CS-340 Introduction to Computer Networking

Lecture 2: The Internet core, and Layering

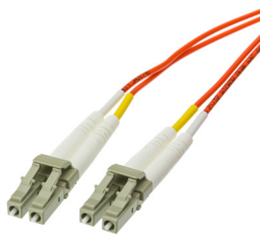
Steve Tarzia

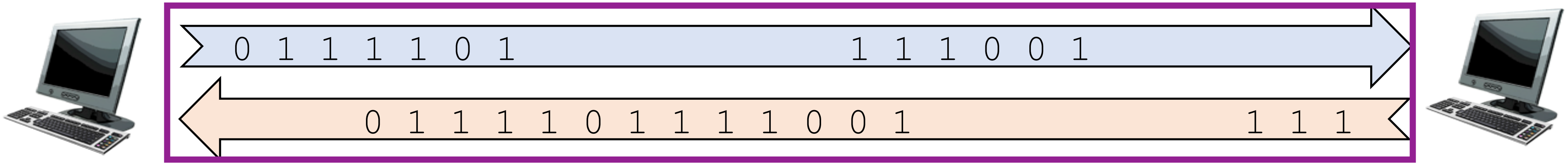
Many diagrams & slides are adapted from those by J.F Kurose and K.W. Ross

Last Lecture

- Gave a high-level view of the Internet
- Various physical media and access network types are used
- It's loosely-couple and de-centralized
  - Everyone's invited to the Internet, assuming you follow the protocols and can somehow connect.
- A ***protocol*** is a set of rules for communication.
- Telephone network uses circuit switching (*reserves* end-to-end path).
- Internet uses packet switching (store-and-forwarded along path)
  - Delivery is "best effort," not guaranteed.

Simple Communication Link Model: *tubes?*



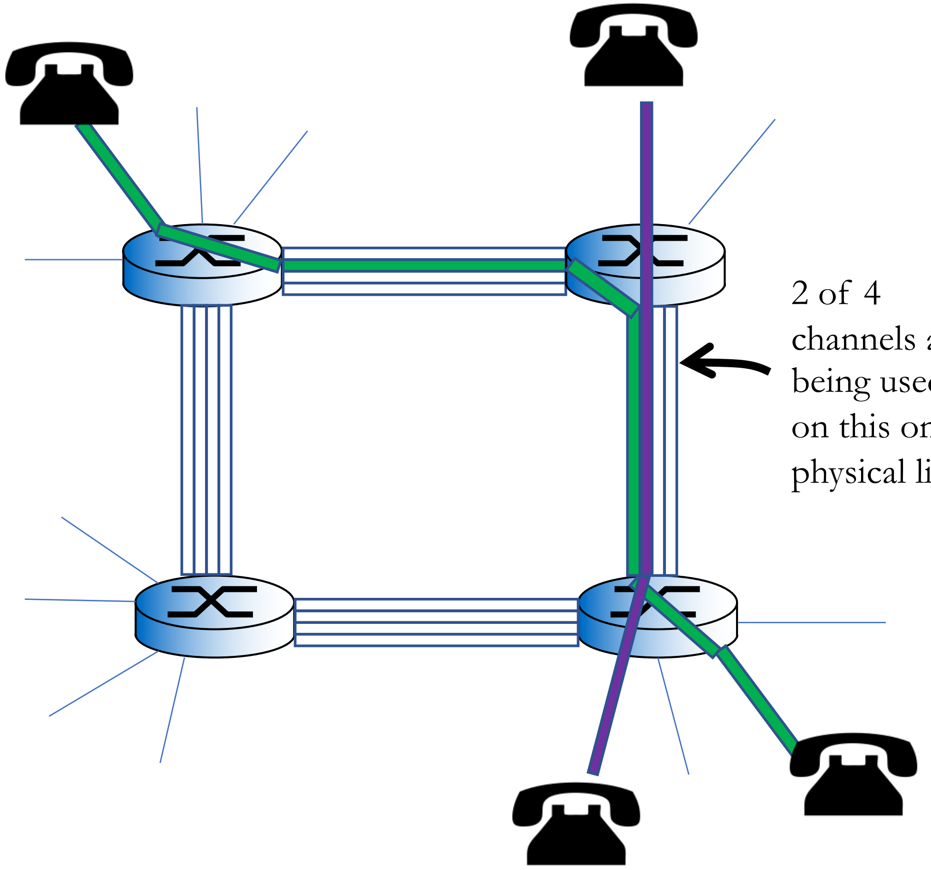


Chapter 6 has more detail, but for now, a **communication link** is:

- A bi-directional, point-to-point connection (one party on each side).
- Allows 0's or 1's to be transmitted (or it can be *idle*, sending nothing).
- It has a *propagation delay* – the time it takes for bits to flow from sender to receiver (at most the speed of light!).
  - This is the link **latency**. Often determined by the length of the link.
- Operates at a certain, constant bitrate (bits per second).
  - This is the link **throughput**. Determined by the technology of the link.

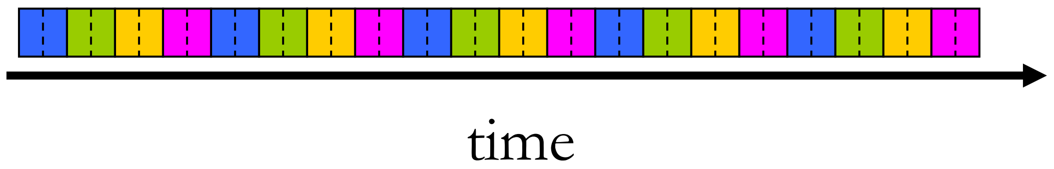
Circuit Switching Review

- If two people wanted to talk, a dedicated electrical path (circuit) is established.
- The circuit remains dedicated to the call until the parties hang up.
- Physical links in the of the network are divided into several virtual channels and thus can be shared by a finite number of users.

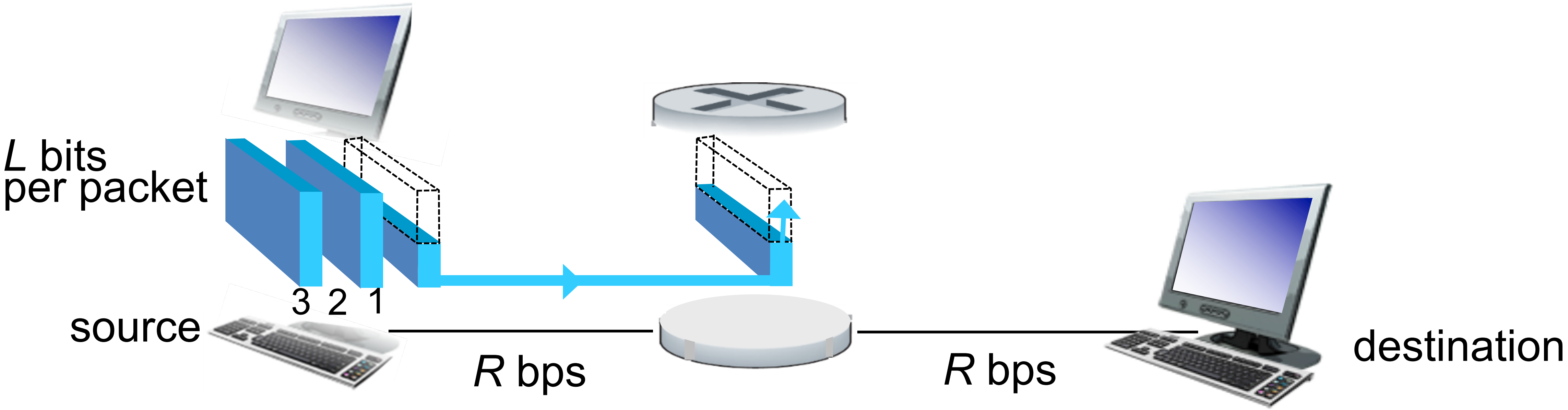


2 of 4 channels are being used on this one physical link

Time division multiplexing (TDM) 4 users: 🟦🟩🟨🟪



Packet-switching uses "store and forward"

