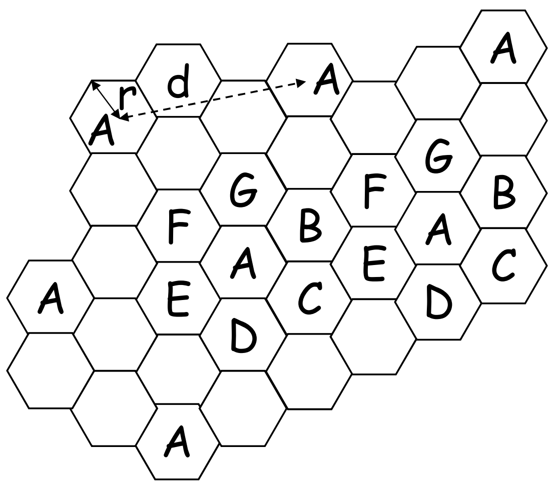
Duration of frame (or superframe) is 125 μsec in digital telephone networks

TDM in Telephone Networks

- Why 125 μsec for frame duration?
- Sampling Theorem: An analog signal can be reconstructed from samples taken at a rate equal to twice the signal bandwidth
- Bandwidth for voice signals is 4 Khz; for hi fidelity music, 22.05 Khz per channel
- Sampling rate for voice = 8000 samples/sec or one voice sample every 125 μsec
- Digital voice channel (uncompressed), 8 bits x 8000/sec = 64 Kbps

Other Multiplexing Techniques

- Space division multiplex
  - Same frequency used in different cables
  - Same frequency used in different (nonadjacent) cells

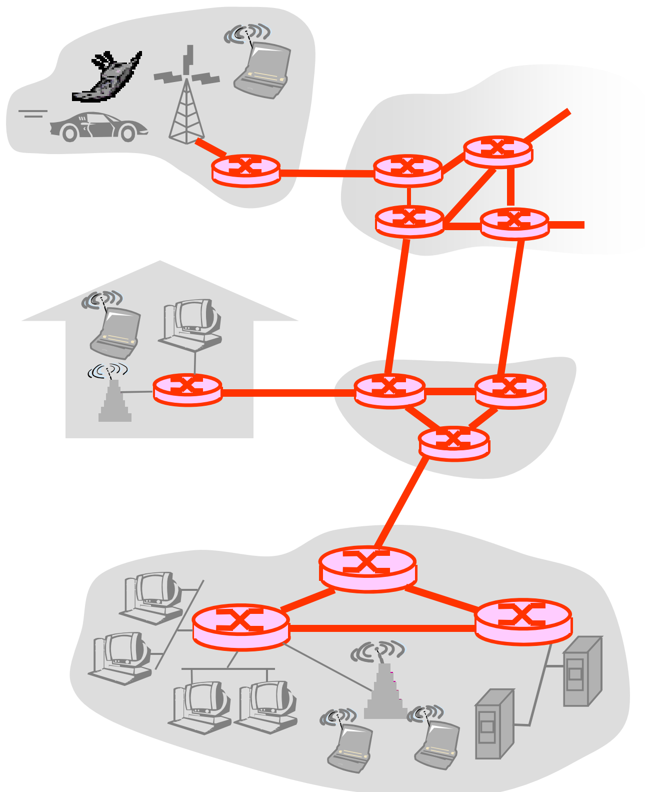


- Wavelength division multiplex
  - Light pulses sent at different wavelengths in optical fiber
- Code division multiplex

e.g., CDMA for cell phones

The Network Core

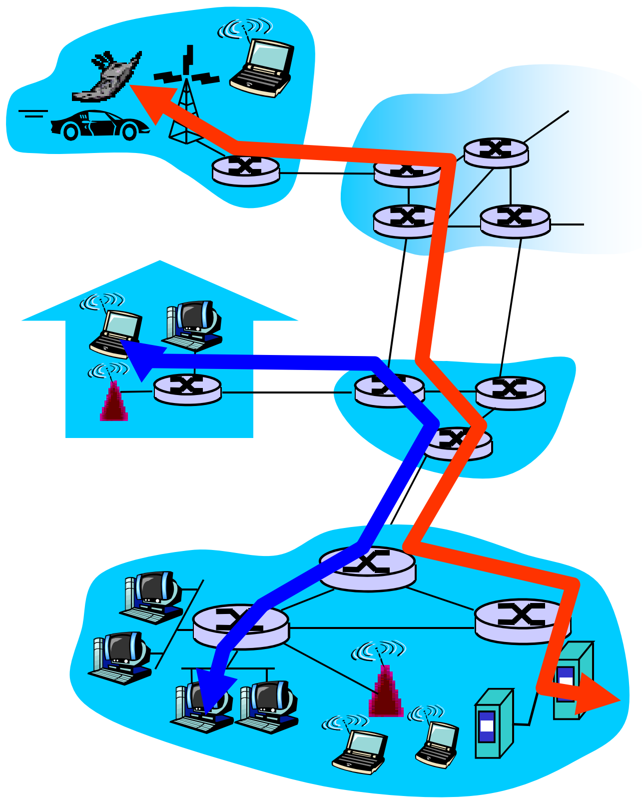
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



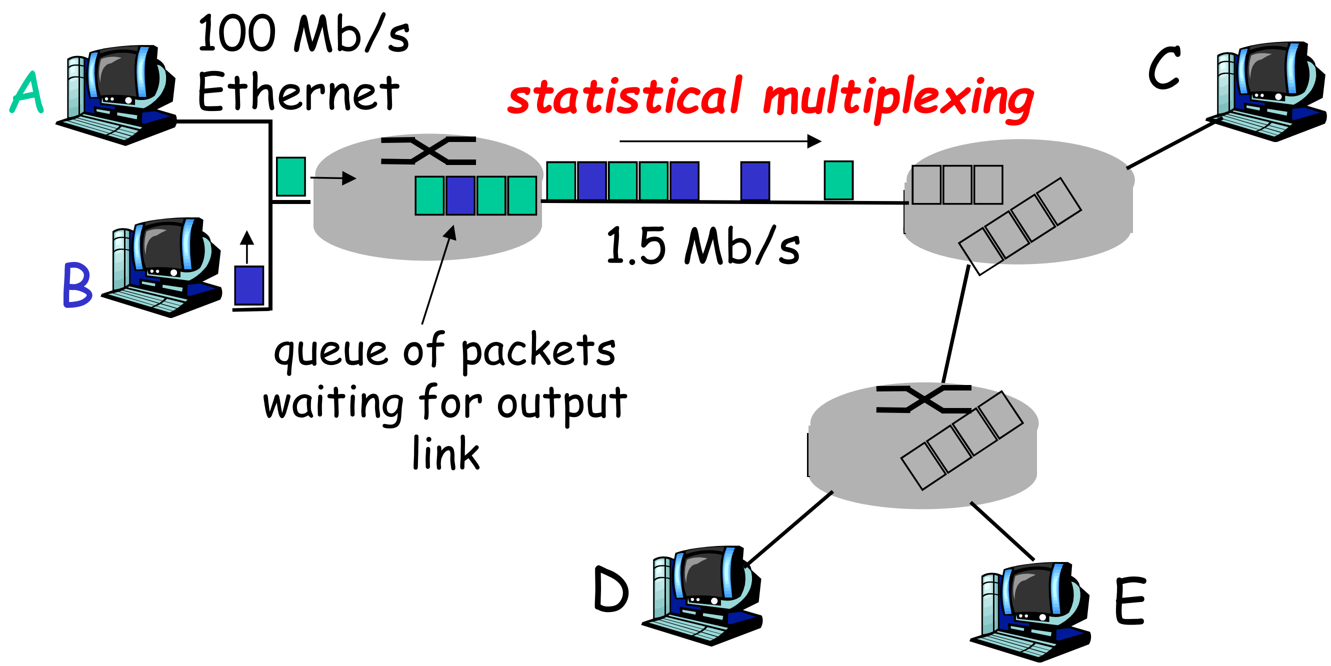
Network Core: Circuit Switching

End-to-end resources reserved for each "call"

- E.g., link bandwidth
  - FDM, TDM
- end-to-end circuit-like (guaranteed) performance
- call setup required
  - resource piece *idle* if not used by the call *(no sharing)*
  - *state information each step along the way*



