50.005 (Spring 2016)

Networking Basics

Objectives: What is a network or internetwork? What are the key challenges in designing a large-scale global internetwork? What are some basic design principles that can help overcome the challenges?

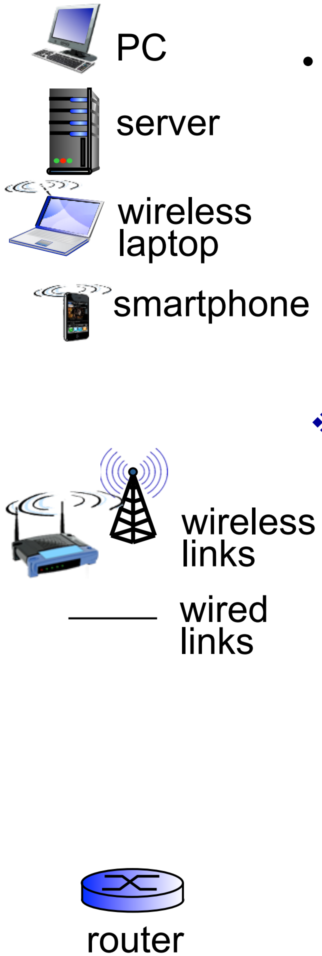
NS1: March 14, 2016

Textbook (**K&R**): Sections 1.1, 1.3, 1.5

What is a network (Internet)?

- We all know …
  - We use the network everyday
  - We're using it *now*
  - All our favorite gadgets are *online*!
  - The Internet means: my iPhone, my Powerbook, the web, my email, WhatsApp, FB, Twitter, …
- But no authoritative formal answer …
  - Depends on who you ask, what's the purpose

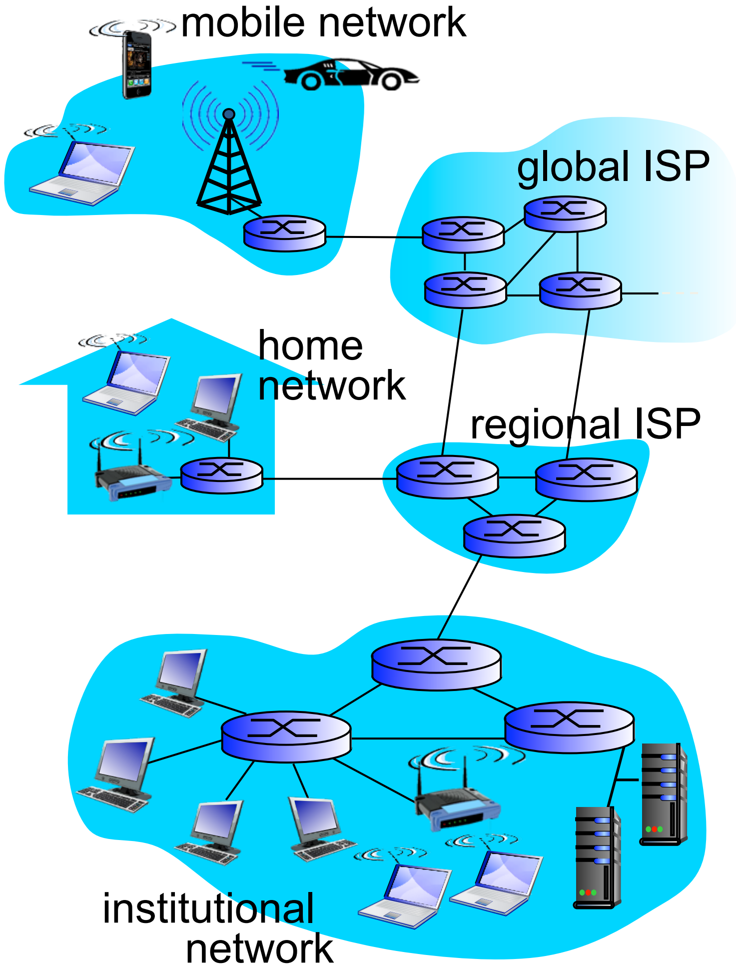
What's the Internet: "nuts and bolts" view



- 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*



"Fun" internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

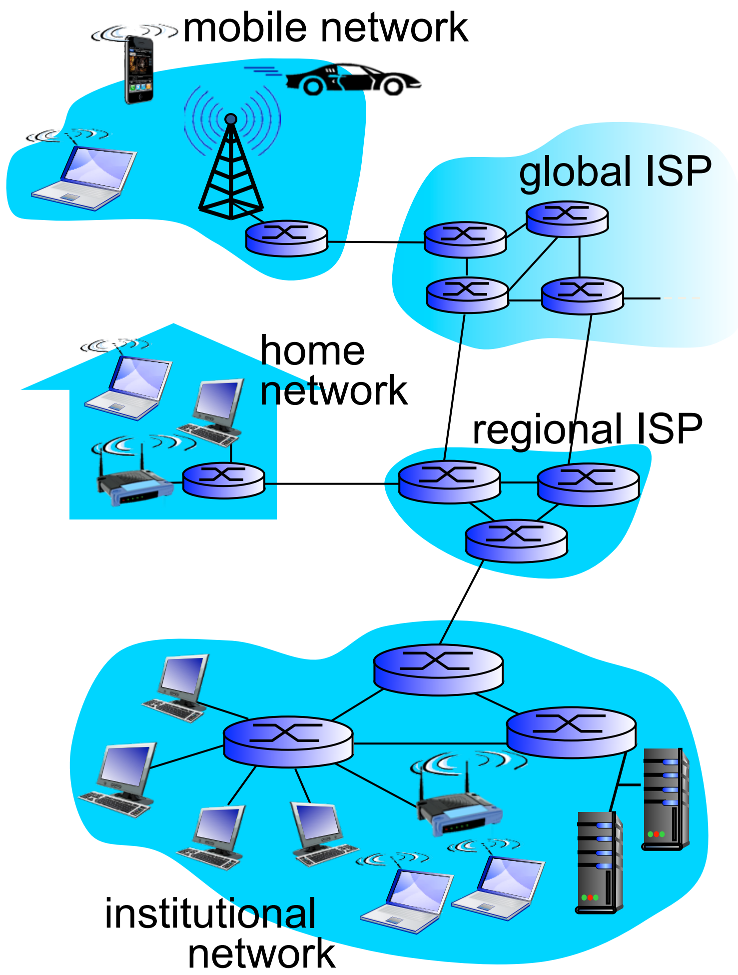
Internet
refrigerator

Slingbox: watch,
control cable TV remotely

Internet phones

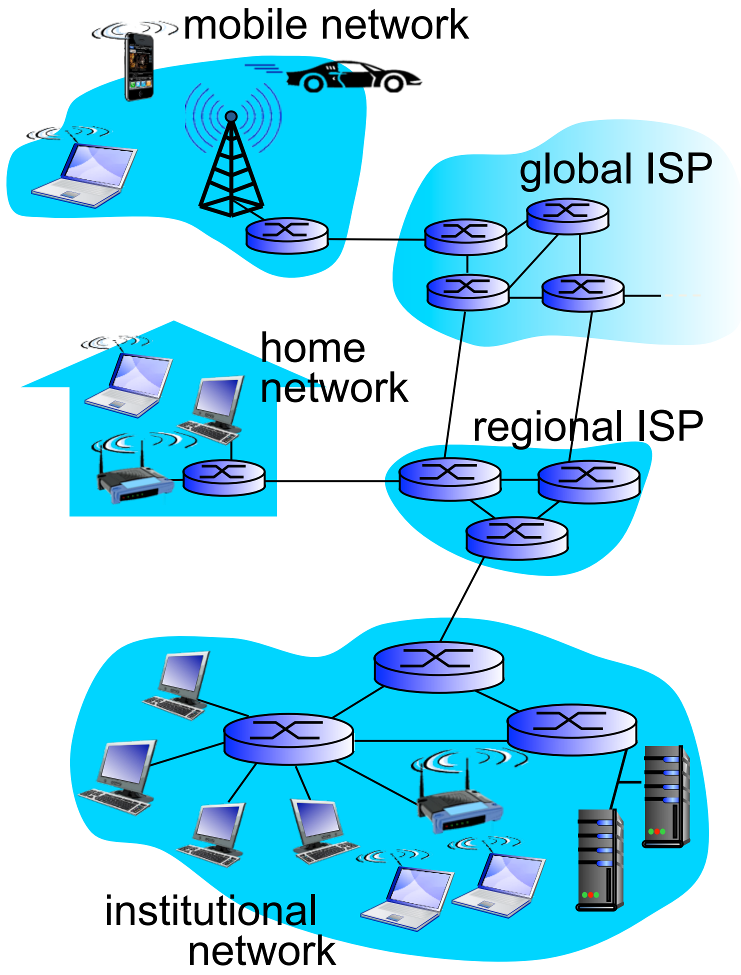
What's the Internet: "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* control sending, receiving of messages
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards (open, free)*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's 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
  - provides service options, analogous to postal service
  - *Socket* API



Let's build an Internet!