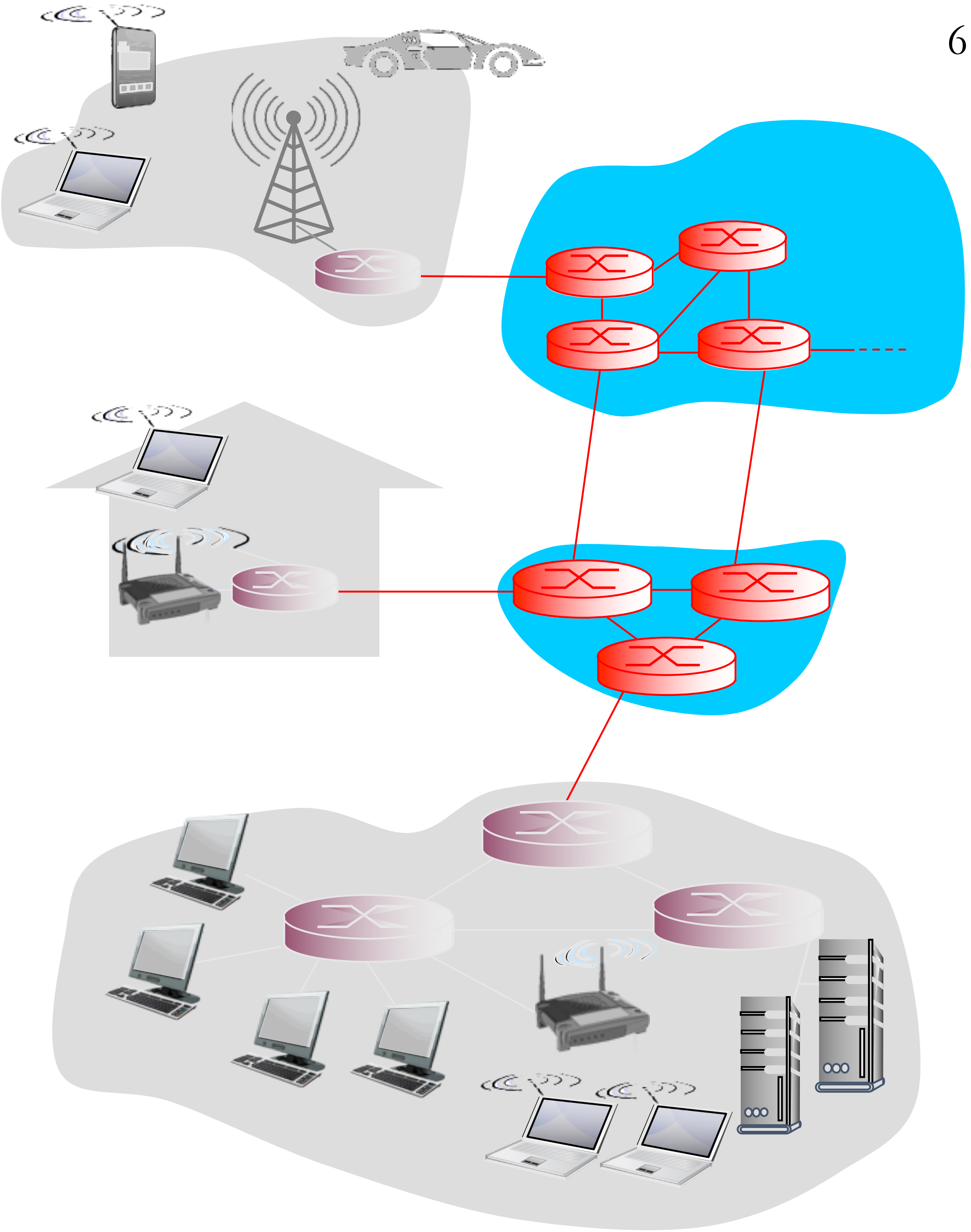


- In any network, data takes several *hops* (steps) to reach destination.
- In *packet switching*, each intermediate router has a queue of packets waiting to be sent along the next hop. Unlike circuit switching, there is no reserved capacity (time slot) for the data to be sent. Instead send packets in FIFO order. This is a more **dynamic** way to share the link!

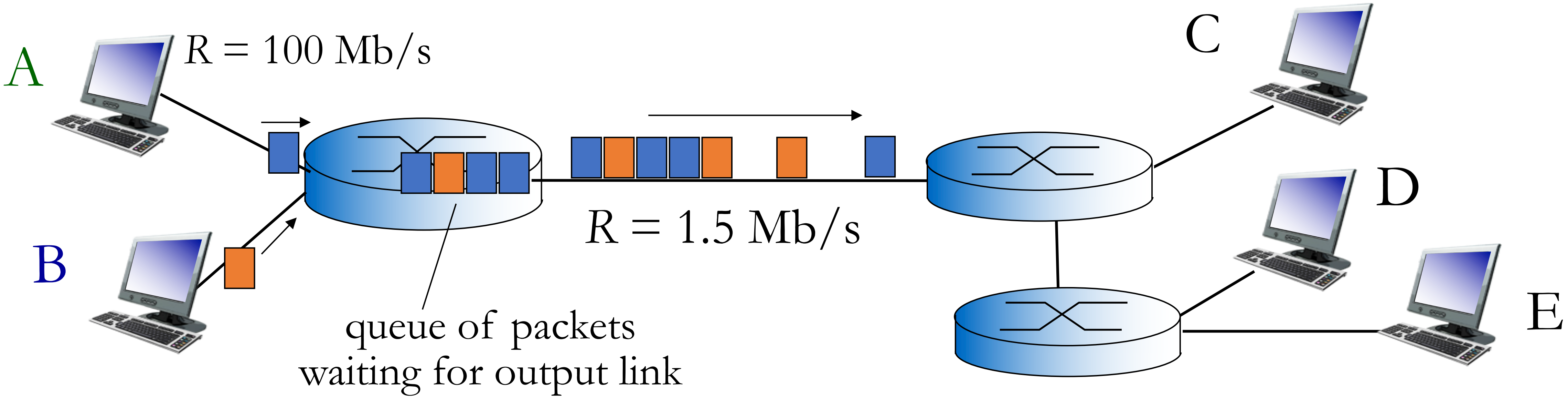
Routers form the network core

Routers have two responsibilities:

- *Distributed routing algorithms* to determine which address ranges are most quickly reachable on each of its outbound links.
- *Packet forwarding*, to direct packets according to the decisions made above.

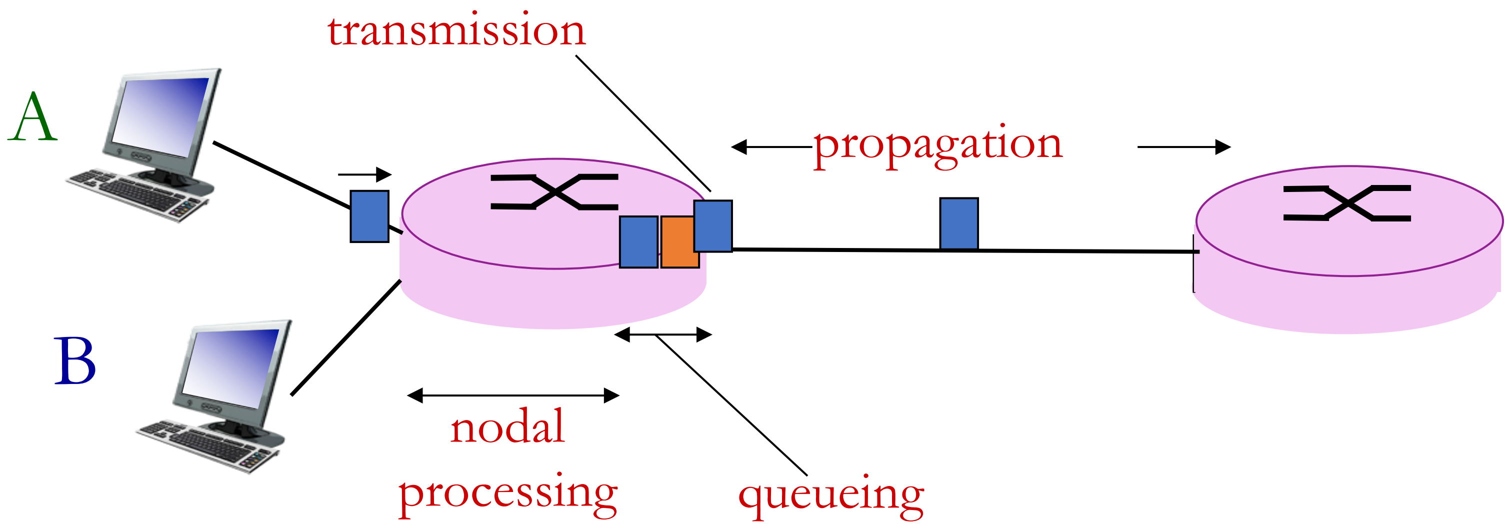


Queueing delay and loss



- Packets arriving at a router are placed on a finite-sized queue.
  - Waiting in the queue introduces delay
  - If the queue is full, new packets are dropped
    - Remember that packet delivery is not guaranteed on the Internet.

Four sources of packet delay



$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$

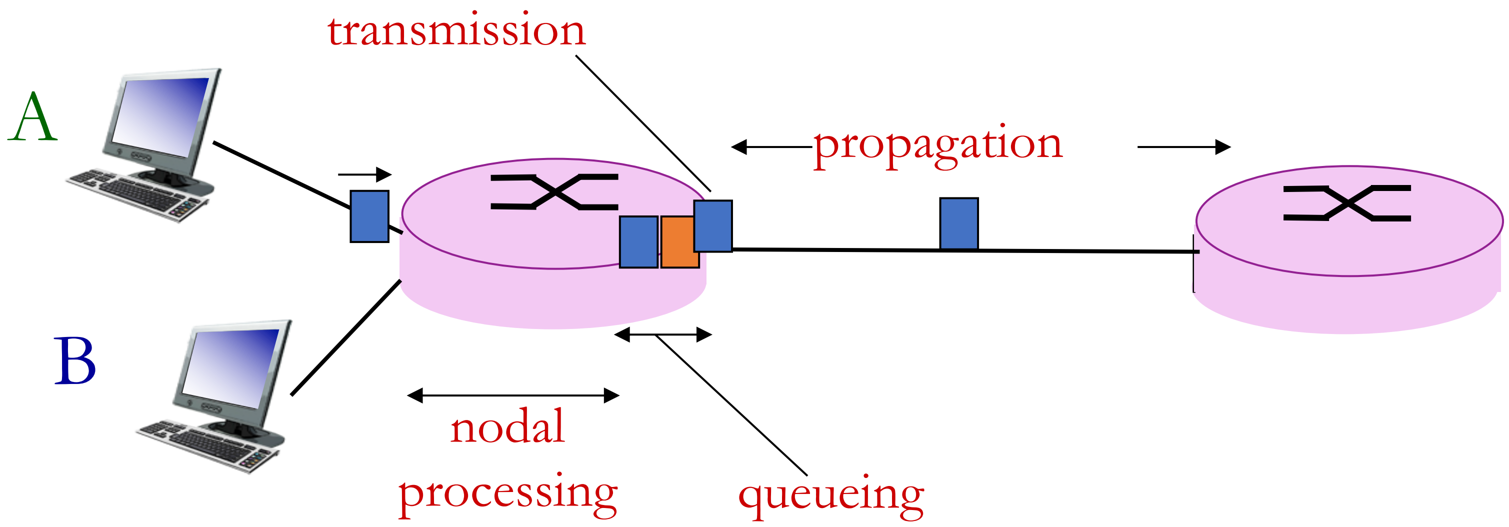
d_{proc}: nodal processing delay

- check bit errors
- choose output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of the outbound link