Packet Switching: Statistical Multiplexing



- Sequence of A & B packets does not have fixed pattern
  bandwidth shared on demand → *statistical multiplexing*
- queueing delay, packet loss

Network Core: Packet Switching

each end-end data stream divided into packets

- packets of different users *share* network resources
- each packet uses full link bandwidth

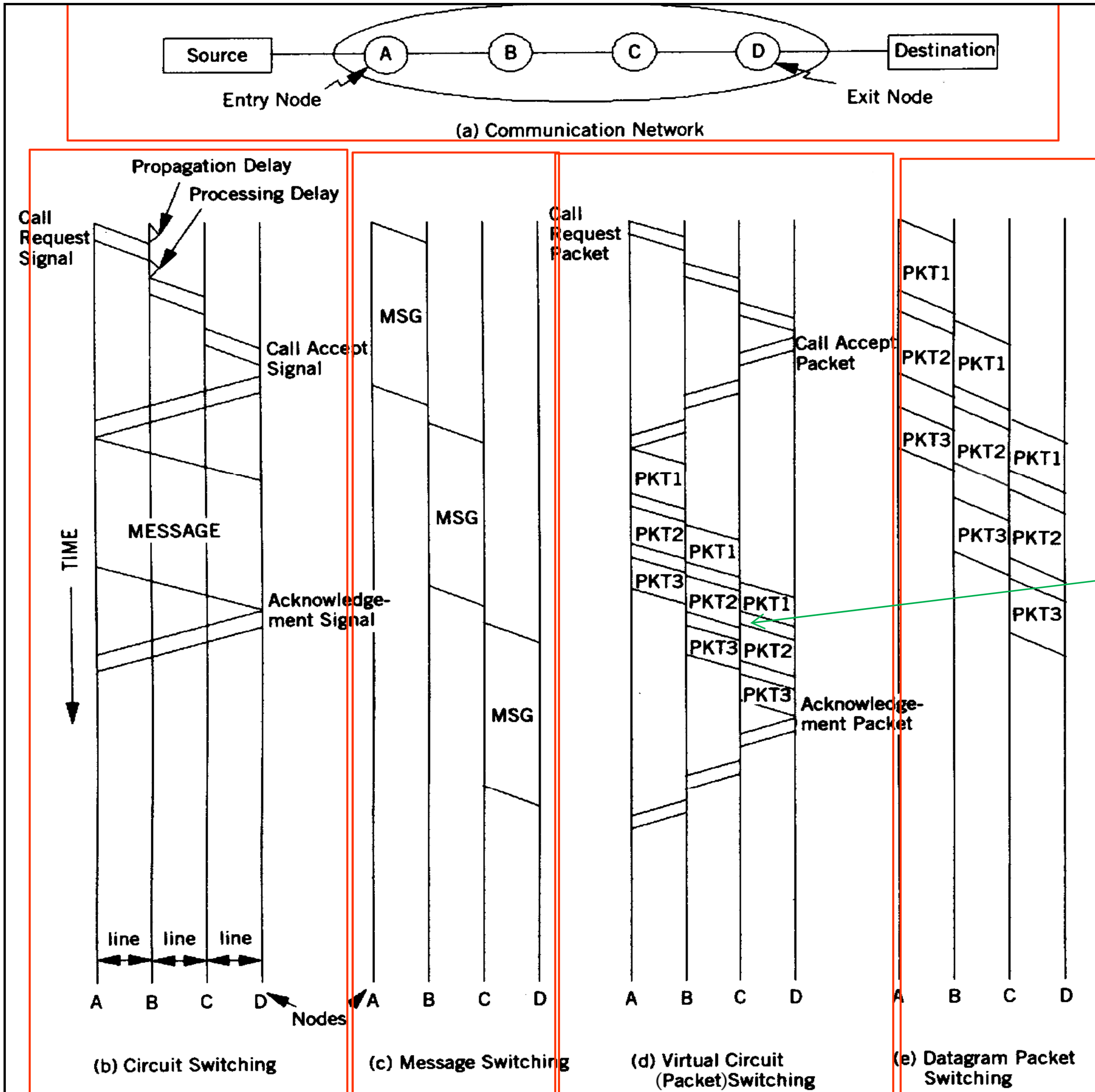
~~Bandwidth division into "pieces"~~
~~Dedicated allocation~~
~~Resource reservation~~

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
  - Each node receives the ***complete*** packet before forwarding it

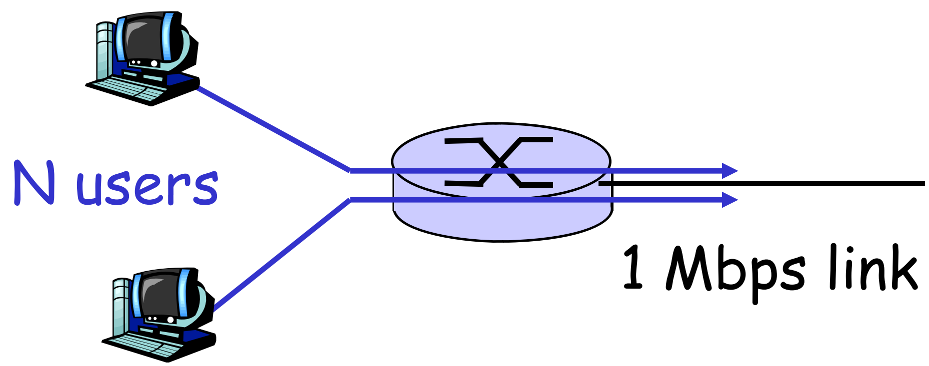
Circuit vs. Message vs. Packet Switching



violates store-and-forward?

Packet switching versus circuit switching

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time (a "bursty" user)
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004



Q: how did we get value 0.0004?

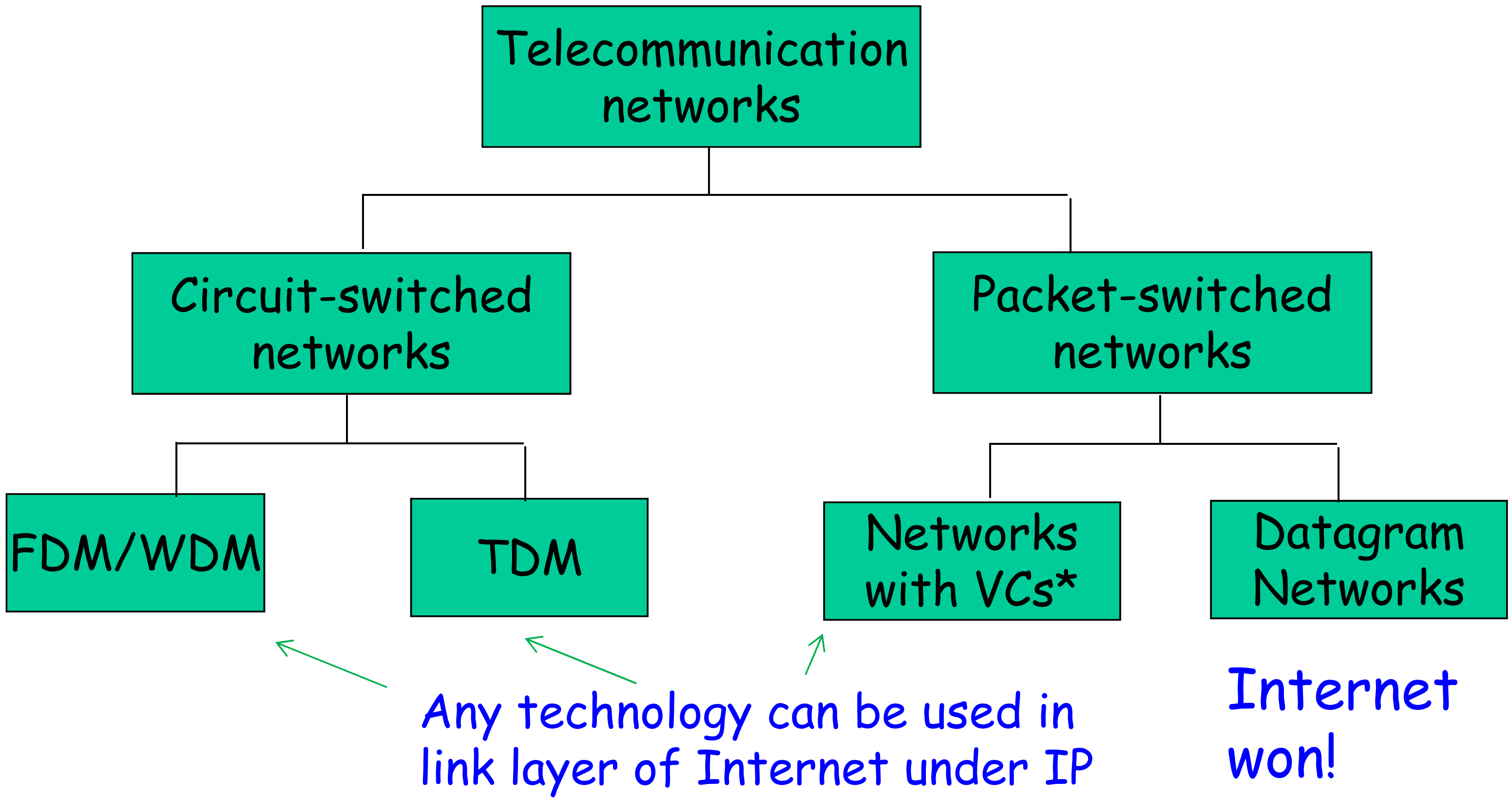
Packet switching versus circuit switching