- One that connects billions of people, and many times more devices?
- Wow, it's so difficult! Well …
- Know your enemy
  - What are the basic *challenges*?
- Know your weapons
  - What are the design *principles* to overcome the challenges?

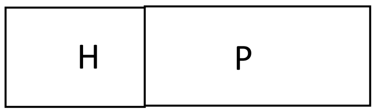
Challenge 0: Interoperability

- How to make everything talk? They're all different!
- How do we (you and I) do it?
  - Fix the language (e.g., English)
  - Fix some conventions, etiquette

- Now make it more formal & precise
  - Standardization (IETF/RFC) into communication *protocols*
  - HTTP, TCP, UDP, IP, ARP, whatever-P
- Each "P" deals with unit of data called *packet*

H - header (interpreted)

P – payload (uninterpreted)



Principle of *encapsulation* (info hiding)

What's 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

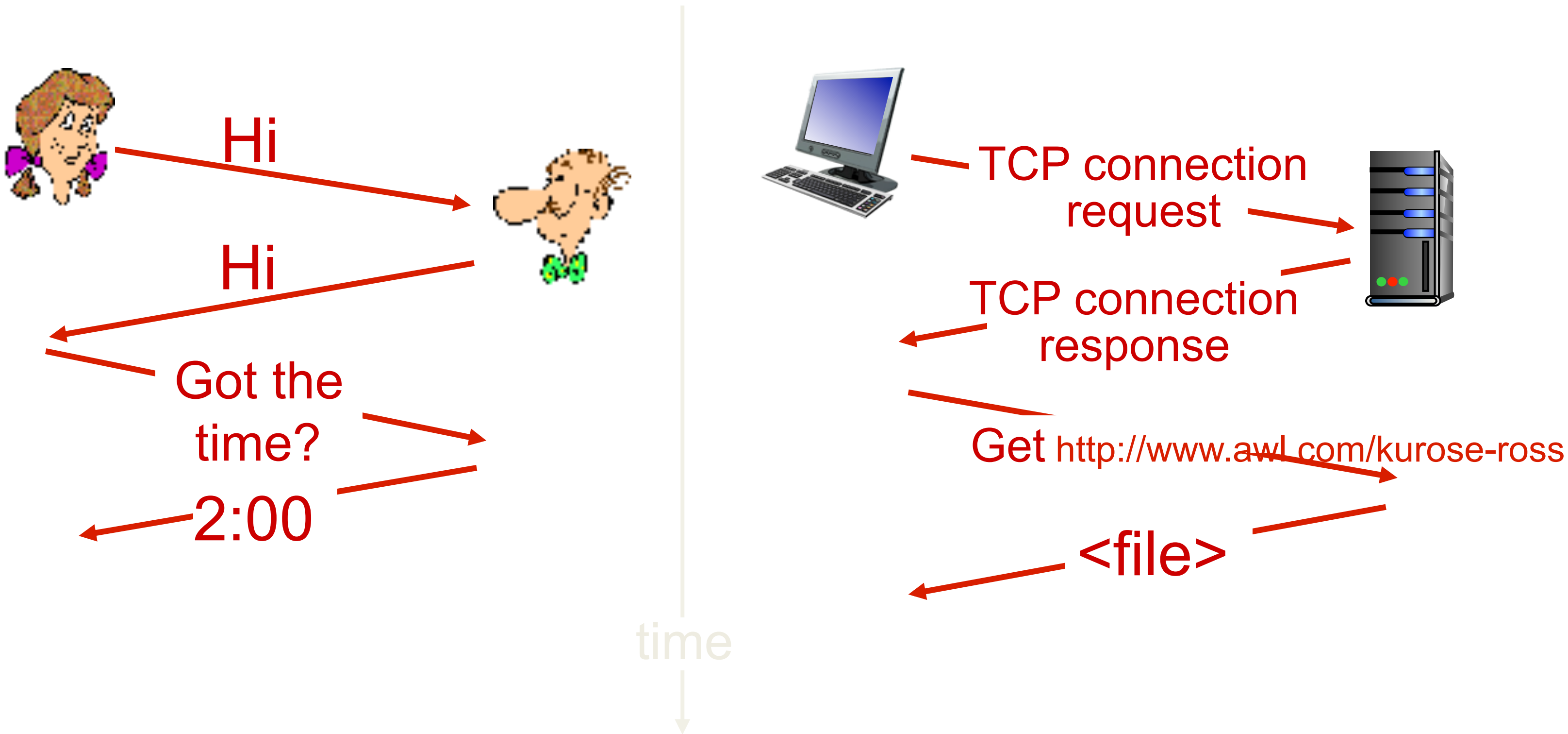
network protocols:

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

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

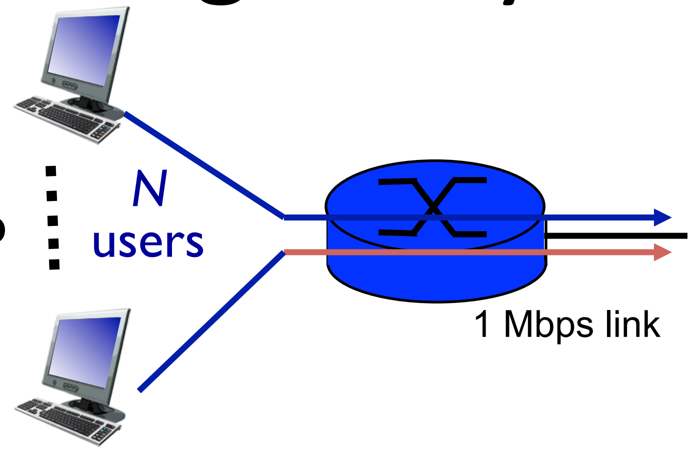
What's a protocol?

a human protocol and a computer network protocol:



Q: other human protocols?

Challenge 1: Sharing among many



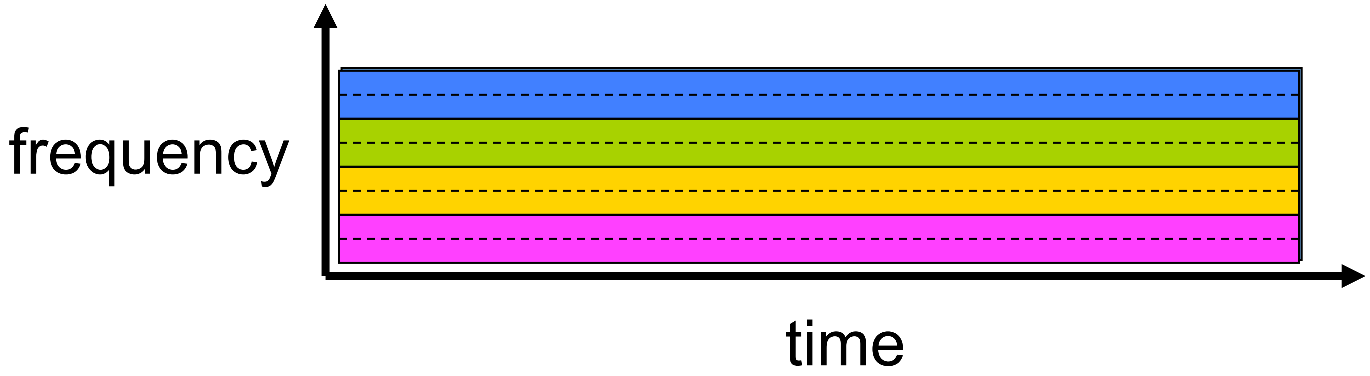
- Internet is a *shared* resource
- How do you share a function room?
  - Book a time and own that time slot
  - TDM! (time division multiplexing)
- TDM works well for some applications
  - Phone calls: smooth and same bit rate all the time
  - Simple: Allocate fixed rate to each established call!
- But, data traffic is *bursty*
  - How do you surf? Look + think -> click a link -> look + think ...
    (net silent) (net busy)
- Packet magic: *packet switching* & *statistical multiplexing*

