Introduction to Computer Networks

TDTS04 – Computer Networks and Distributed Systems

Ulf Kargén

Division for Database and Information Techniques (ADIT)
Department of Computer and Information Science (IDA)

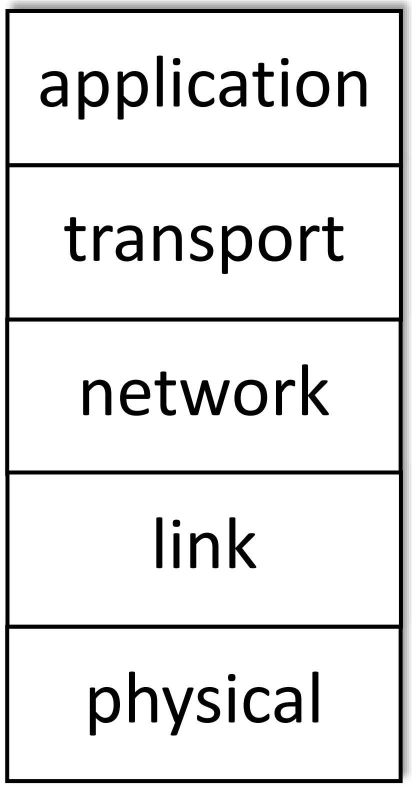
LINKÖPING UNIVERSITY

Slides based in part on https://gaia.cs.umass.edu/kurose_ross/ppt-8e/Chapter_1_v8.2.pptx

Agenda

- Big picture of computer networking
- Important terminology
- Understanding *network layers*

- Based on slides by Kurose and Ross
- Added slides for more context
  - The “why”, not just the “what”…



Running example: reinventing the Internet

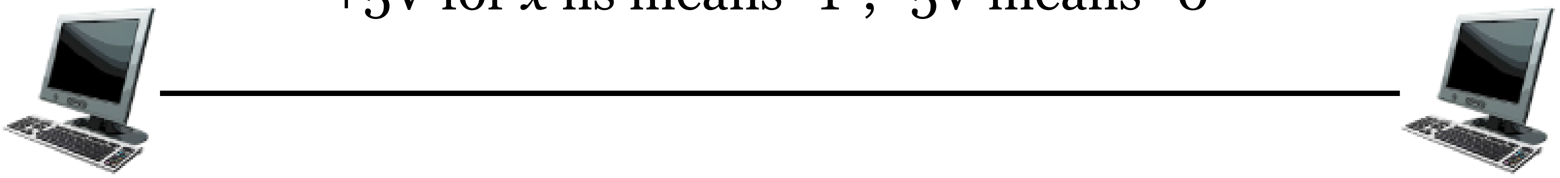
The apocalypse has struck. Now it’s up to you to reinvent the internet…

Where to start?

Making computers communicate

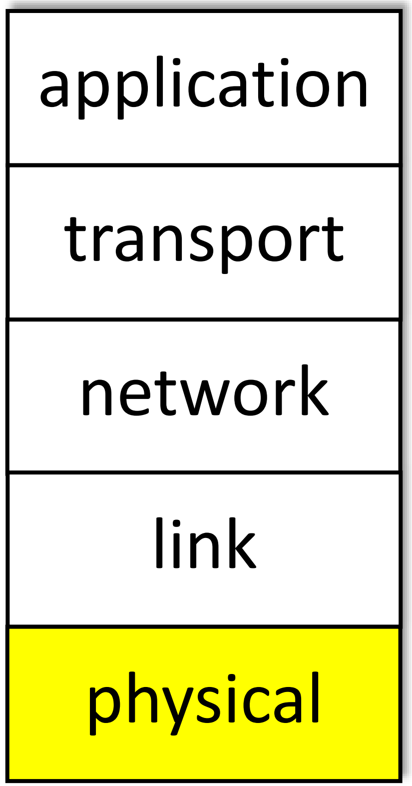
- You have two computers (or *hosts*) and some copper wire.
  - How to make the computers talk to each other?

For example:
+5V for *x* ns means “1”, -5V means “0”



This is a *physical layer* specification

- Not the focus of this course

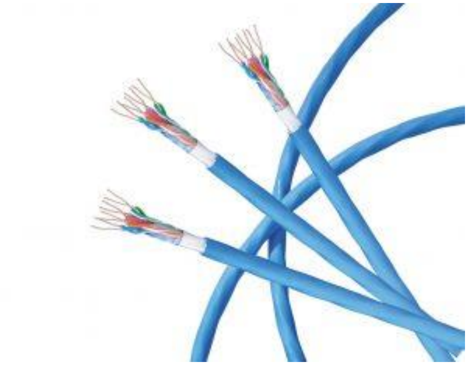


Links: physical media

- bit: propagates between transmitter/receiver pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted pair (TP)

- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps Ethernet



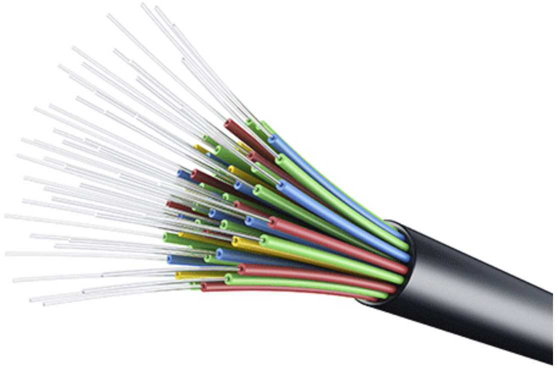
Links: physical media

Coaxial cable:

- two concentric copper conductors
- bidirectional
- broadband:
  - multiple frequency channels on cable
  - 100's Mbps per channel

Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (10's-100's Gbps)
- low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise



Links: physical media

Wireless radio

- signal carried in various "bands" in electromagnetic spectrum
- no physical "wire"
- broadcast, "half-duplex" (sender to receiver)
- propagation environment effects:
  - reflection
  - obstruction by objects
  - Interference/noise

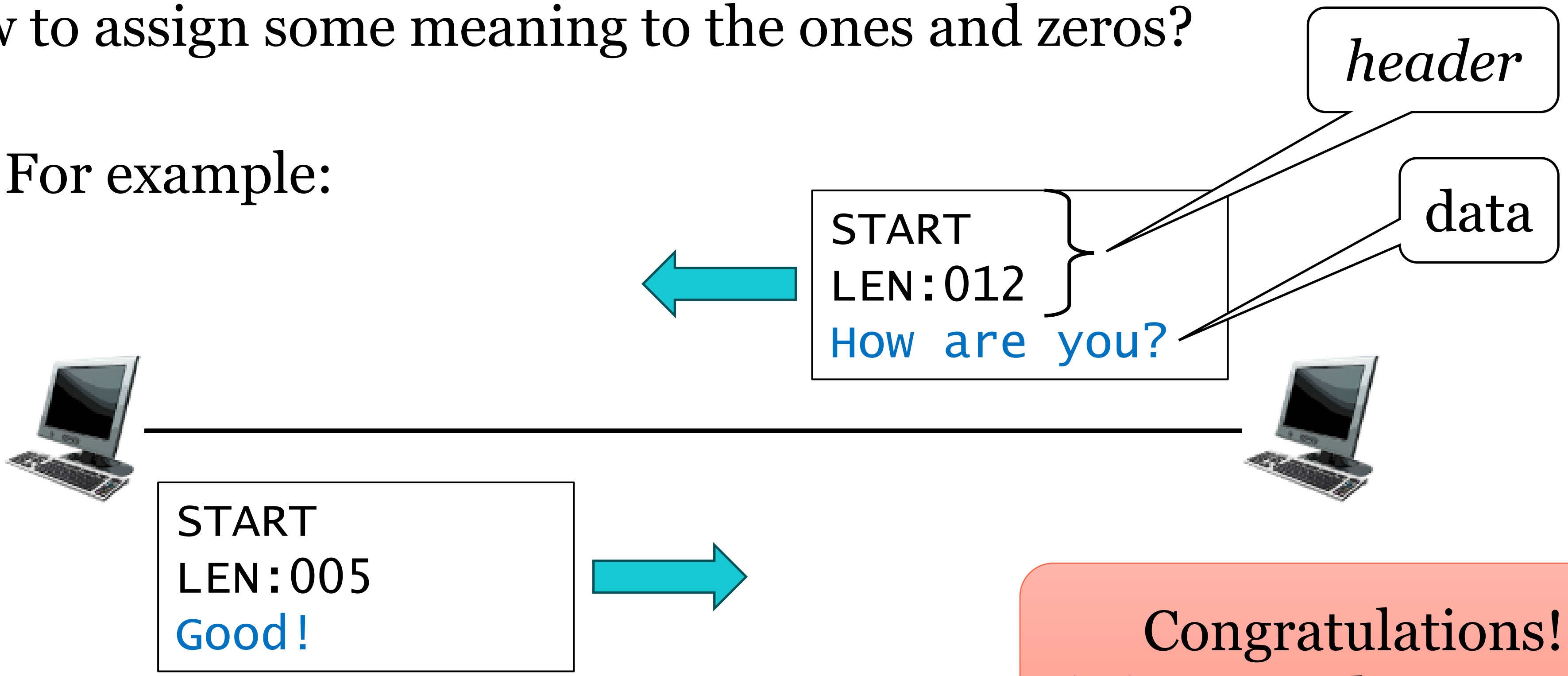
Radio link types:

- Wireless LAN (WiFi)
  - 10-100's Mbps; 10's of meters
- wide-area (e.g., 4G/5G cellular)
  - 10's Mbps (4G) over ~10 Km
- Bluetooth: cable replacement
  - short distances, limited rates
- terrestrial microwave
  - point-to-point; 45 Mbps channels
- satellite
  - up to < 100 Mbps (Starlink) downlink
  - 270 msec end-end delay (geostationary)

Sending messages

How to assign some meaning to the ones and zeros?