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
- Q: How to provide circuit-like behavior?

  bandwidth guarantees needed for
  - *interactive* audio/video apps
  - providing virtual links to enterprise network customers (under service contracts)

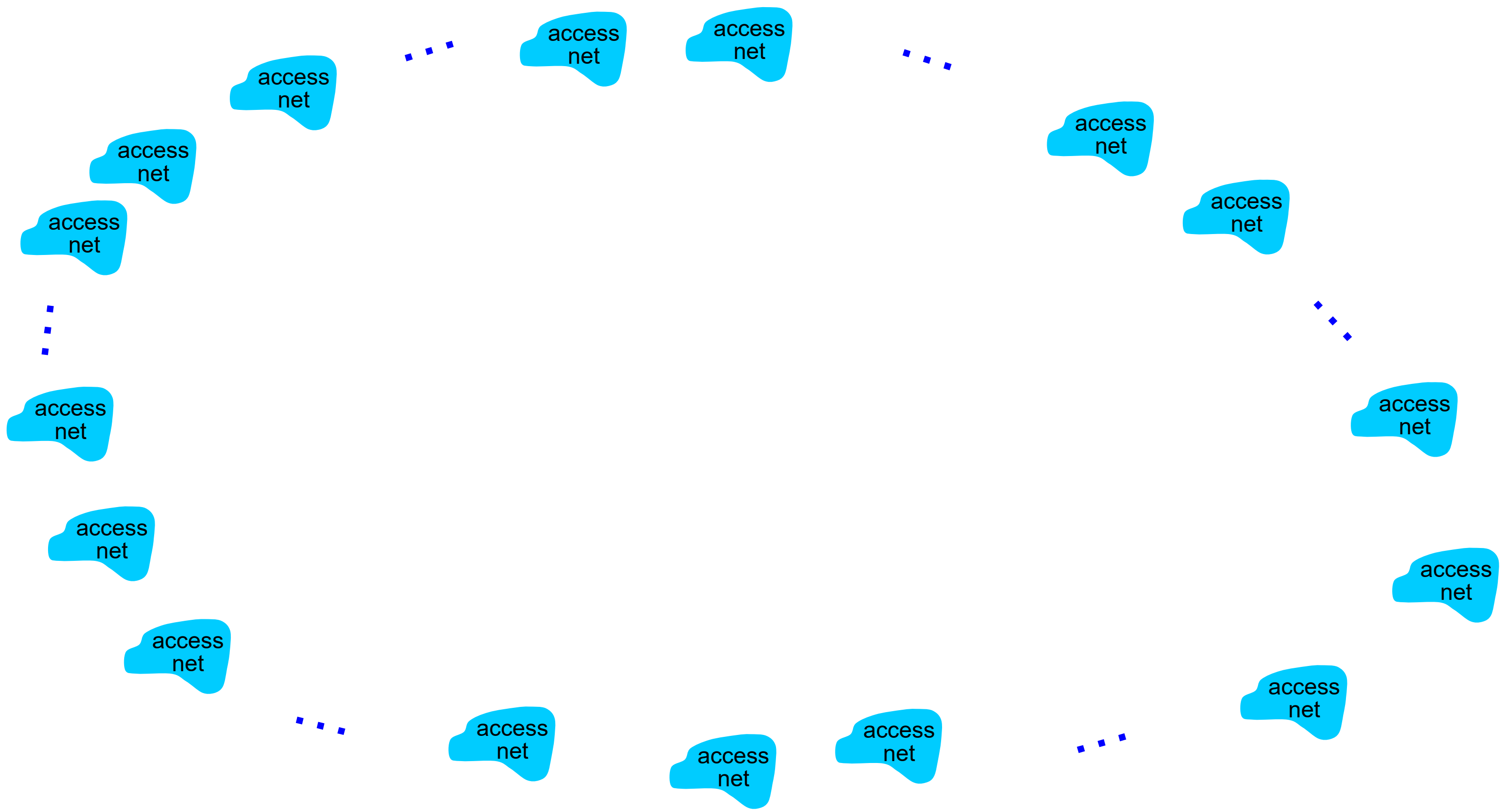
Network Taxonomy



VC examples: ATM networks, MPLS tunnels

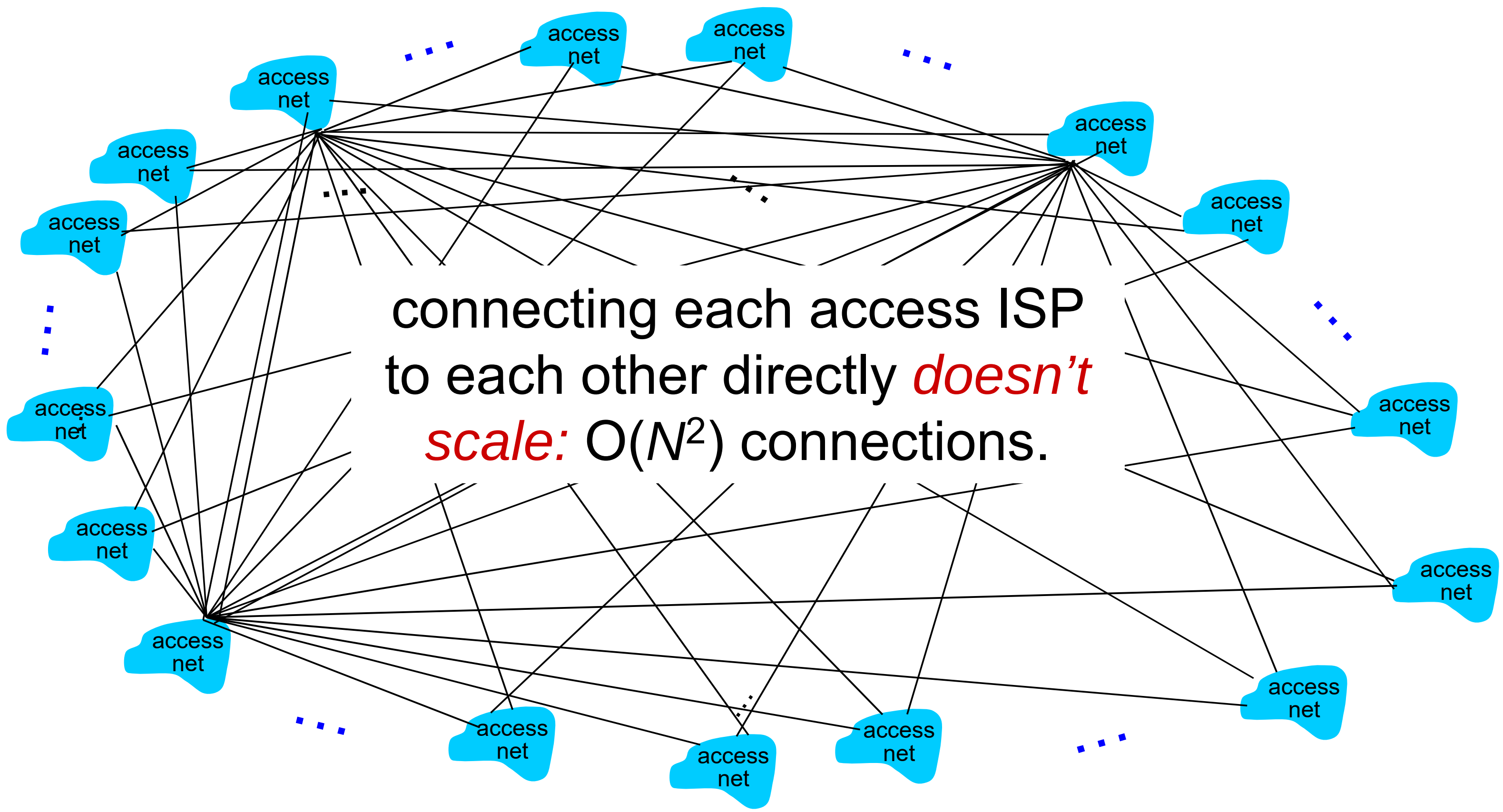
Internet structure: network of networks