Four sources of packet delay



$$\boxed{d_{\mathrm{nodal}} = d_{\mathrm{proc}} + d_{\mathrm{queue}} + d_{\mathrm{trans}} + d_{\mathrm{prop}}}$$

$\mathrm{d}_{\mathrm{trans}}$: transmission delay

= packet size (bits)
/ link bandwidth (bps)

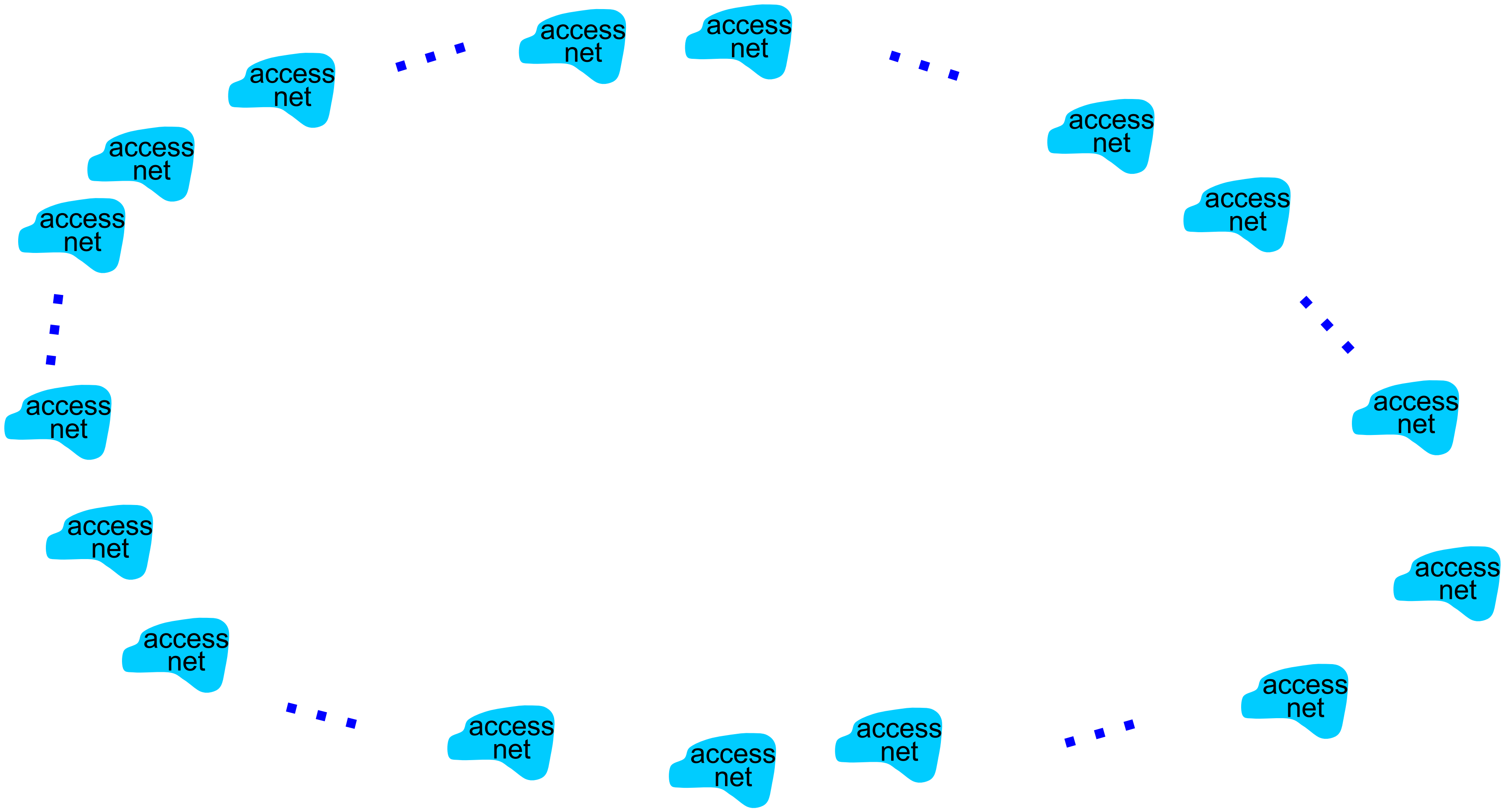
$\mathrm{d}_{\mathrm{prop}}$: propagation delay

≅ length of the link
/ speed of light (~$2\mathrm{x}10^8$ m/s)

The Internet is a network of networks

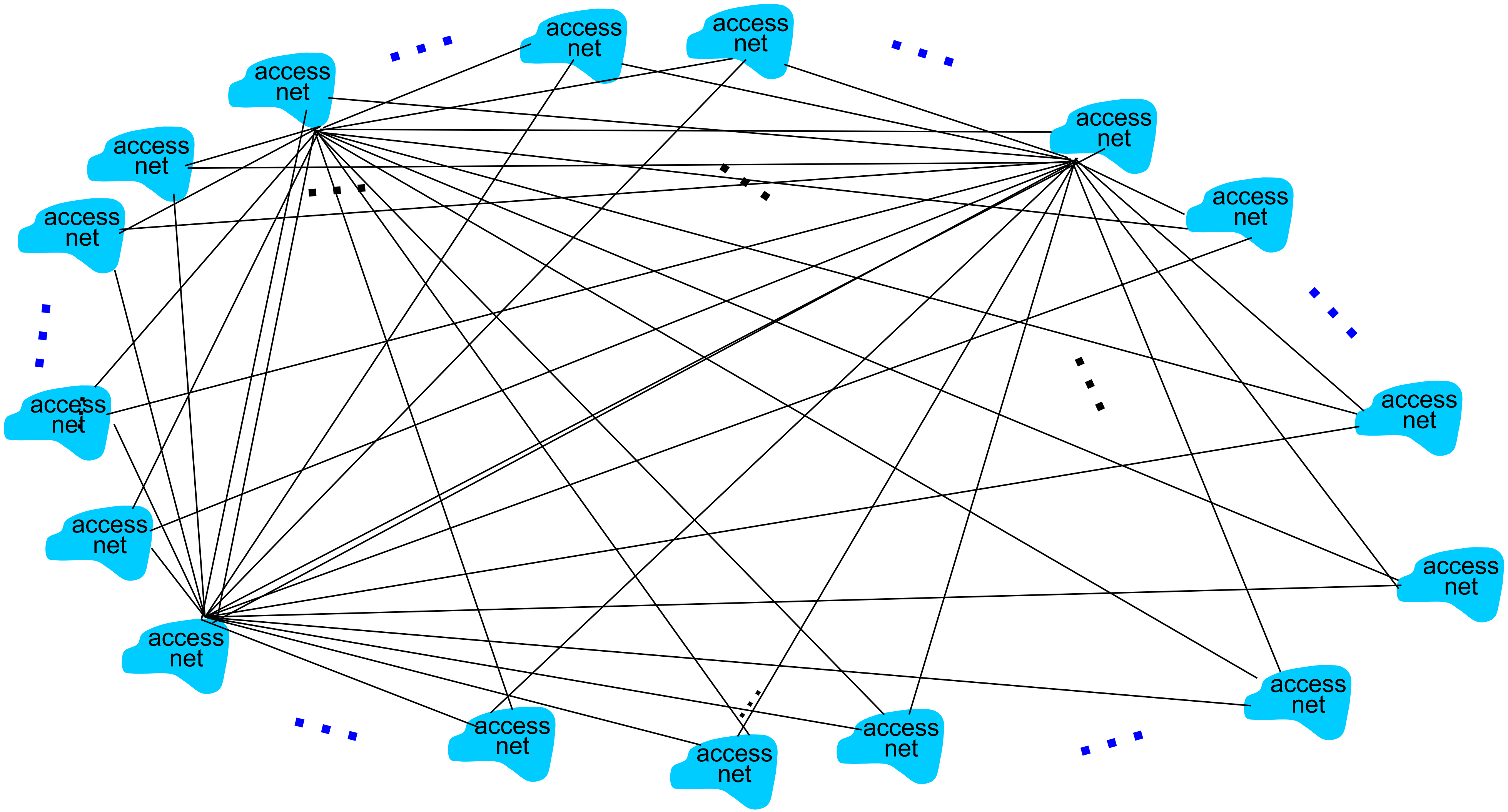
- We know that each local area will have its own telephone and cable companies, universities, etc. *How should we connect all these local ISPs?*