For example:



Congratulations! You just (re)invented a network *protocol*.

What’s a protocol?

Human protocols:

- “what’s the time?”
- “I have a question”
- introductions

Rules for:

… specific messages sent

… specific actions taken when message received, or other events

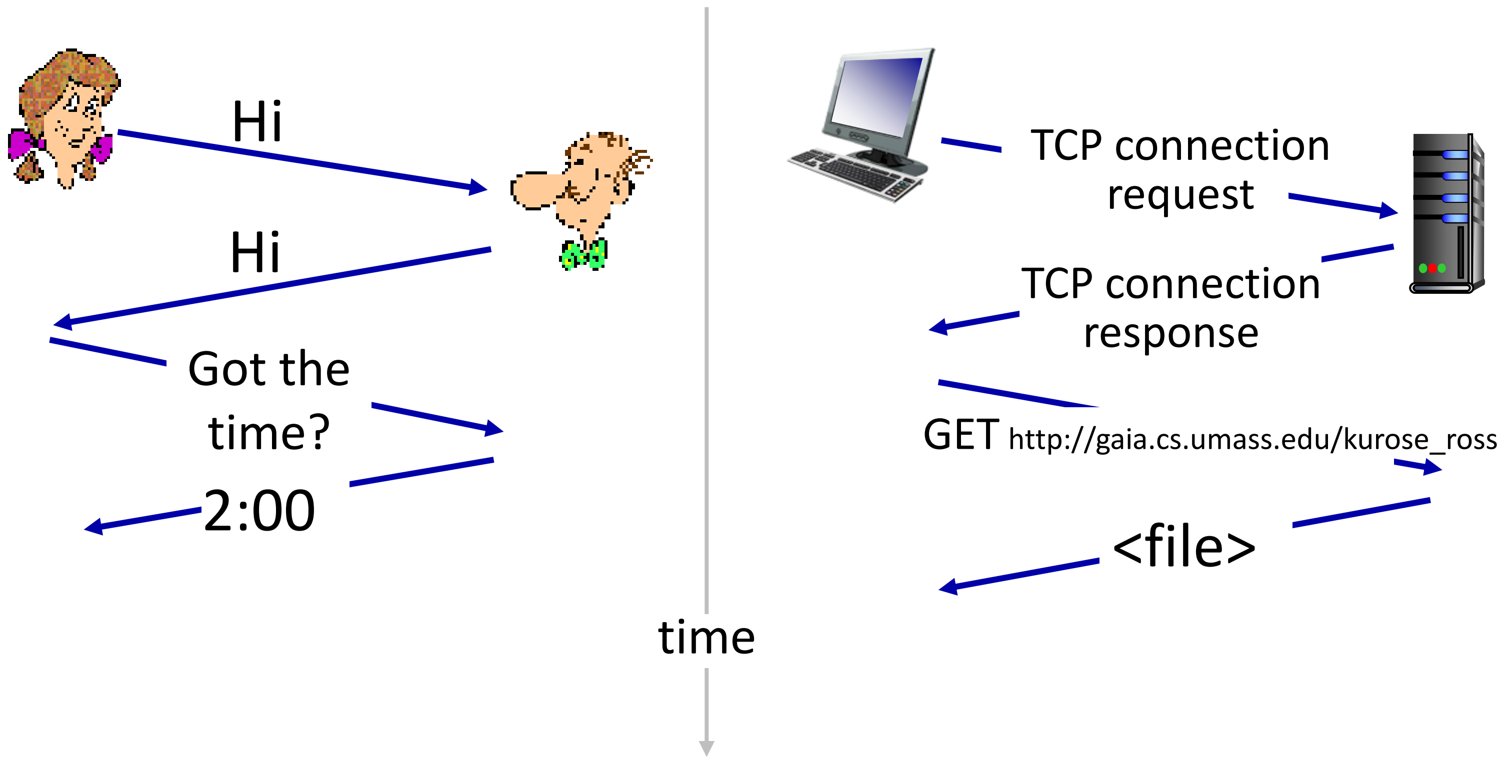
Network protocols:

- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

Protocols define the format, order of messages sent and received among network entities, and actions taken on message transmission, receipt

What’s a protocol?