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



Internet structure: network of networks

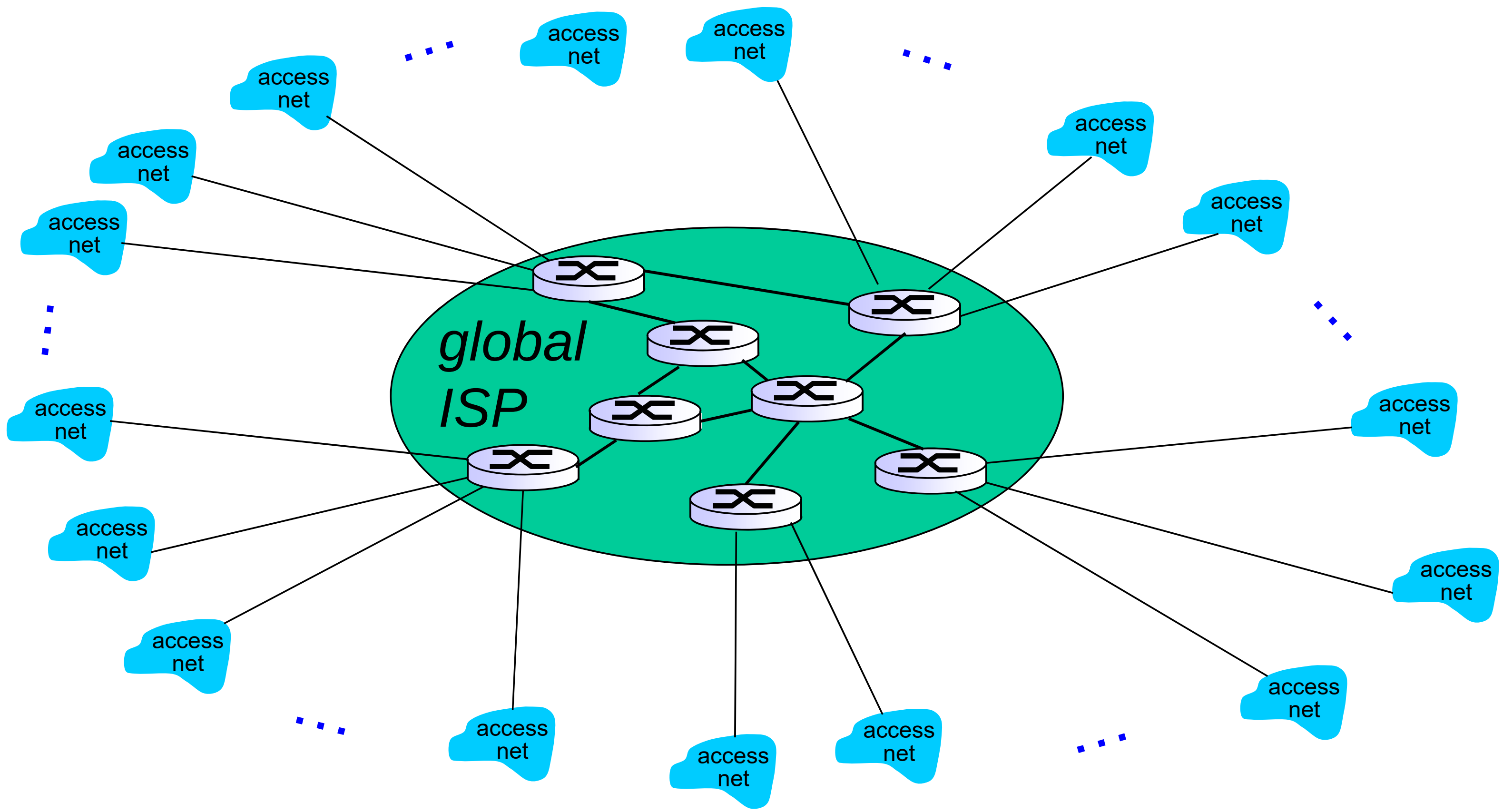
Option: connect each access ISP to every other access ISP?



connecting each access ISP to each other directly *doesn't scale:* $O(N^2)$ connections.

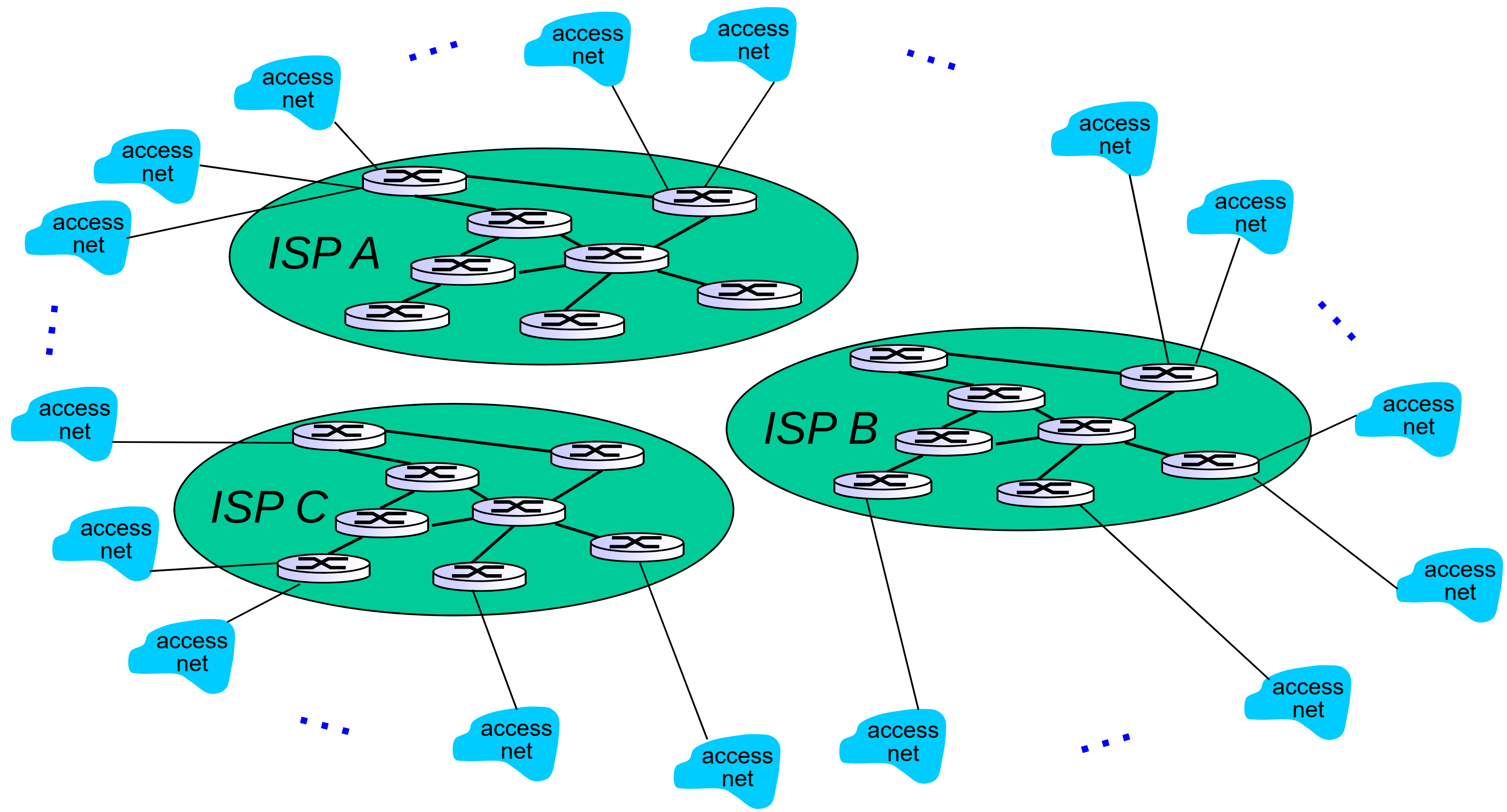
Internet structure: network of networks

1970-1991: connect each access ISP to a global transit ISP: 1. *Financed by US government:* ARPAnet, NSFnet