STOP
and
THINK



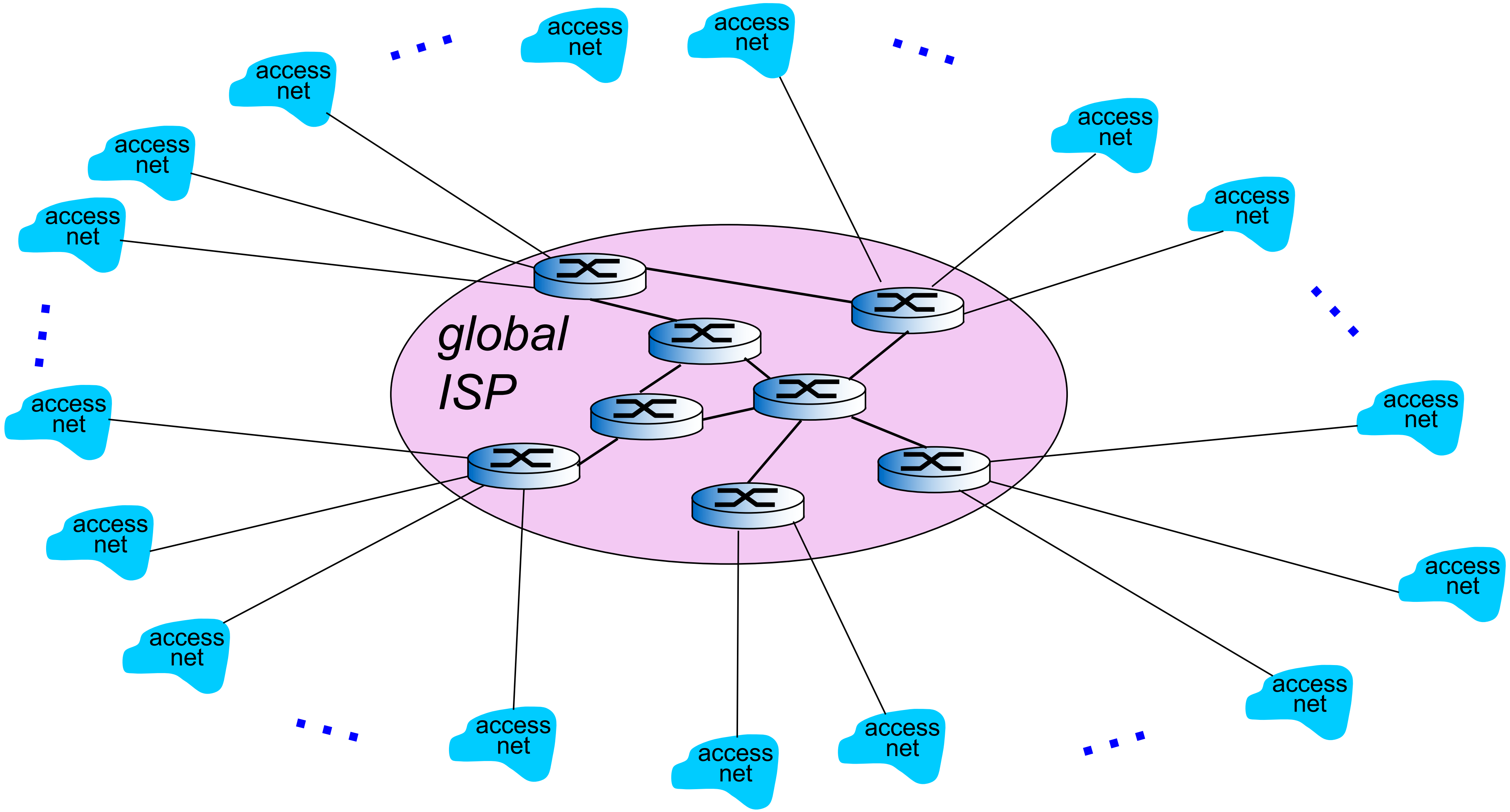
Idea 1: Connect them pairwise?

- But this would require N^2 connections, and very long connections!