Circuit switching: FDM and TDM

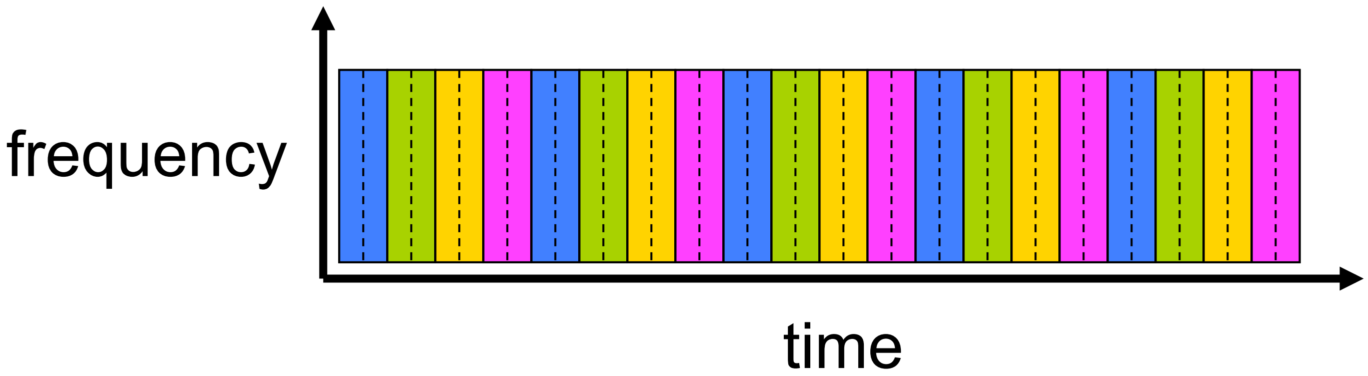
Example:

4 users

FDM



TDM



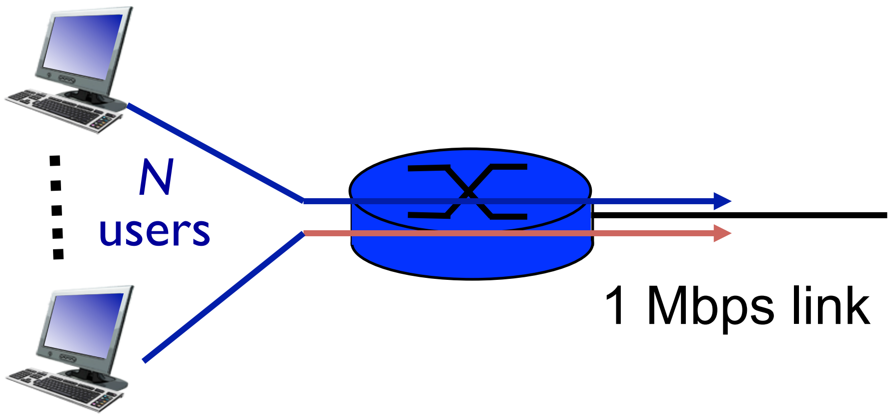
Circuit switching (vs. packet switching) isn't flexible: blue can't use what green doesn't use

Statistical multiplexing w/ packet switching

Why? Bursty data

Example:

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time



- *circuit-switching:* Fixed, dedicated fraction of link
  - 10 users flat
- *packet switching:* Packet occupies link on demand
  - How many?

Activity 1.1

- Can I get away with 35 users?

- Prob. first 11 of them active at the same time?
- Prob. any 11 of them active at the same time?
- Prob. > 10 of them active at the same time?
  - Formula: CDF of binomial distribution
  - Exact answer: write a program (homework), or

    http://www.danielsoper.com/statcalc3/calc.aspx?id=71

Challenge 2: Complex interacting components

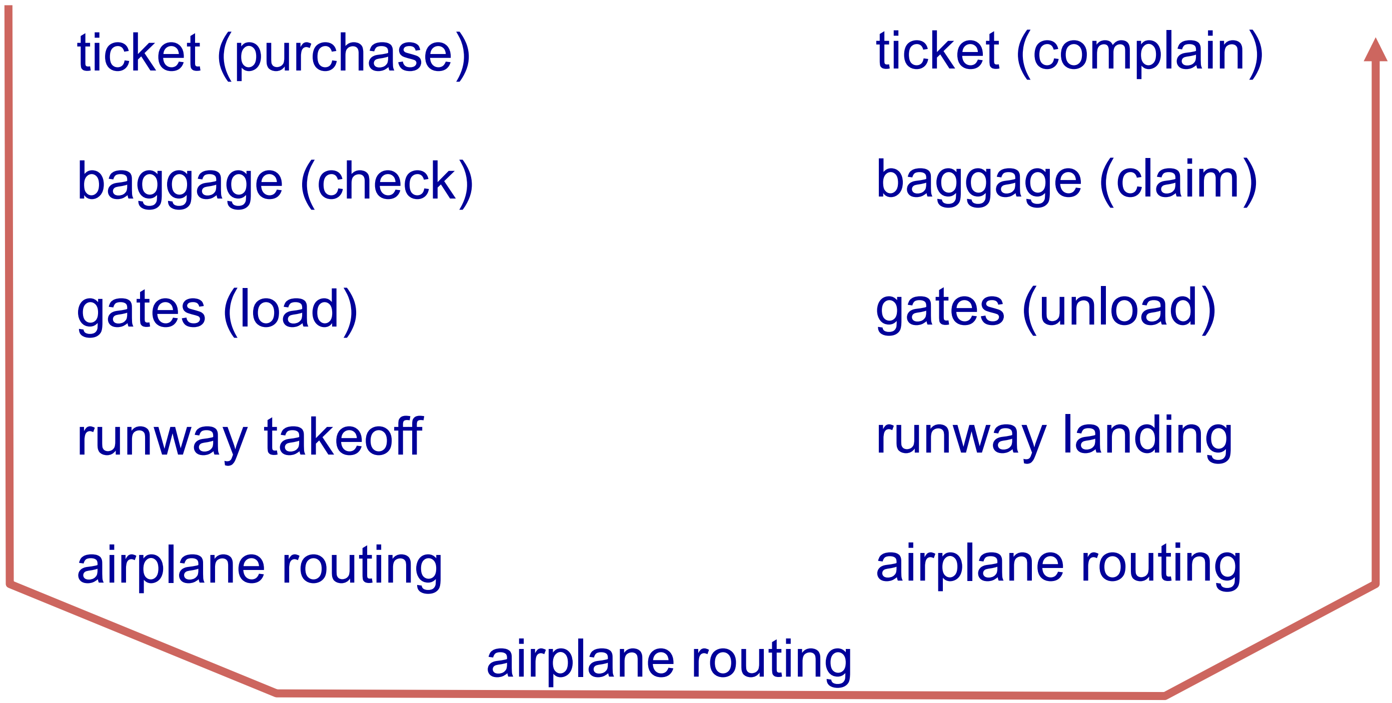
- Network needs many "pieces": naming, routing, reliability, (lots of) applications, …
- Modularity: divide-&-conquer complexity
  - Think functions, processes (sep. of interface from implementation; easy to maintain/update)
- But not enough
  - Also interactions between modules
  - *Emergent behavior* gives (not-so-nice) surprises
- Example of emergent behavior (OS revisited)
  - Priority scheduling + locking = *priority inversion*

Activity #1.2: Priority Inversion

- 3 processes: A of priority 1, B of 10, C of 100 (static priority: higher runs first)
- A and C access same critical section S guarded by lock L; B doesn't syncrhonize w/ A or C
- Consider A and C only: If A runs first; then C runs and now wants to enter S, how long C has to wait at most?