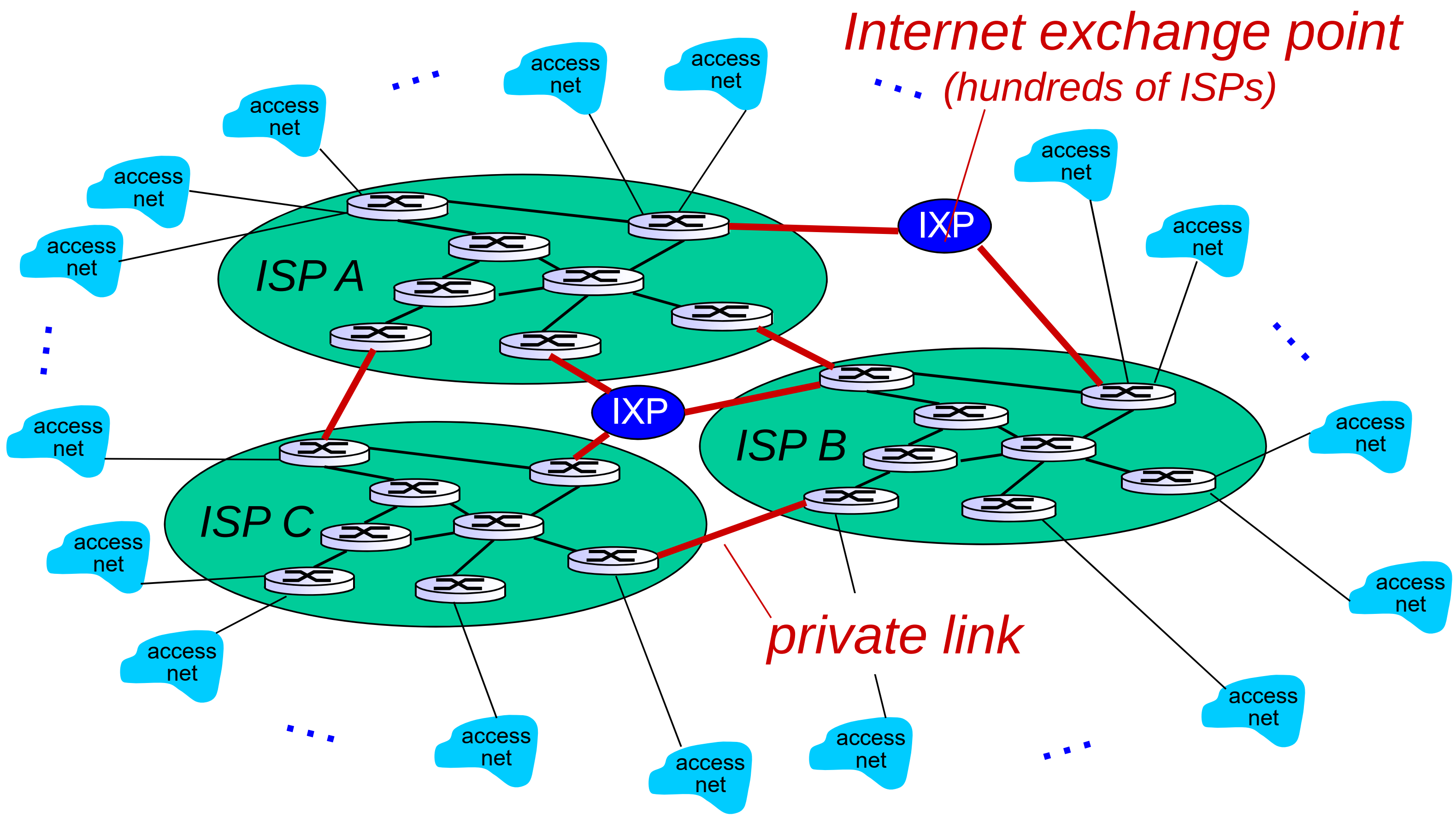
Internet structure: network of networks

Post 1991 -Transition to commercial ISPs - If one global ISP is viable business, there will be competitors ….



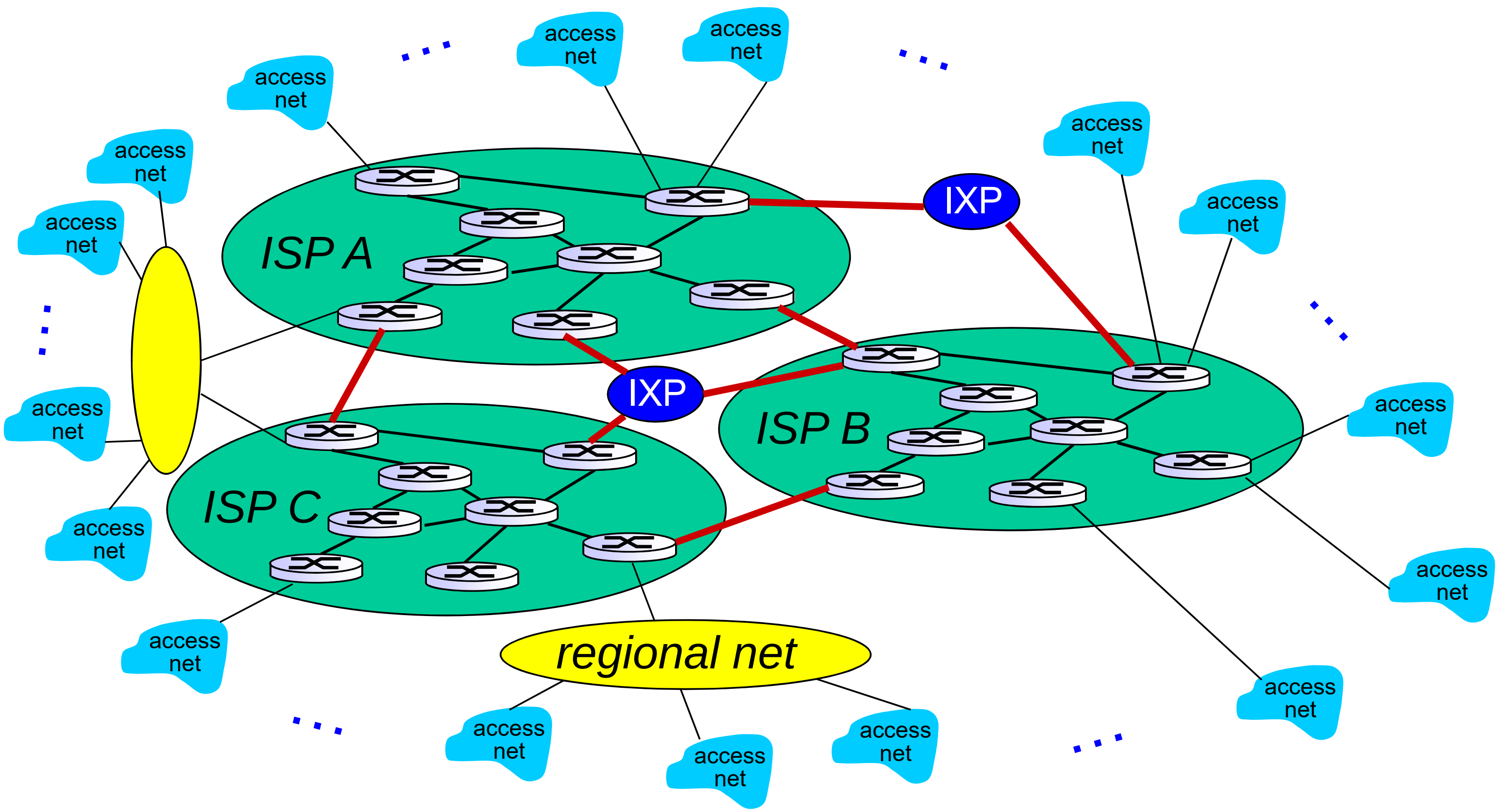
Internet structure: network of networks

Two ISPs are connected in a "provider-customer" or "peer-peer" relationships according to peering agreements