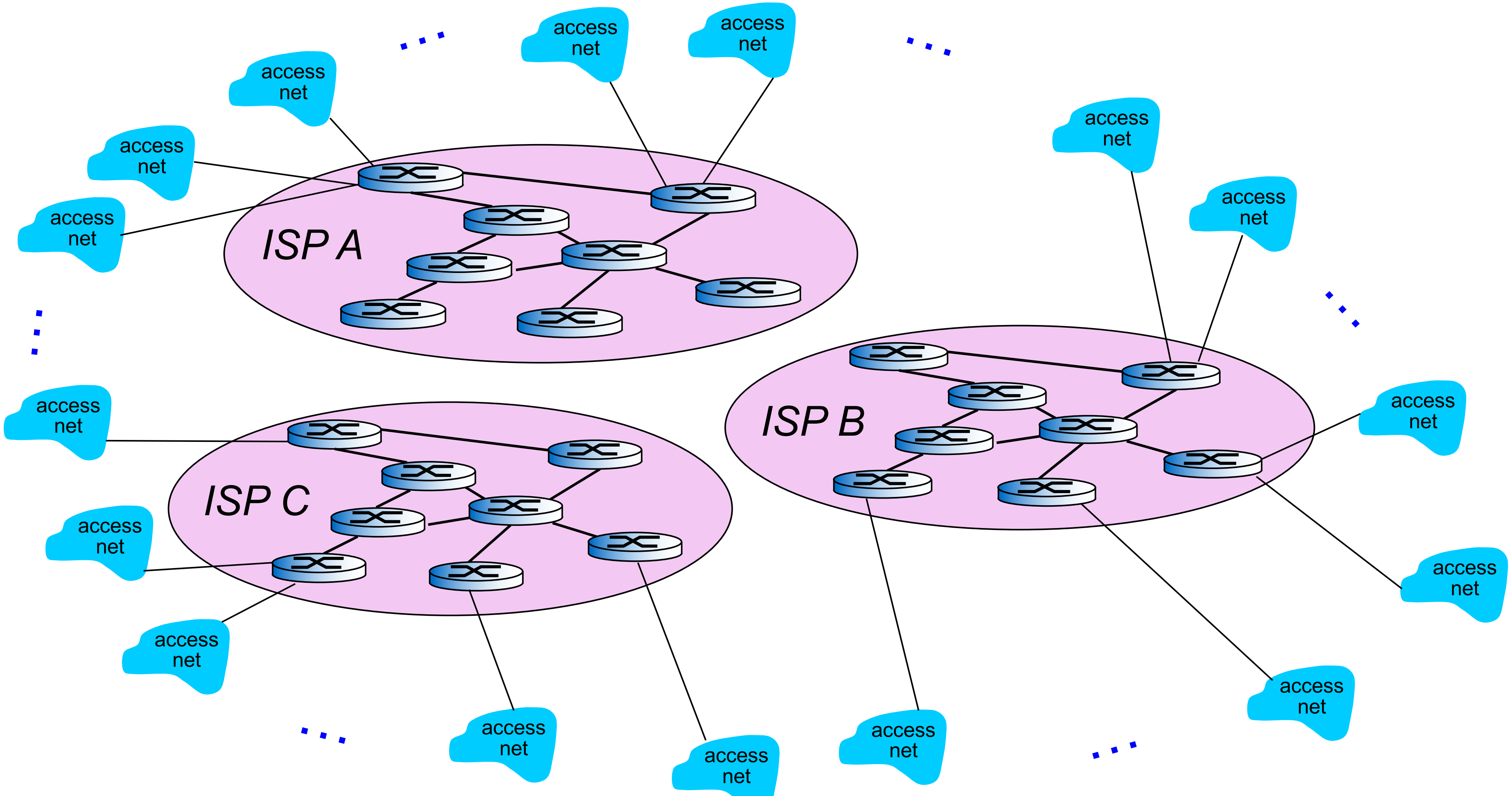
Idea 2: A global "Tier 1" ISP

- An ISP for ISPs, with a few long and fast connections



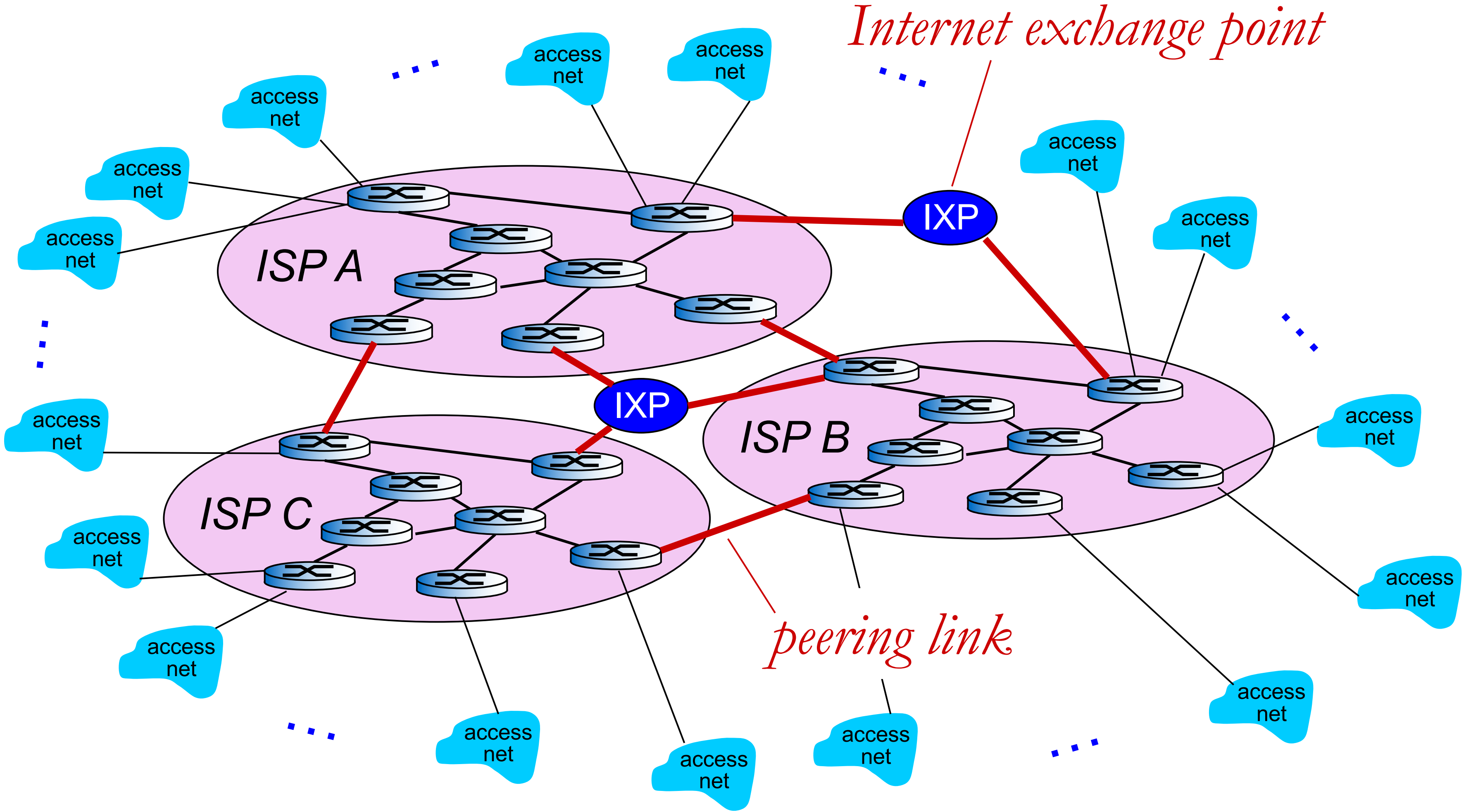
Idea 3: Multiple Tier 1 ISPs

- Multiple Tier 1 ISPs will arise to compete for business and to operate in different continents.



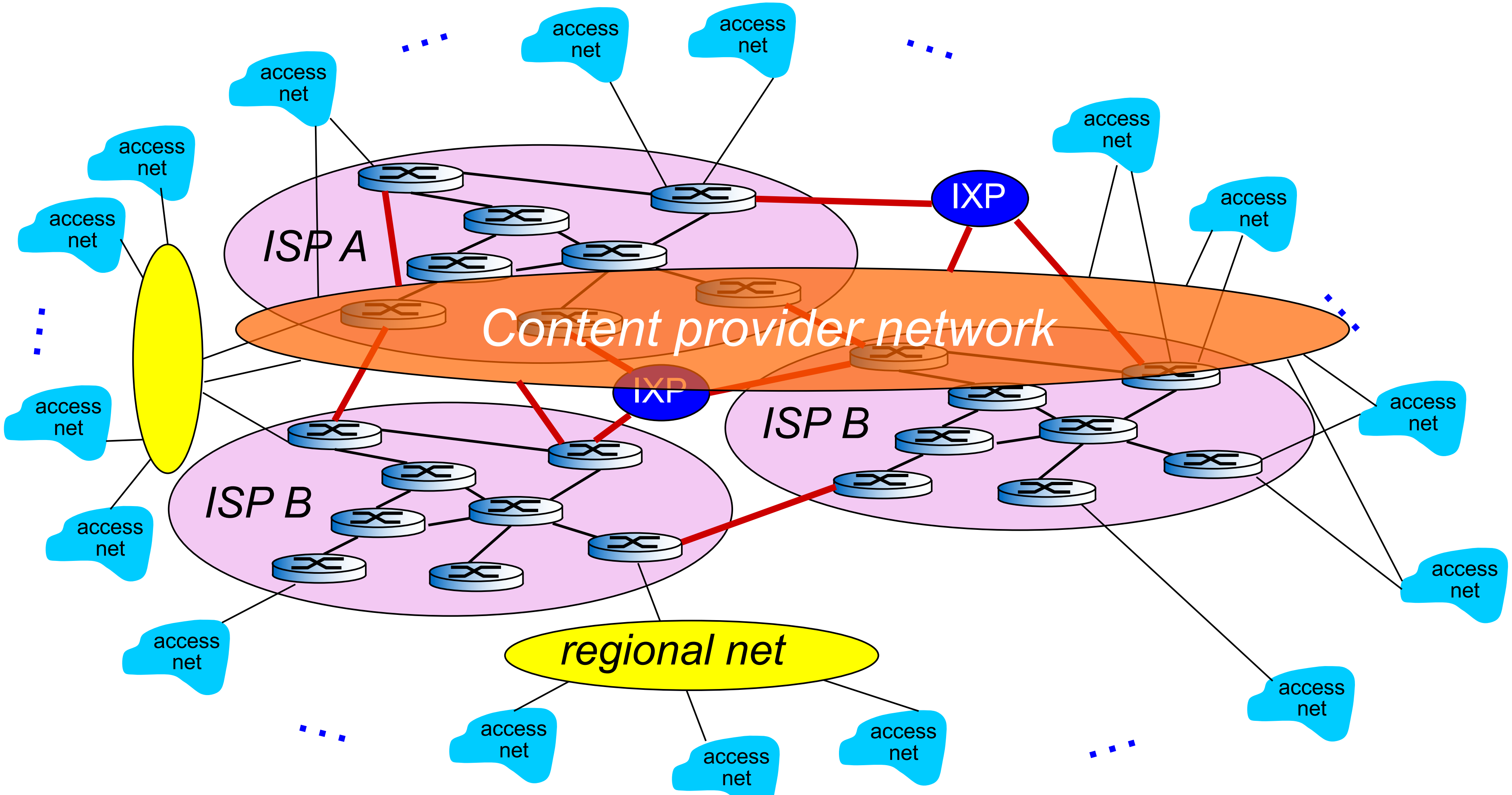
Tier 1 ISPs must be interconnected

- If your network is big & fast enough, you don't pay others for an Internet connection. You *are* the Internet (or at least a big chunk of it).
- ISP interconnection is often "settlement-free" (no money is exchanged).

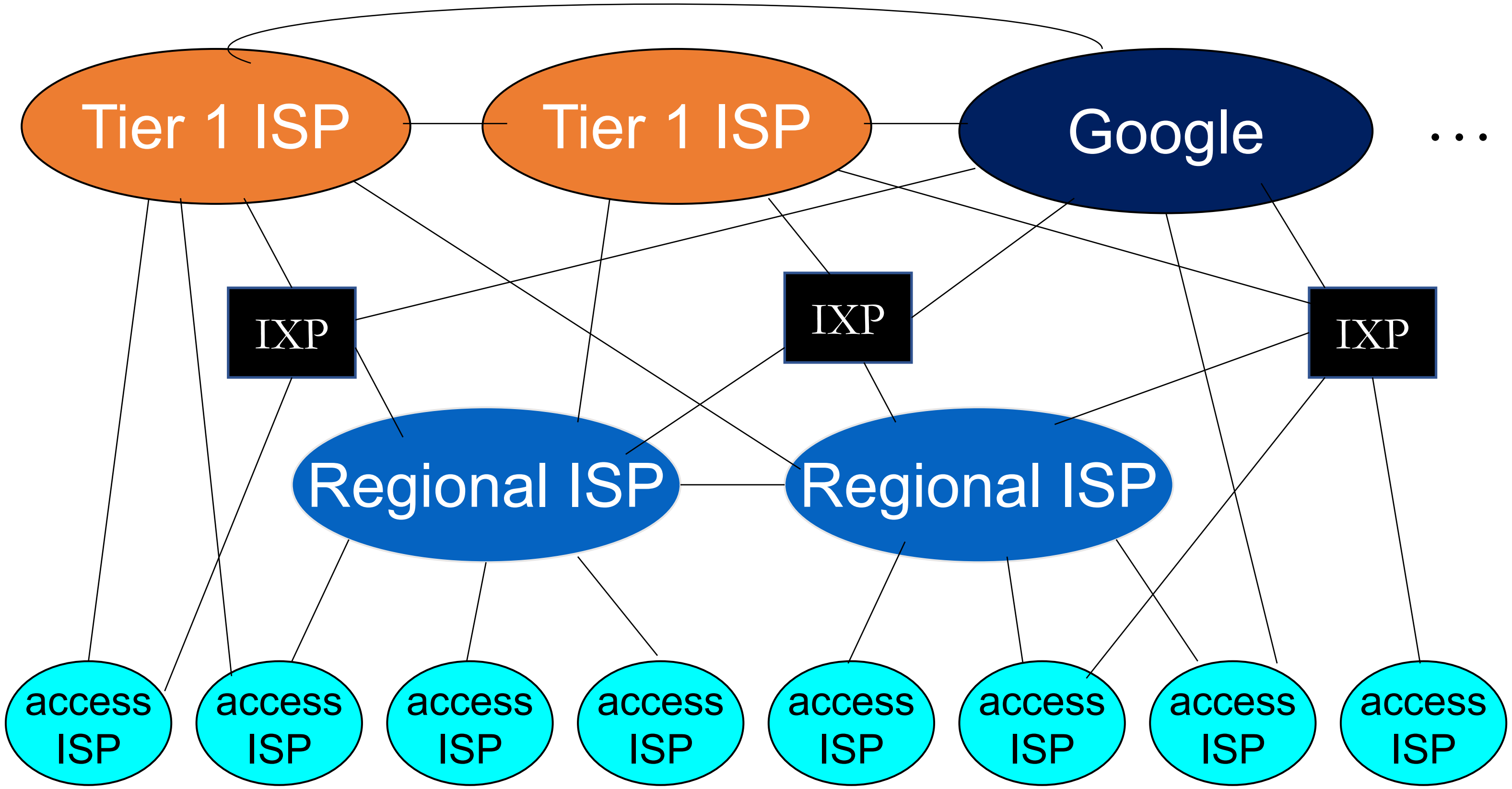


Final version: add regional ISPs and Content Providers

- Google, AWS, Akamai, and other big content providers build their own global networks, much like ISPs, to speed access to their data centers.