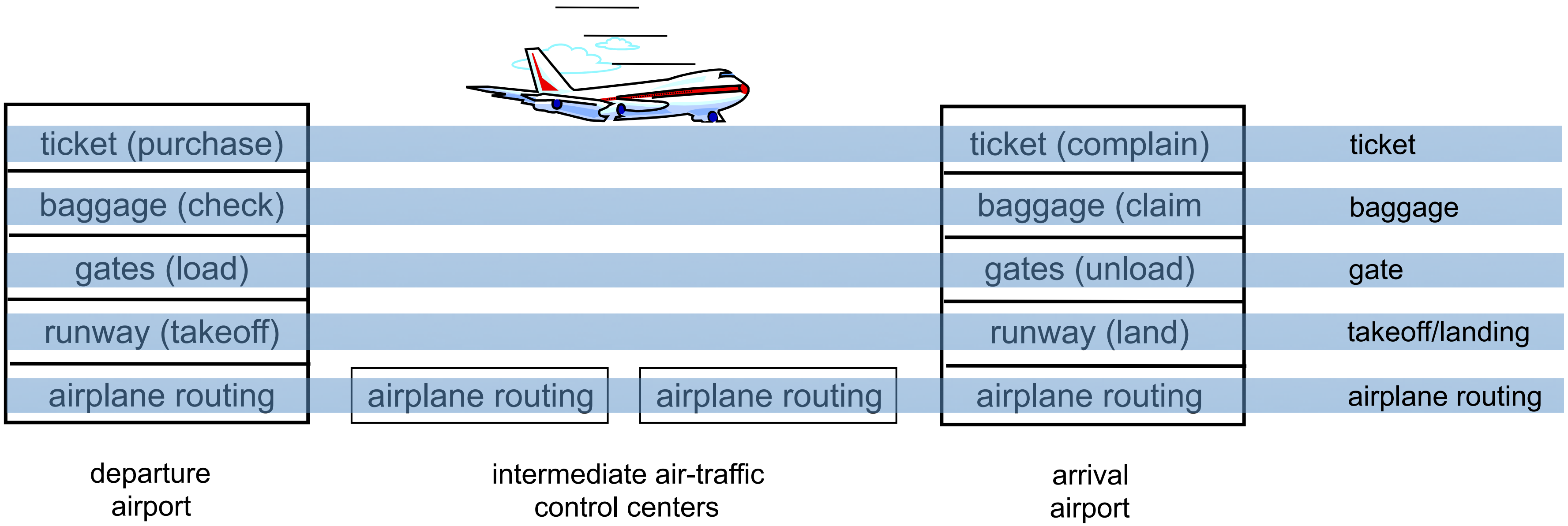
- Now consider A, B, C: If A runs first, followed by B, followed by C, and C now wants to enter S, how long might C have to wait (worst-case scenario)?

Layering: reducing interactions



Organization of air travel: a series of steps

Layering of airline functionality



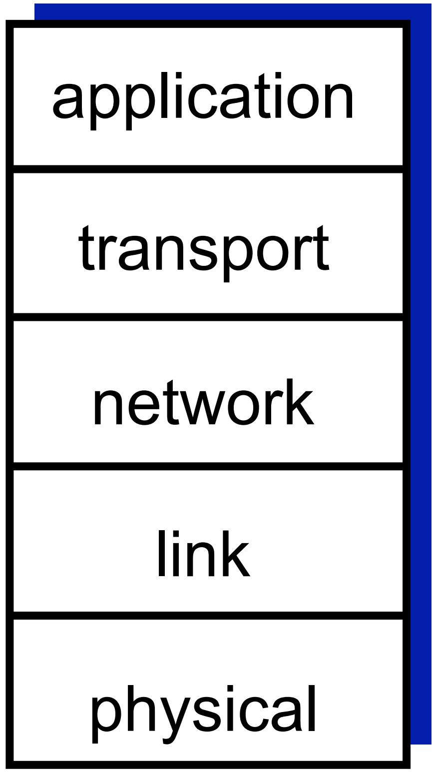
Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below
- in turn provides services to layer above

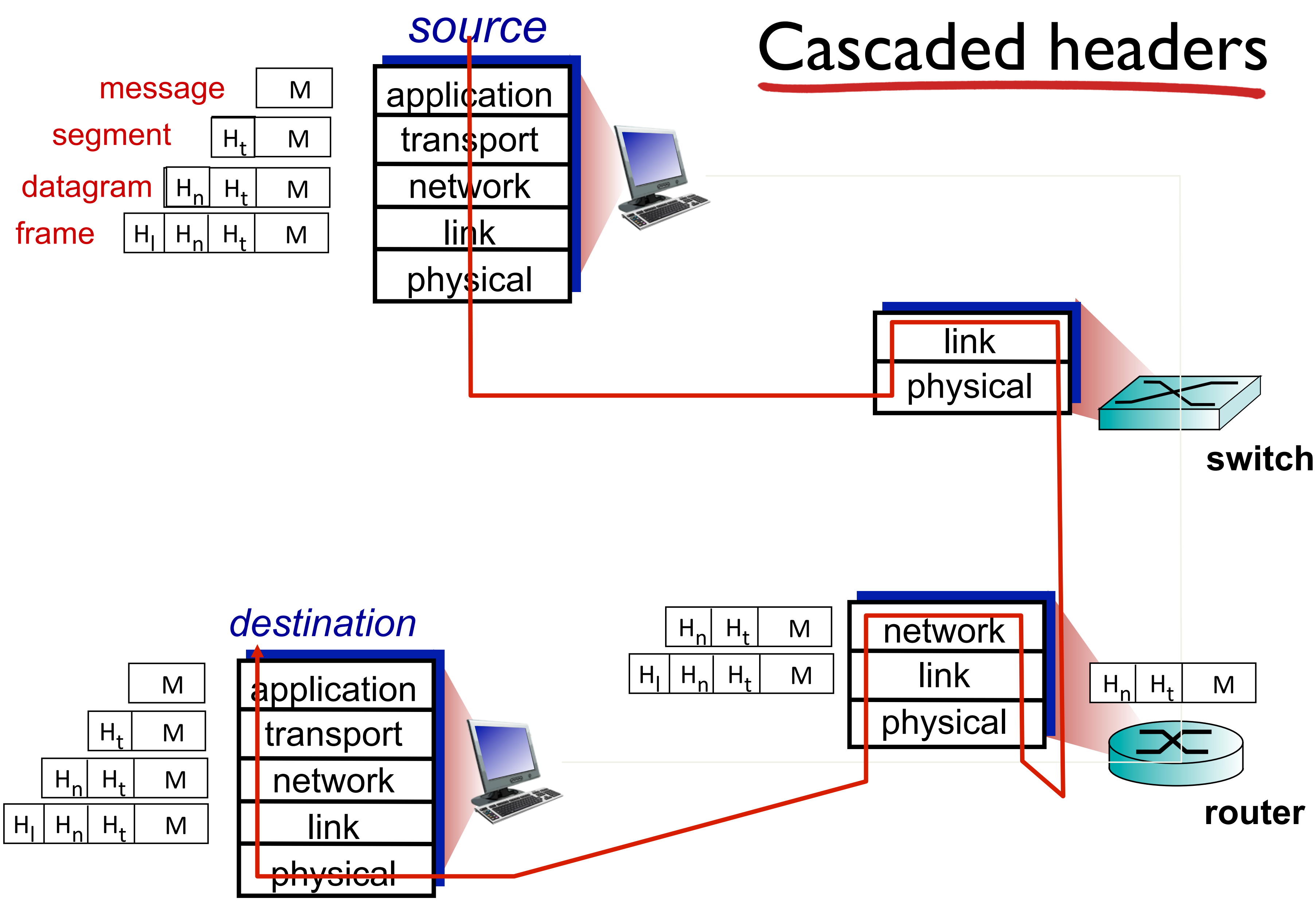
Question: N modules; how many possible interactions if they interact freely? How about with layering?

Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *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
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits "on the wire"



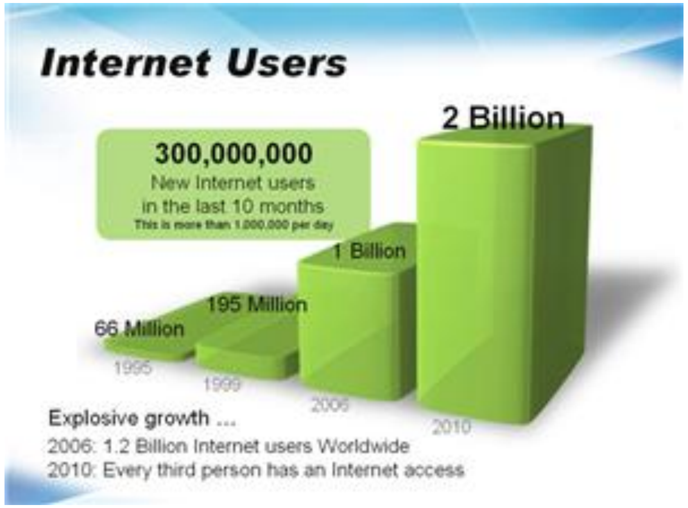
Cascaded headers

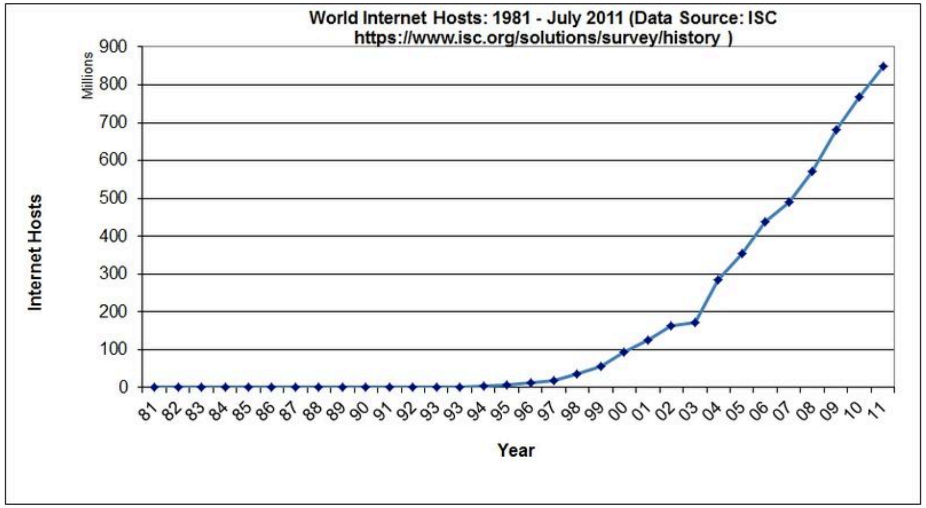


Challenge 3: Scalability

Bigger & bigger size of Internet

- Sustained fast growth
- Increasing structural complexity
- *Incommensurate scaling!*





Can you build bigger and bigger pyramids?

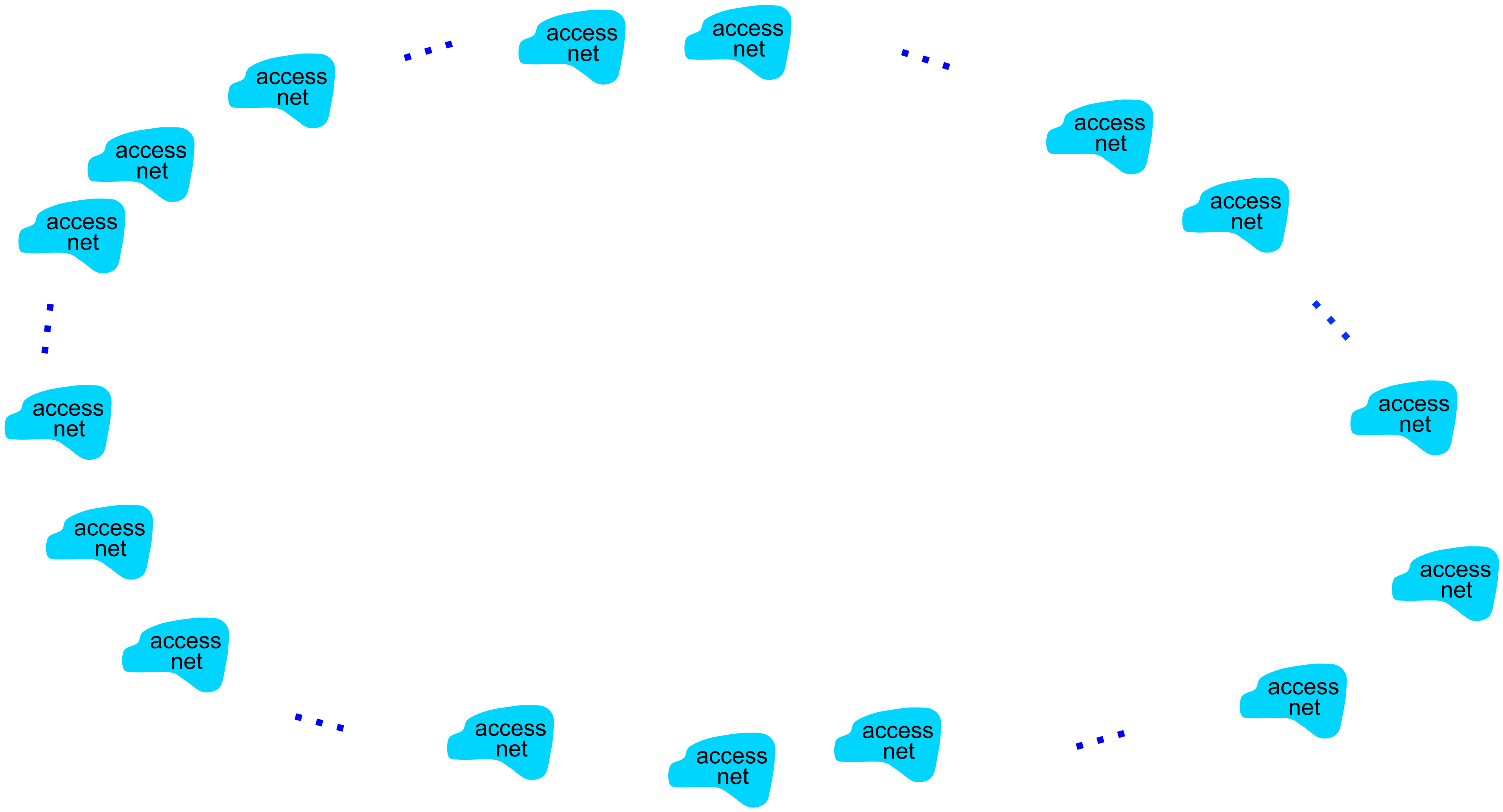
Designing for scalability

Use design principle of *hierarchy* to manage large size and structure

- Network of networks (recursive interconnect)
- Access, regional, national, global ISPs