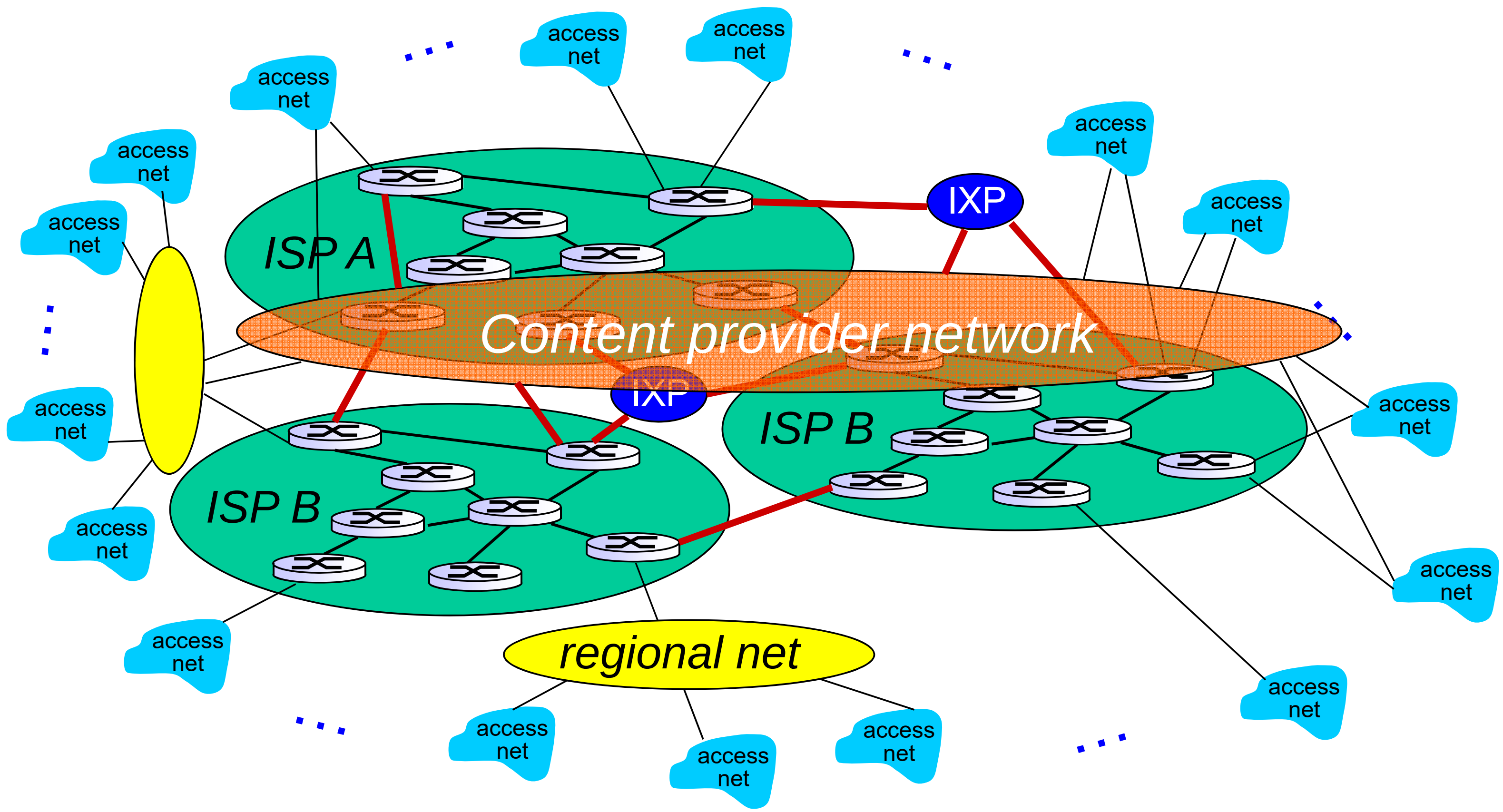
Internet structure: network of networks

... and regional networks may arise to connect access nets to ISPs

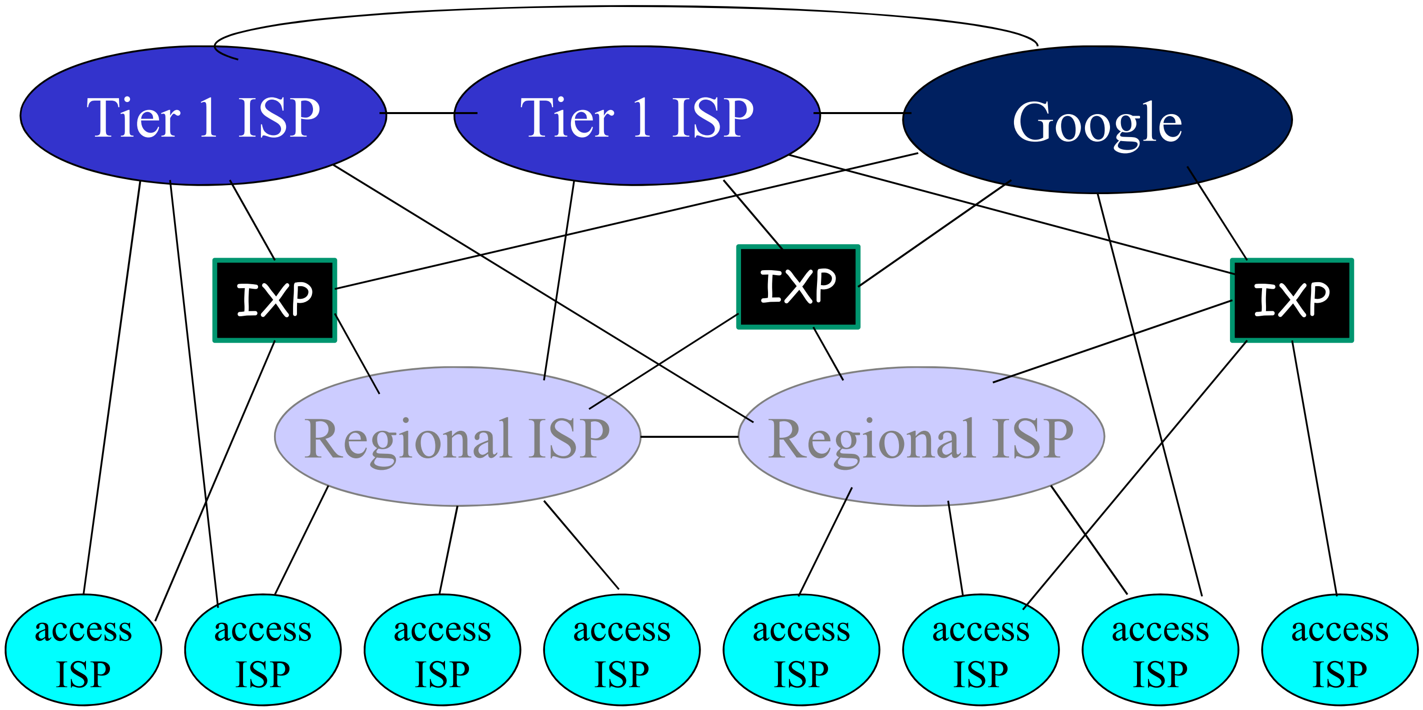


Internet structure: network of networks

... and a content provider network (e.g., Akamai, Google, Microsoft) may run its own network to bring services, content close to end users