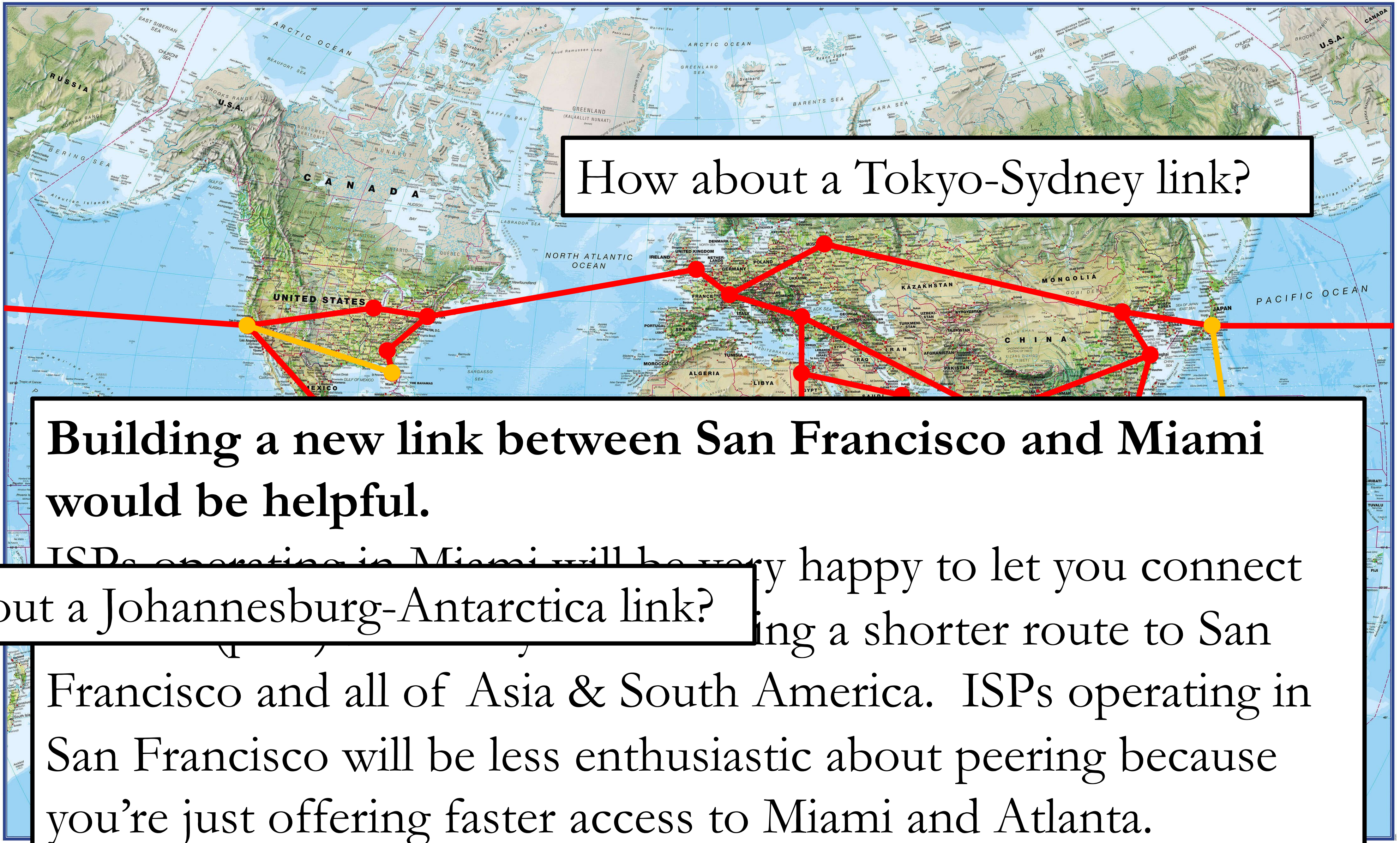
Internet's structure



At the center are a few well-connected, large networks

- Tier-1 commercial ISPs (e.g., Century Link, Verizon, AT&T, China Telecom), national & international coverage
- Content provider networks (e.g, Google, Amazon): private network that connects its distributed data centers and connect to various ISPs.

How the Internet grows and improves



How about a Tokyo-Sydney link?

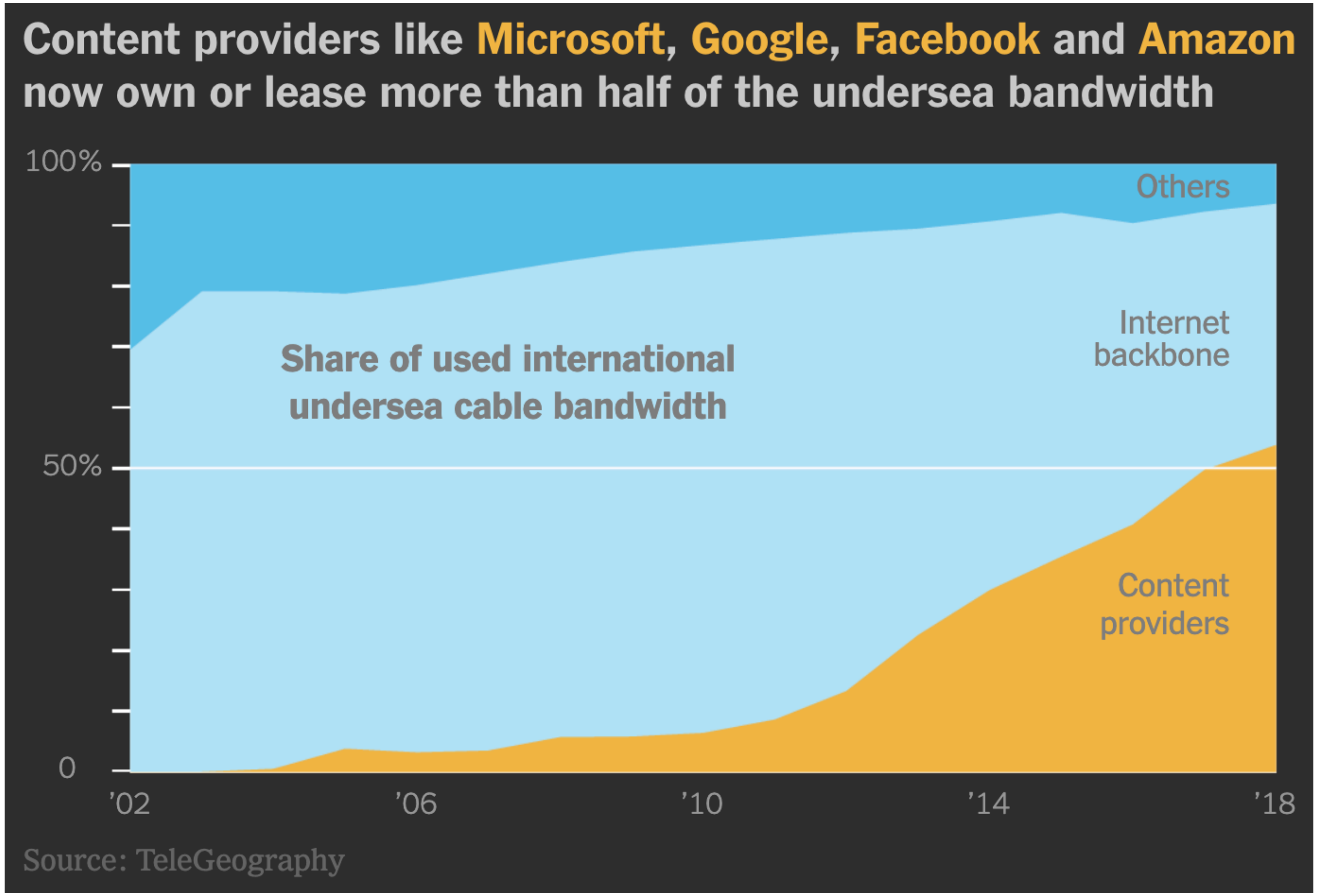
Building a new link between San Francisco and Miami would be helpful.

ISPs operating in Miami will be very happy to let you connect [illegible]ing a shorter route to San Francisco and all of Asia & South America. ISPs operating in San Francisco will be less enthusiastic about peering because you're just offering faster access to Miami and Atlanta.

How about a Johannesburg-Antarctica link?