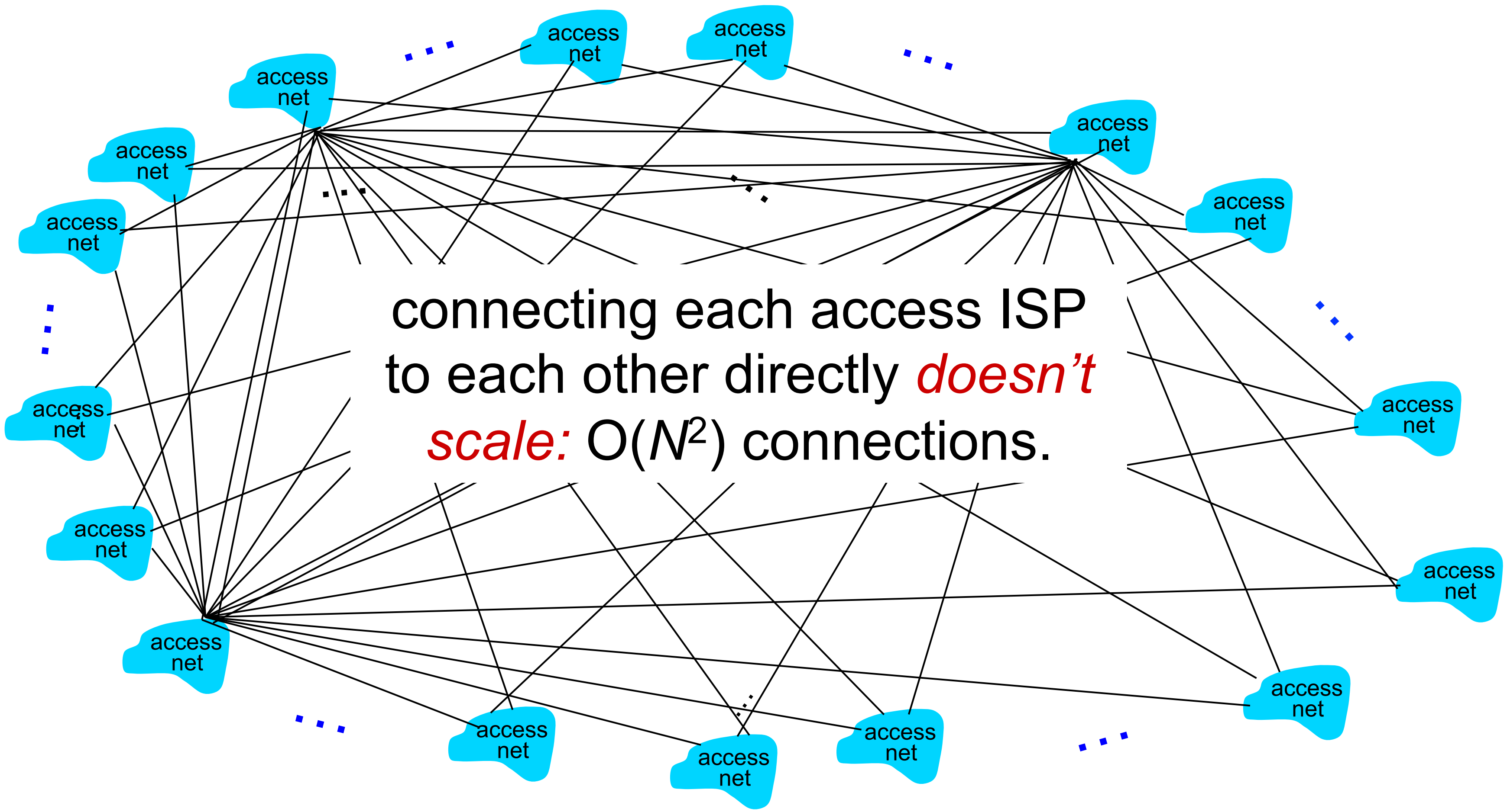
Internet structure: network of networks

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



Internet structure: network of networks

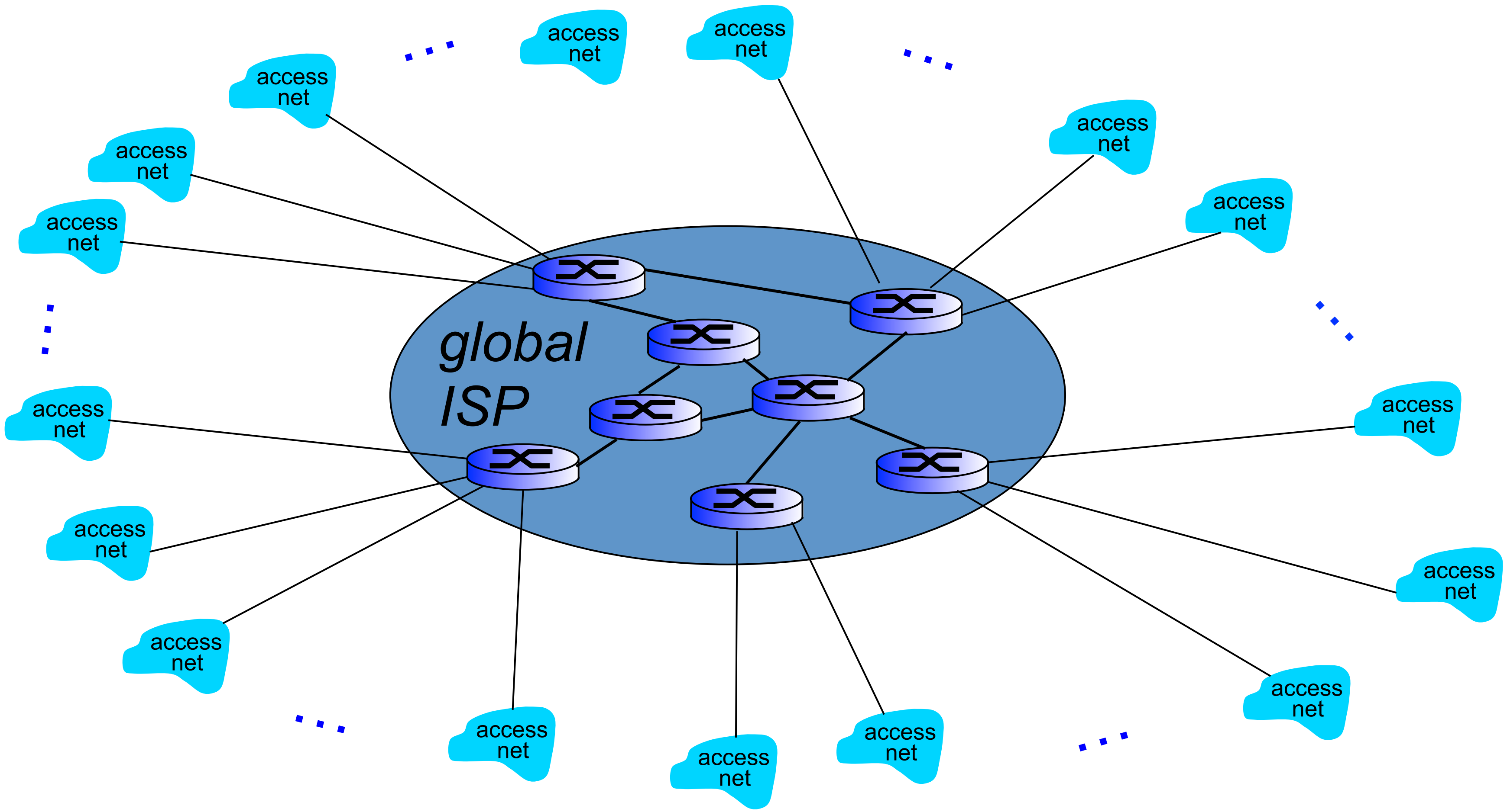
Option: connect each access ISP to every other access ISP?