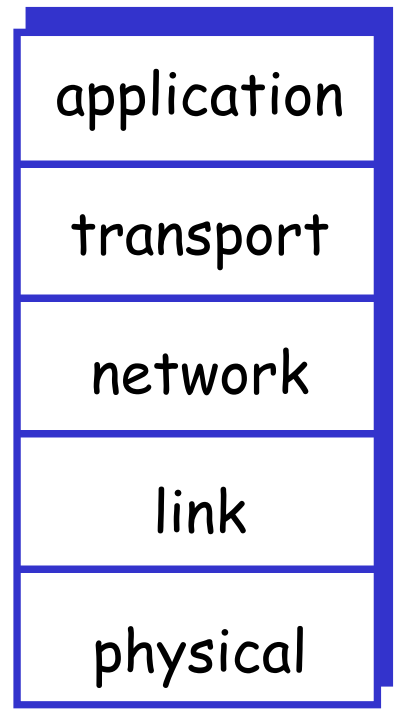
Internet structure: network of networks



- 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 networks (e.g., Google): private network that connects its **data centers** to Internet, often bypassing tier-1 and regional ISPs

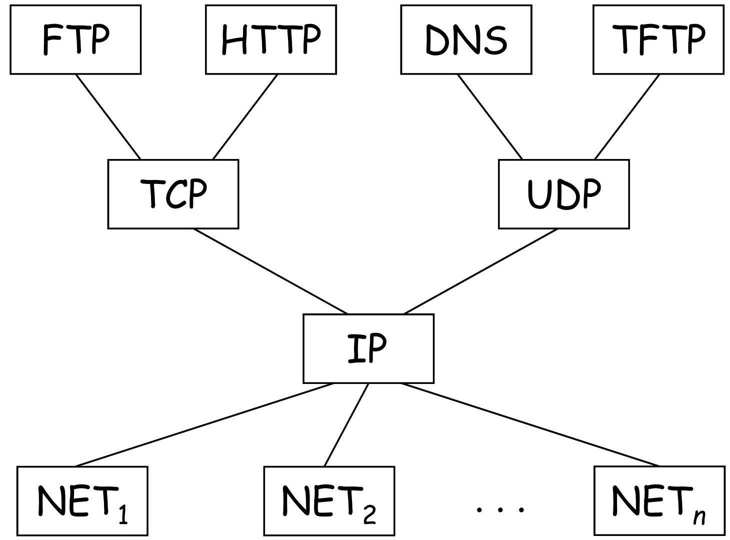
Internet protocol stack

- application: protocols that support network applications
  - SMTP, HTTP, DNS
- 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, 802.11 (WiFi)
- physical: how to send and receive bits



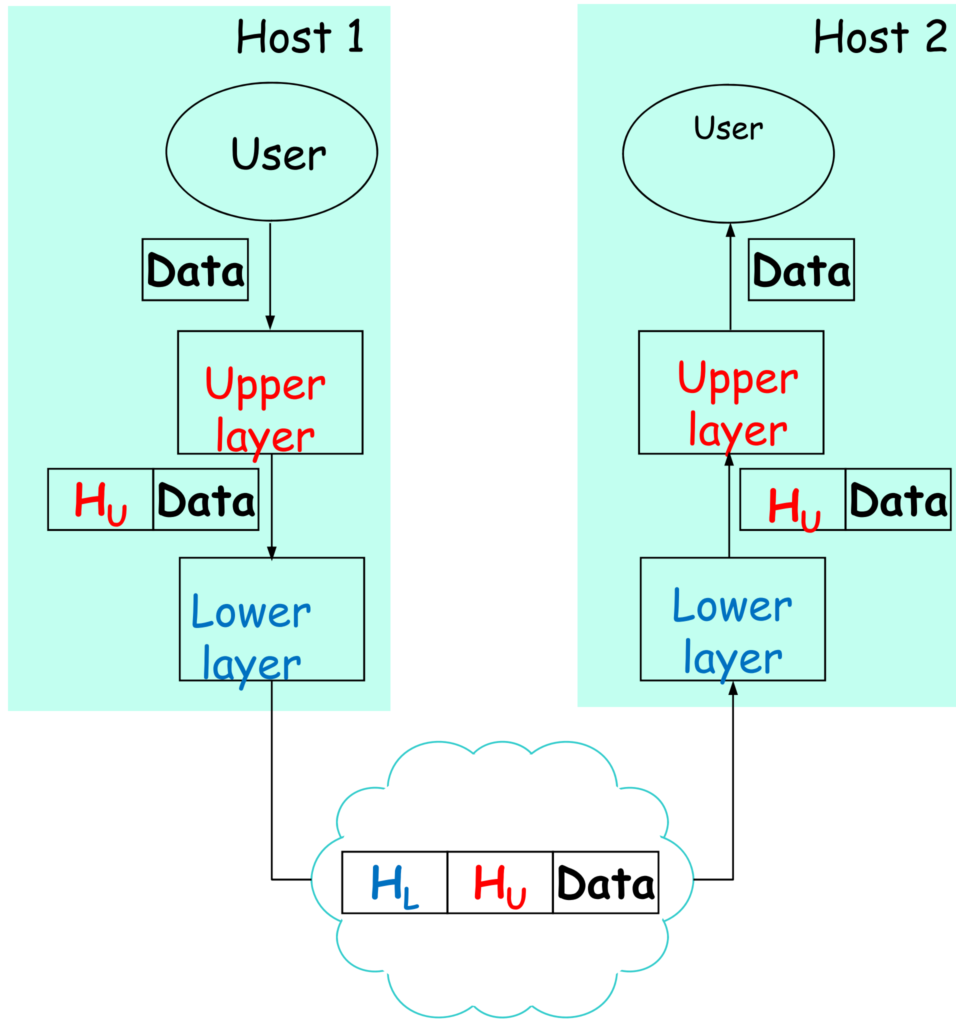
Internet Architecture

- Internet Engineering Task Force (IETF)
- application protocols support applications
- *hourglass* shape (only IP in network layer)
  - best effort service => any delivery service can be used by IP
- limitation of hourglass



Encapsulation

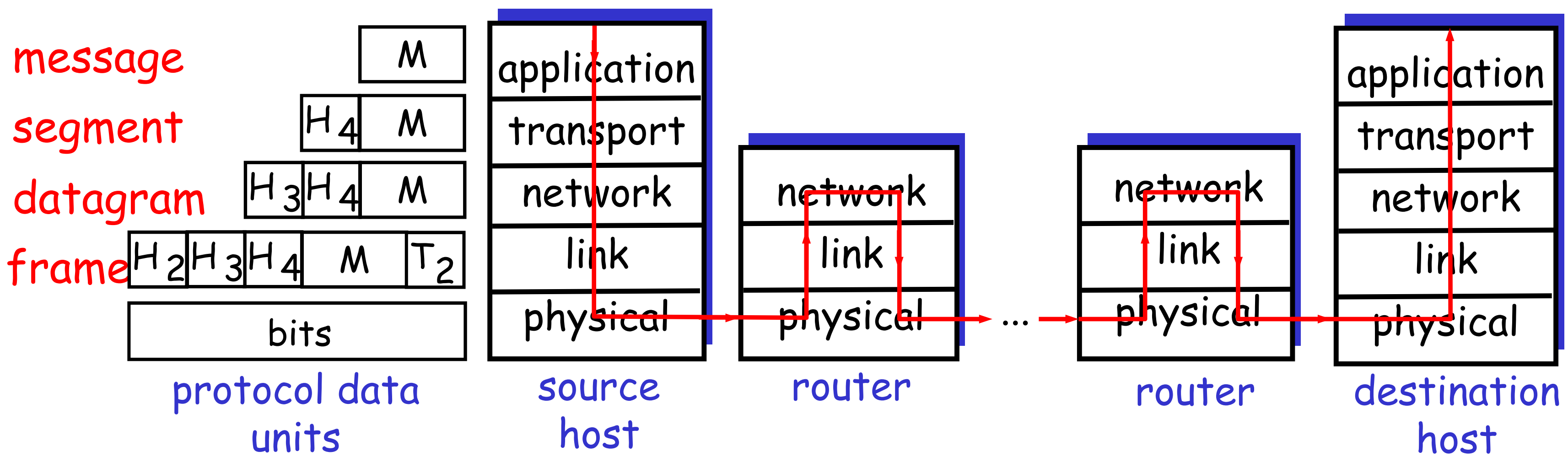
- Protocol peers provide a data delivery service
- How do protocol peers in different machines exchange protocol messages between themselves?
  - In protocol header encapsulated and de-encapsulated