STOP and THINK

Visualization of undersea Internet cables



Content providers like **Microsoft**, **Google**, **Facebook** and **Amazon** now own or lease more than half of the undersea bandwidth

Source: TeleGeography

- https://www.nytimes.com/interactive/2019/03/10/technology/internet-cables-oceans.html
- https://www.submarinecablemap.com/
- https://www2.telegeography.com/submarine-cable-faqs-frequently-asked-questions

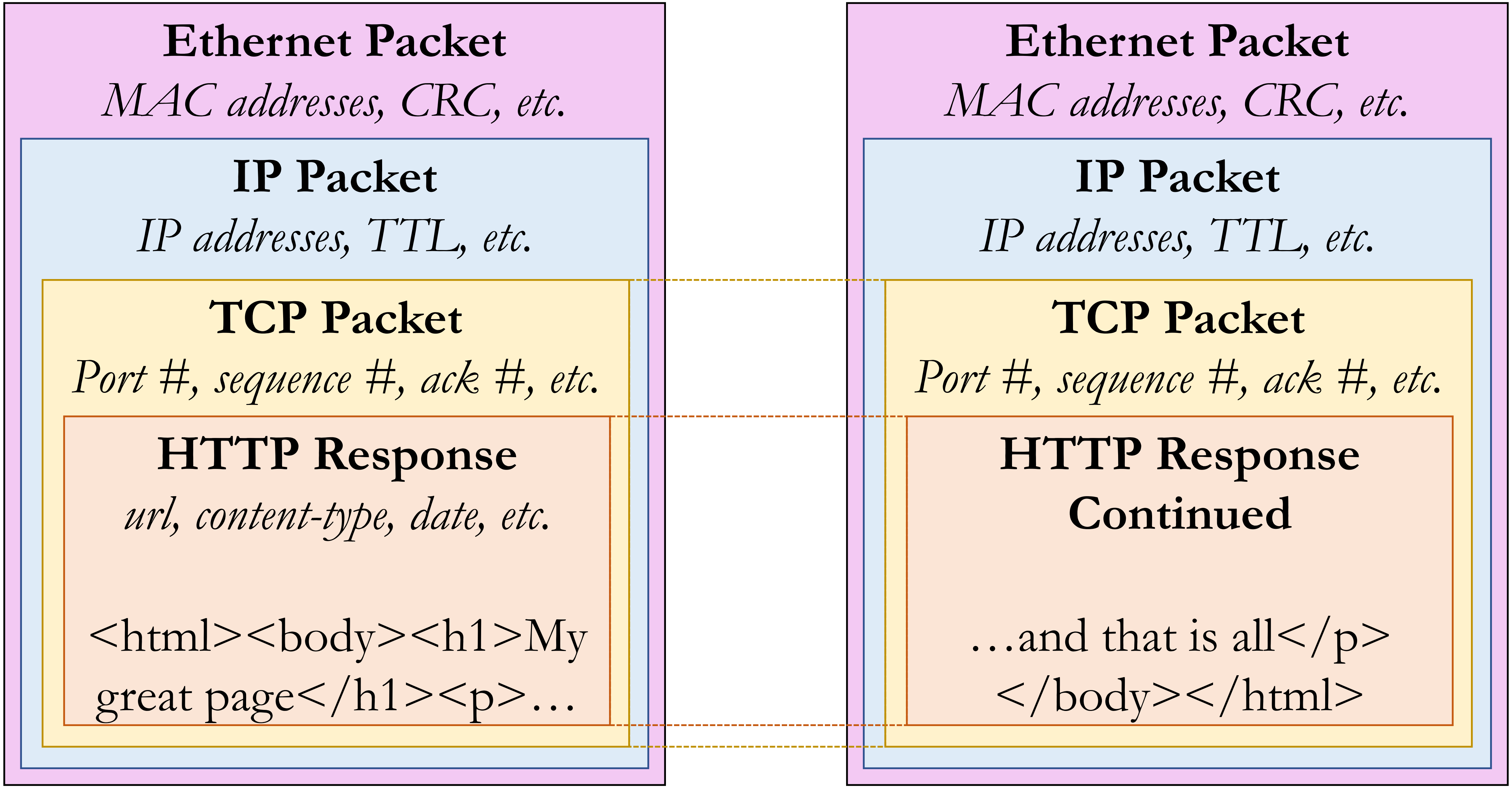
Intermission

Network **layers** typically involved in the WWW

Networking involved many different protocols with different jobs:

- *HTTP* – get web pages, images, etc. and post data to servers
  - Handles URLs, redirects, caching, proxies, cookies
- *TLS* – encrypt traffic (optional, but nowadays almost universal)
- *TCP* – byte streams (ordering, delivery confirmation, pacing)
  - Gives a file-like interface to network connections (read and write byte[])
  - Handles limited packet size, dropped & reordered packets.
- *IP* – forward packets across multiple hops
  - Makes best-effort attempt to route packets to their destination IP address
- *Ethernet/Wifi* – share a communication link with multiple local devices.

Layers add additional **header** bytes to packets


Ethernet Packet
MAC addresses, CRC, etc.
IP Packet
IP addresses, TTL, etc.
TCP Packet
Port #, sequence #, ack #, etc.
HTTP Response
url, content-type, date, etc.
<html><body><h1>My great page</h1><p>…
Ethernet Packet
MAC addresses, CRC, etc.
IP Packet
IP addresses, TTL, etc.
TCP Packet
Port #, sequence #, ack #, etc.
HTTP Response Continued
…and that is all</p></body></html>

Wireshark demo

It’s a tool for observing network traffic.

Sockets

- Each computer on the Internet has a unique IP address.
- However, a machine may have many processes running, each with its own, separate network connections.
    - How do we know if an arriving packet is for Chrome or for WhatsApp?
- Each TCP or UDP packet has a 16-bit *port number*
- The OS manages ports: only one process may "listen" on a given port.
- Publicly-reachable services (applications) run on standard ports:
    - HTTP uses port 80
    - HTTPS (with TLS encryption) uses port 443
    - SMTP (email transport) uses port 25
    - DNS uses port 53

Can we run multiple web servers on one machine?

- A webserver is a process that responds to HTTP requests.
- Normally we cannot run more than one, because HTTP uses port 80, and the OS will only allow one process to listen to port 80.
- However, if necessary, we can run multiple webservers on a non-standard ports, and tell clients of the special port number. Eg.:
  - http://murphy.wot.eecs.northwestern.edu (connects to standard port 80)
  - http://murphy.wot.eecs.northwestern.edu:8000 (connects to port 8000)
  - http://murphy.wot.eecs.northwestern.edu:8001 (another student can run this)
  - http://murphy.wot.eecs.northwestern.edu:8002 (yet another process here)
- Note that HTTPS uses a different port:
  - https://murphy.wot.eecs.northwestern.edu (connects to standard port **443**)

How do we run multiple web browsers on one host?

- Let’s say we have opened multiple tabs in Chrome and Firefox.
- How does a given HTTP response get directed to the correct tab?
- Client can use different *source* ports for each outbound connection:

Outbound:

Client	Source IP	Source Port	Destination IP	Destination Port
Chrome, Tab 1	100.0.0.10	12345	129.105.8.237	80
Chrome, Tab 2	100.0.0.10	34209	129.105.8.237	80
Firefox, Tab 1	100.0.0.10	39009	129.105.8.237	80
Firefox Tab 2	100.0.0.10	10002	172.217.7.196	80

Inbound:

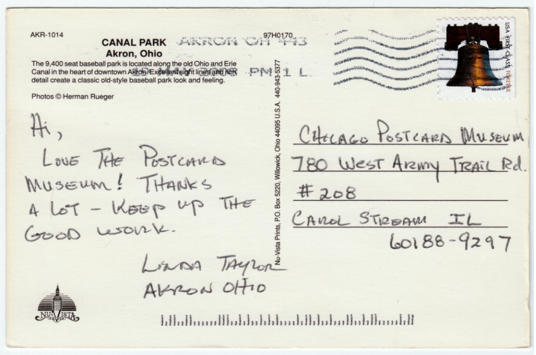
Remote IP	Remote Port	Destination IP	Destination Port	Recipient
172.217.7.196	80	100.0.0.10	10002	Firefox Tab 2
129.105.8.237	80	100.0.0.10	34209	Chrome, Tab 2
129.105.8.237	80	100.0.0.10	12345	Chrome, Tab 1
172.217.7.196	80	100.0.0.10	21111	**dropped!**

STOP and THINK

Privileged ports

- Ports below number 1024 are *privileged* and usually the OS will require root or administrative credentials to use these.
- The standard Internet applications use low-numbered ports by default
- This restriction prevents normal users on a machine from running "official" web or mail servers on that host.
- Sometimes unusual ports are used to "hide" configuration webpages for networked devices like routers, modems, media appliances, etc.

The Internet is a packet switched network (sending "postcards" of data). TCP creates the illusion of a bidirectional, reliable "pipe" for data.



AKR-1014

CANAL PARK
Akron, Ohio

97H0170

AKRON OH 443

PM 1 L

The 9,400 seat baseball park is located along the old Ohio and Erie Canal in the heart of downtown Akron. Excellent sight lines and rich detail create a classic old-style baseball park look and feeling.

Photos © Herman Rueger

Nu-Vista Prints, P.O. Box 5220, Willowick, Ohio 44095 U.S.A. 440-943-5377

USA FIRST-CLASS FOREVER

Hi,
LOVE THE POSTCARD
MUSEUM! THANKS
A LOT - KEEP UP THE
GOOD WORK.