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

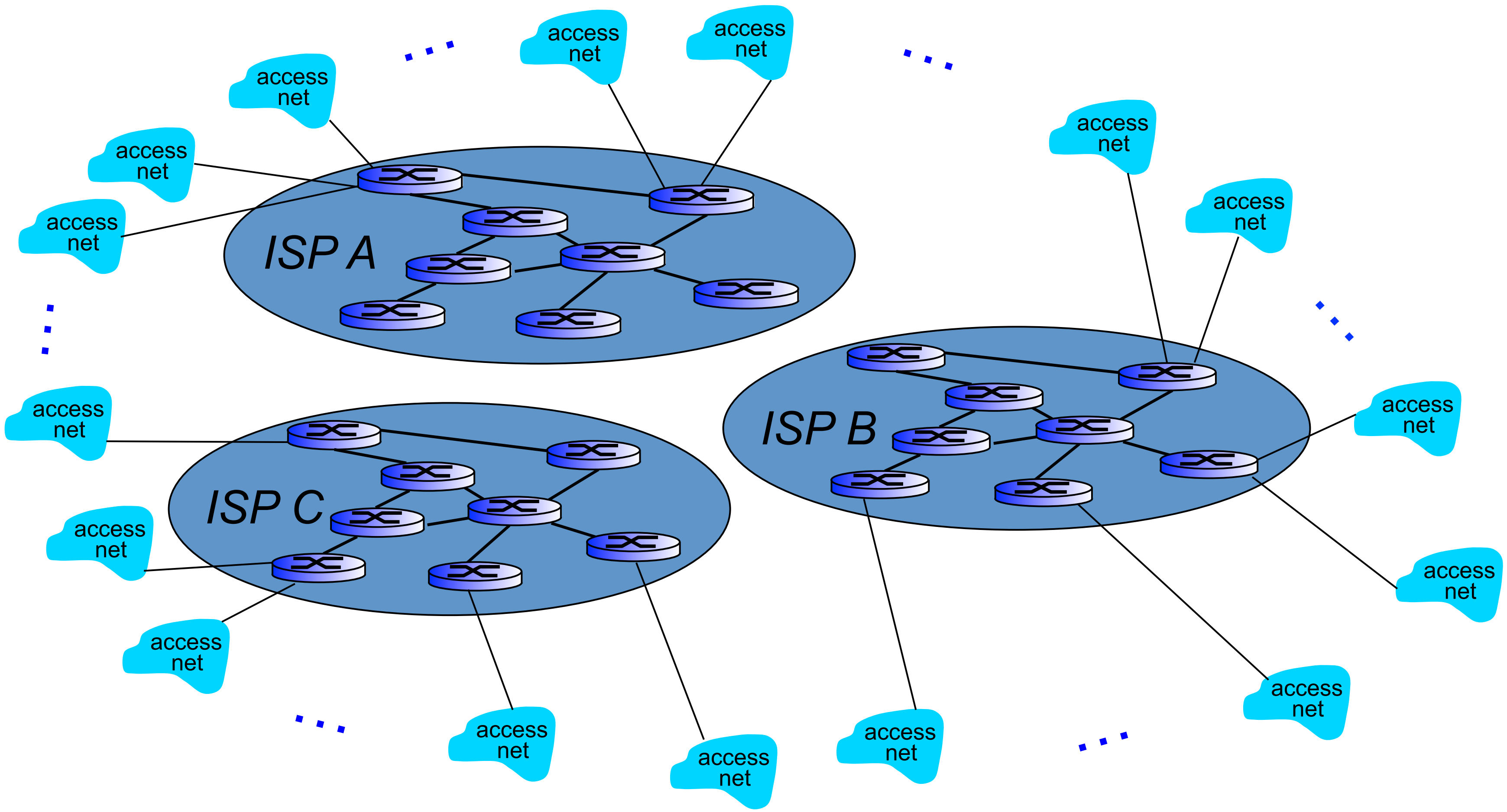
Internet structure: network of networks

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



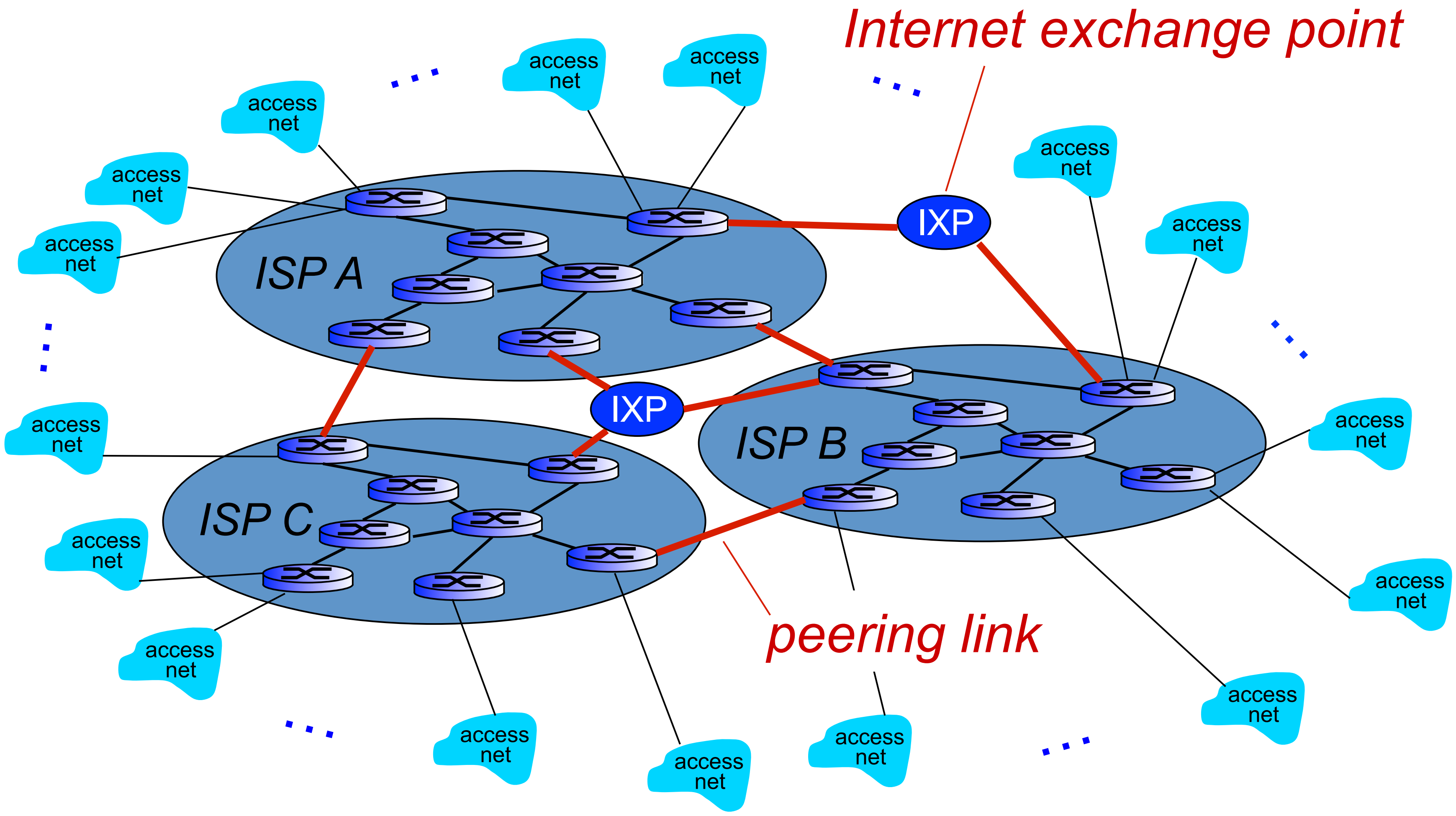
Internet structure: network of networks

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



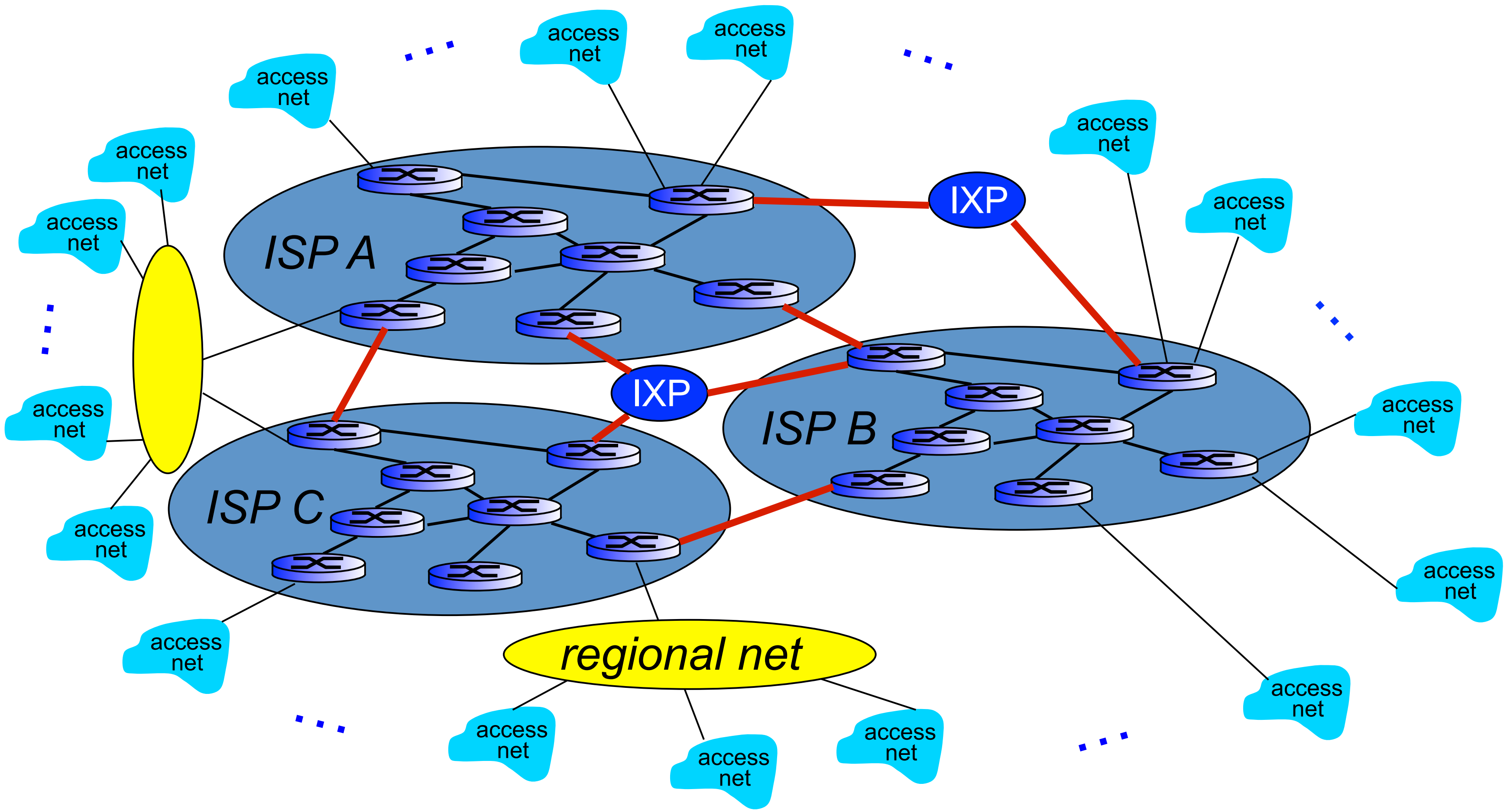