Physical path of data

Each layer takes data (service data unit) from above

- adds header to create its own protocol data unit
- passes protocol data unit to layer below

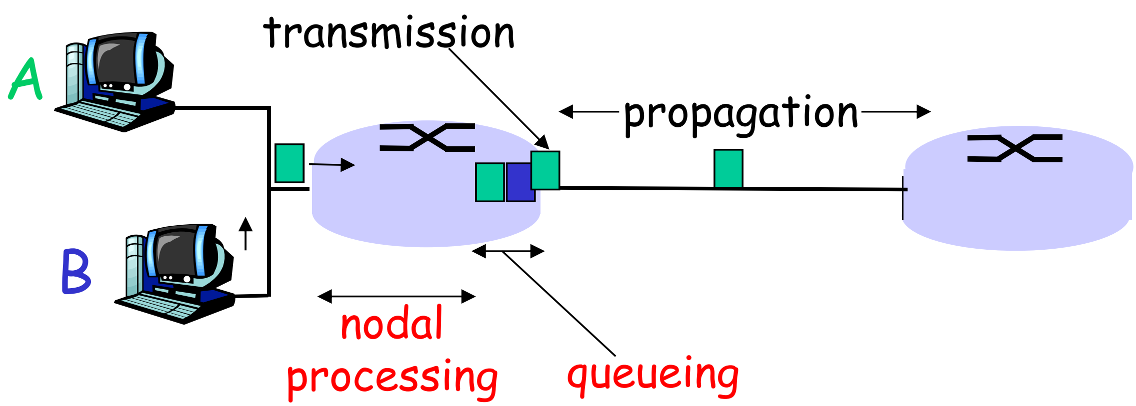


Note: In the past, a switch implements only two layers (physical and link). Nowadays many switches function as routers (3 layers)

Note: layer $2\frac{1}{2}$ may exist (i.e. virtual circuits)

Four sources of packet delay

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

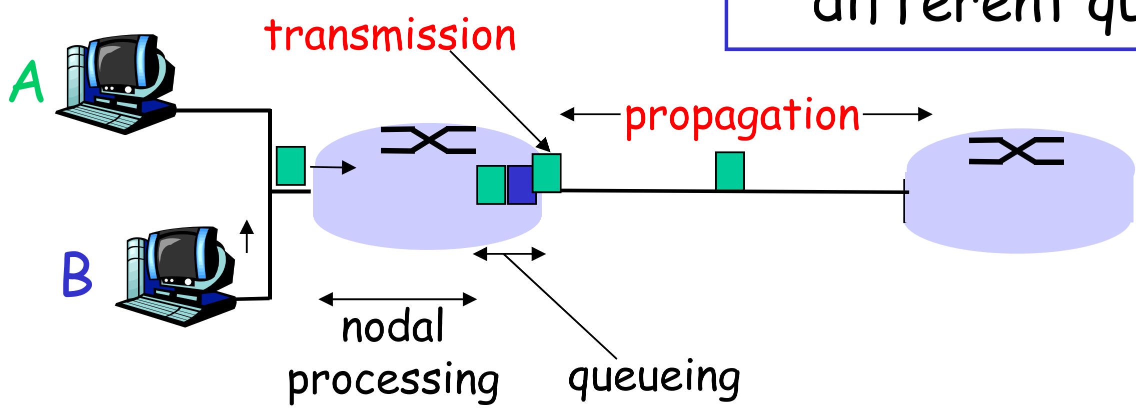


Delay in packet-switched networks

3. Transmission delay:
- R: link bandwidth (bps)
- L: packet length (bits)
- time to send bits into link = **L/R**

4. Propagation delay:
- d: length of physical link
- s: propagation speed in medium (~2x10^8 m/sec)
- propagation delay = **d/s**

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



End-to-End Delay

- Nodal delay (from when last bit of packet arrives at this node to when last bit arrives at next node)

  $d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

- End-to-end delay over N identical nodes/links from client c to server s (from when last bit of packet leaves client to when last bit arrives at server)

  $d_{c\text{-}s} = d_{prop} + N d_{nodal}$

- Round trip time (RTT)

  $RTT = d_{c\text{-}s} + d_{s\text{-}c} + t_{server}$

  where t_{server} is server processing time

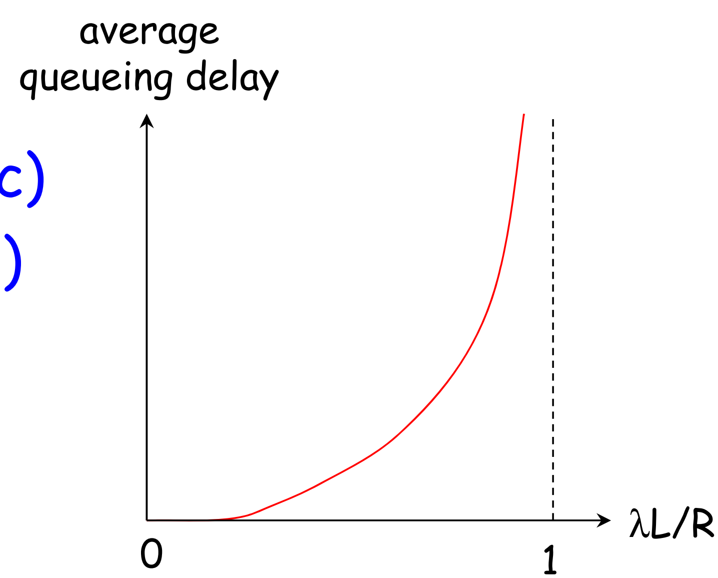
Large delay usually from queueing or loss at "bottleneck"

Queueing delay (waiting time)

- R: link bandwidth (bps)
- L: packet length (bits)
  - service rate = R/L (pkts/sec)
  - ave. service time = L/R (sec)
- λ: packet arrival rate

traffic intensity =

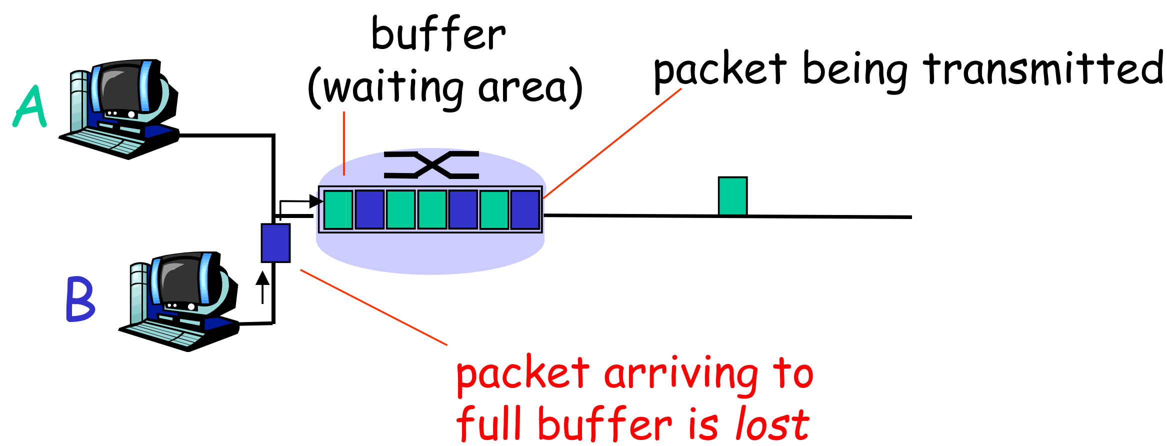
arrival rate/service rate = $\lambda L/R$



- $\lambda L/R \sim 0$: average queueing delay small
- $\lambda L/R$ -> 1: delays become large
- $\lambda L/R > 1$: more "work" arriving than can be served, average delay infinite!
  - In reality, buffer overflow when $\lambda L/R$ -> 1

Packet loss

- buffer in router for each link has finite capacity
- lost packet may be retransmitted by previous node, by source end system, or not at all



End of Introduction