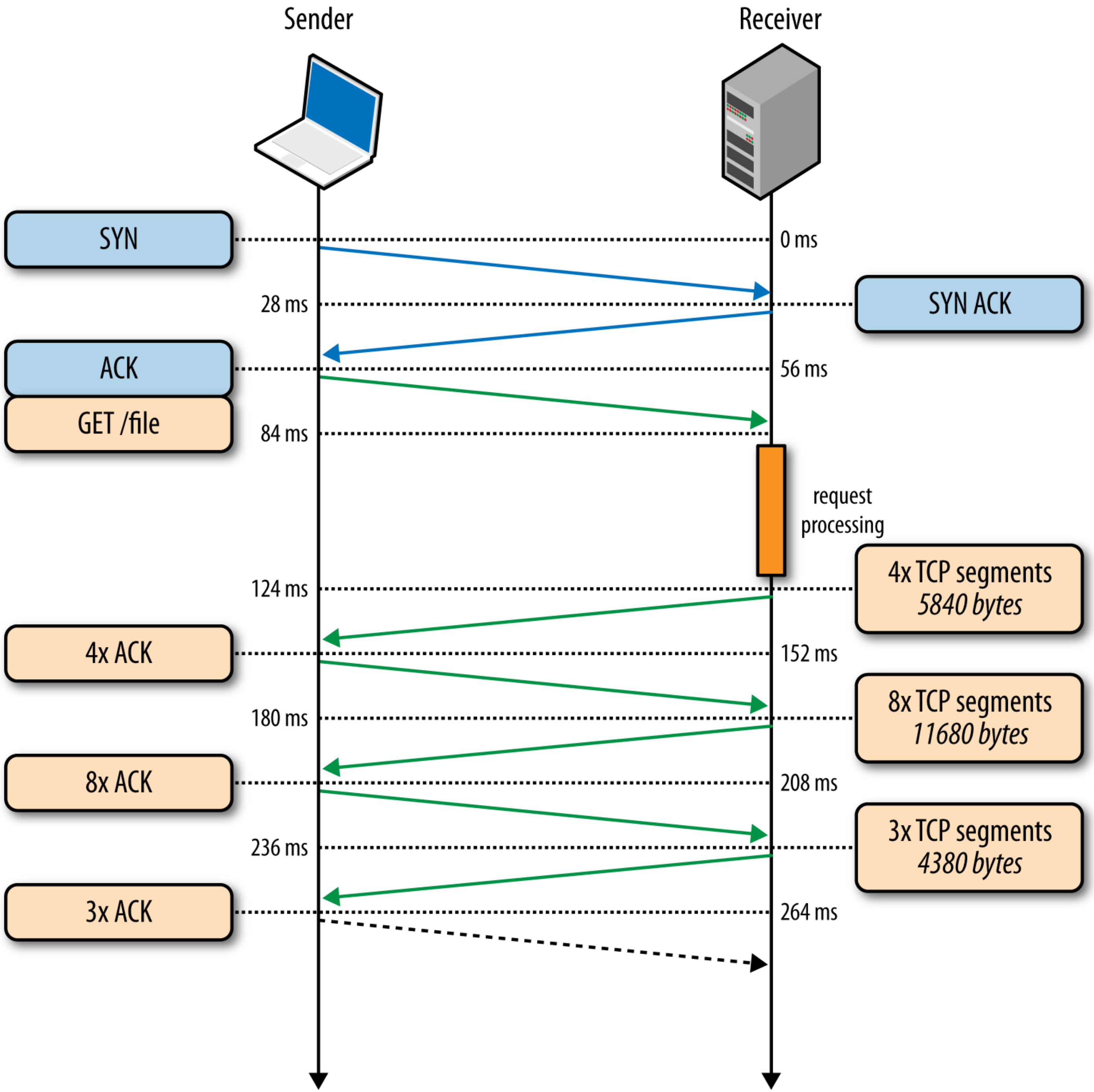
LINDA TAYLOR
AKRON OHIO

CHICAGO POSTCARD MUSEUM
780 WEST ARMY TRAIL Rd.
#208
CAROL STREAM IL
60188-9297

→

TCP streams

- TCP provides applications with a file-like abstraction for a network connection.
  - Just *read* and *write* to the connection. It's like a pipe.
- Packets are reassembled and ordered automatically by the OS/library.
- Must establish a connection first, using a 3-way handshake.
- We'll talk about TCP details later in the class



TCP overview

Reliable transport between sending and receiving processes

Provides:

- *Port numbers* so a computer can have multiple network connections
- *Message fragmentation & reassembly*
- *Delivery confirmation & retransmission*

} Illusion of a reliable queue/"pipe"

- *Flow control:* sender won't overwhelm receiver
- *Congestion control:* throttle sender when network is overloaded

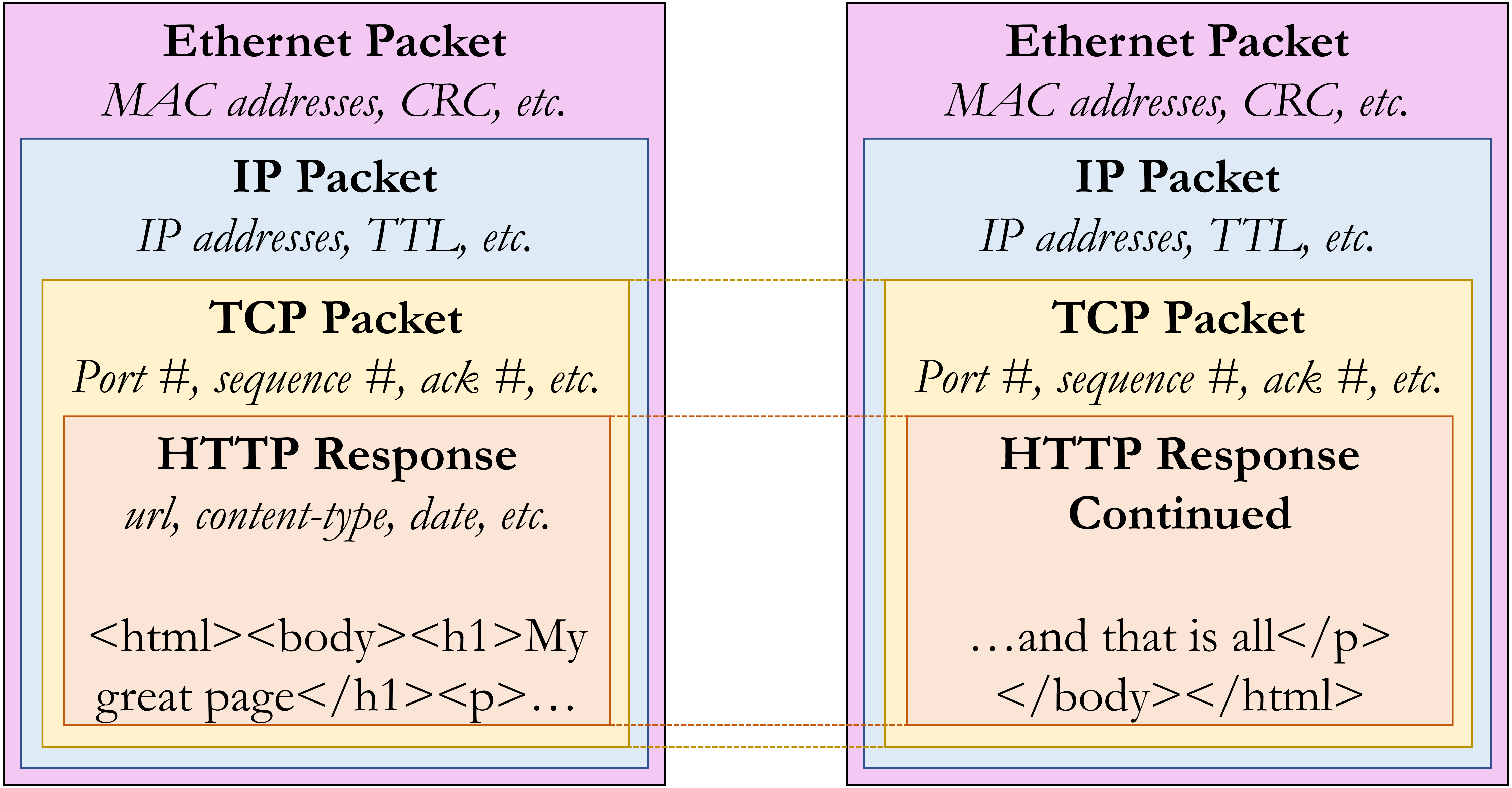
Disadvantages:

- Must first setup a connection between client and server processes
- No timing or minimum throughput guarantees
- Lacks encryption

UDP is a simpler alternative to TCP

- For applications that don’t need a reliable connection.
- Avoids the setup delays associated with TCP handshake
- Protocols built on UDP must deal with more error conditions.
- Like plain IP packets, but adds **port numbers**
- Examples:
  - Session Initiation Protocol (SIP) for setting up VoIP calls.
  - Realtime Transport Protocol (RTP) for sending audio in VoIP call
- We’ll talk more about UDP and TCP in detail in Chapter 3.

Layering review



Ethernet Packet

MAC addresses, CRC, etc.

IP Packet

IP addresses, TTL, etc.

TCP Packet

Port #, sequence #, ack #, etc.

HTTP Response

url, content-type, date, etc.

<html><body><h1>My great page</h1><p>…

Ethernet Packet

MAC addresses, CRC, etc.

IP Packet

IP addresses, TTL, etc.

TCP Packet

Port #, sequence #, ack #, etc.

HTTP Response Continued

…and that is all</p></body></html>

Recap – Introduction to the Internet

- Packets travel along many *hops* to reach the intended destination
  - Each router has a fixed-size queue; packets are dropped if full
  - Packet is also dropped if a bit-flip error is detected
- Showed four different sources of *packet delay* at each hop:
  - Nodal processing, queueing (associated with the router)
  - Transmission, propagation (associated with the link)
- Internet is a "network of networks"
  - Tier 1 ISPs and big content providers build high-speed *backbone* links.
  - *Peering* is when networks connect to each other without any payment.
- Networks use layered protocols, eg.: Ethernet, IP, TCP, TLS, HTTP
- Socket is a software abstraction of a network connection (TCP or UDP)
  - It's one end of a pipe: you can send data in or get data out
  - Each socket is bound to a particular *port* number. Port number determines which process on a host is responsible for handling a given packet.