Internet structure: network of networks

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



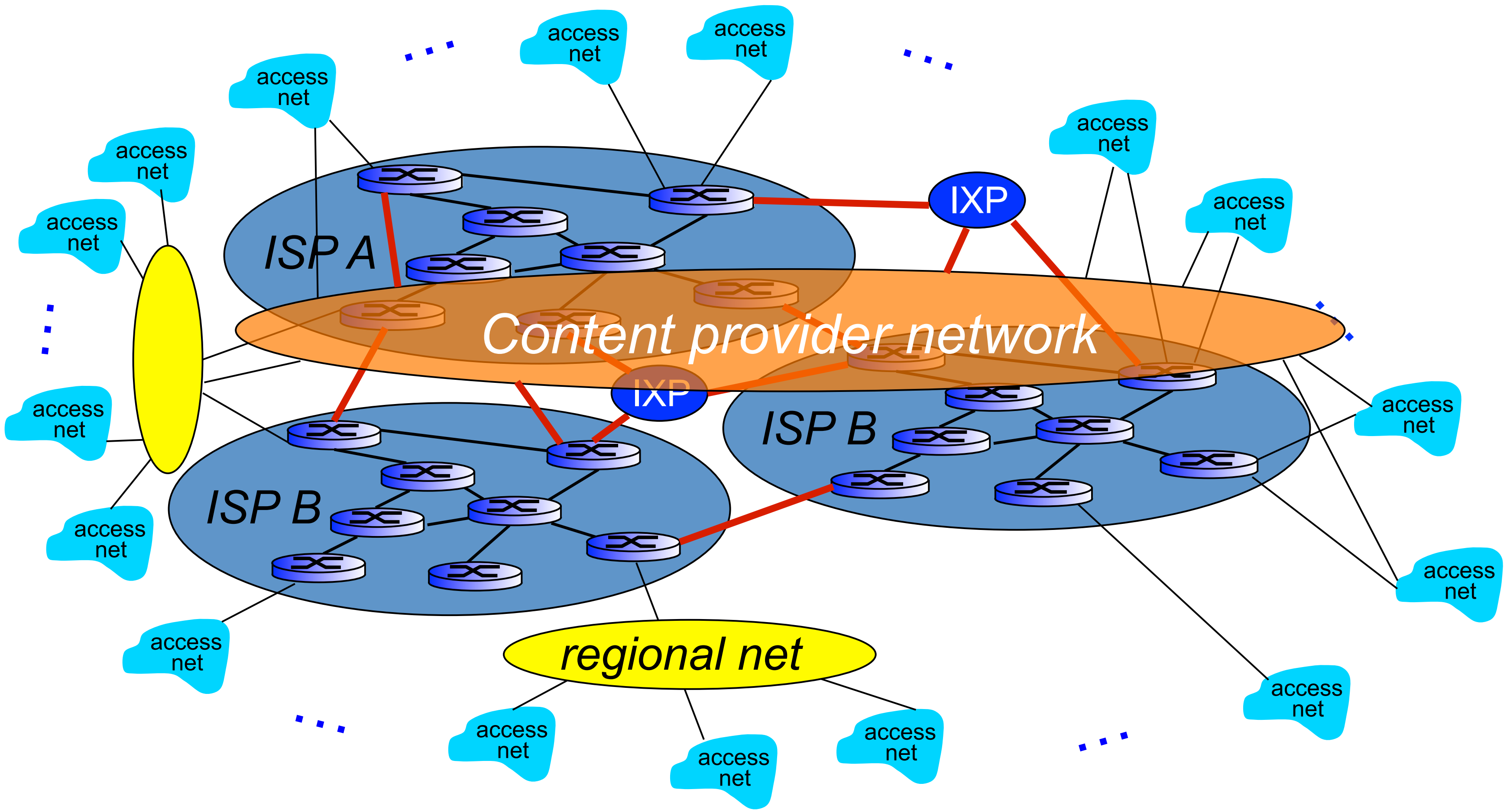
Internet structure: network of networks

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

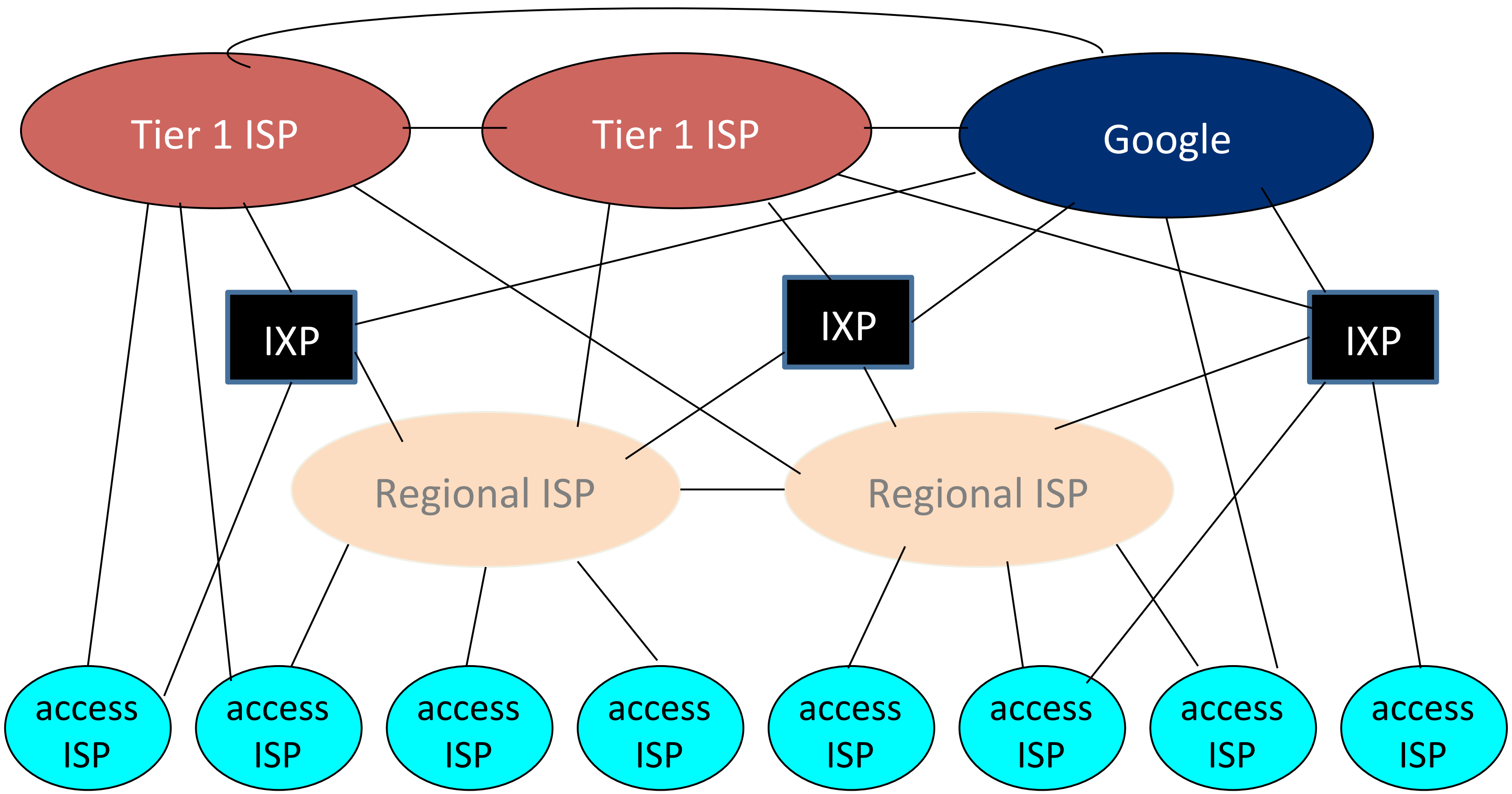


Internet structure: network of networks

… 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

Tier-1 ISP: e.g., Sprint