A human protocol and a computer network protocol:

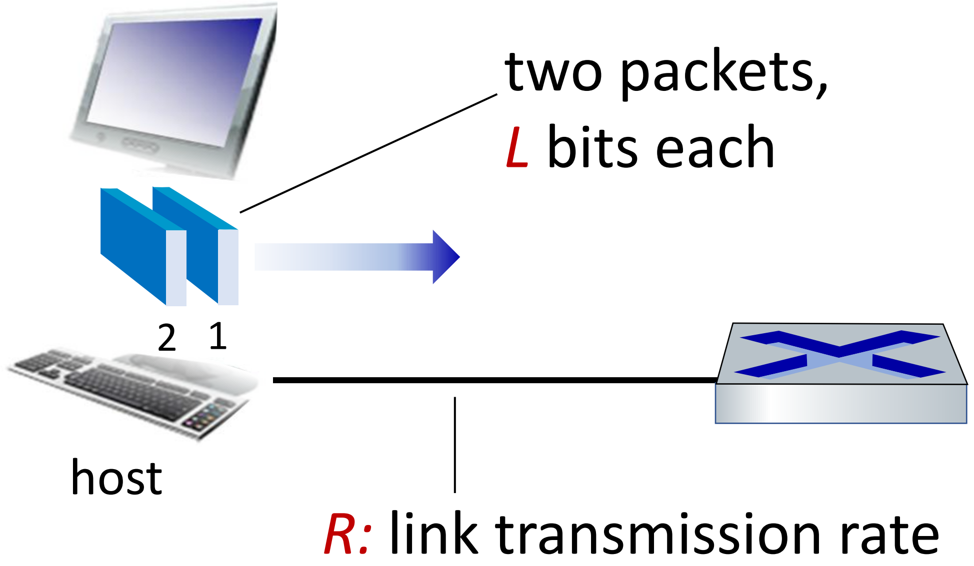


Q: other human protocols?

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*

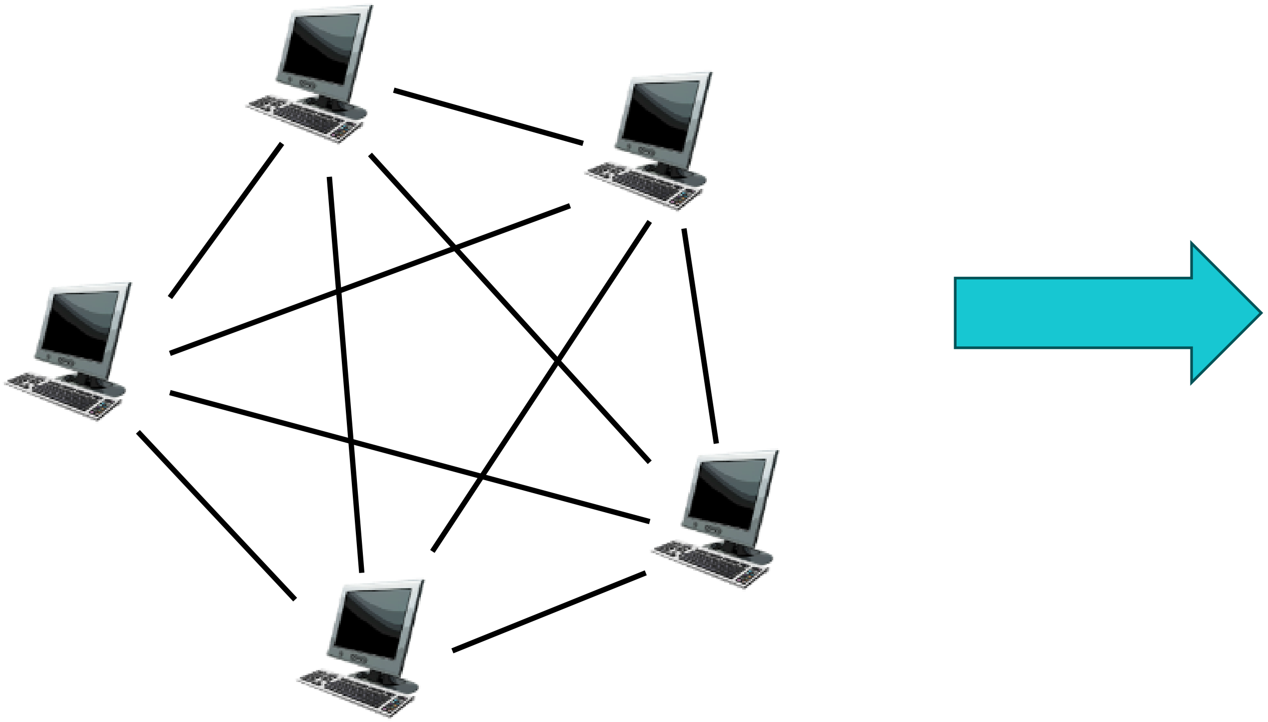


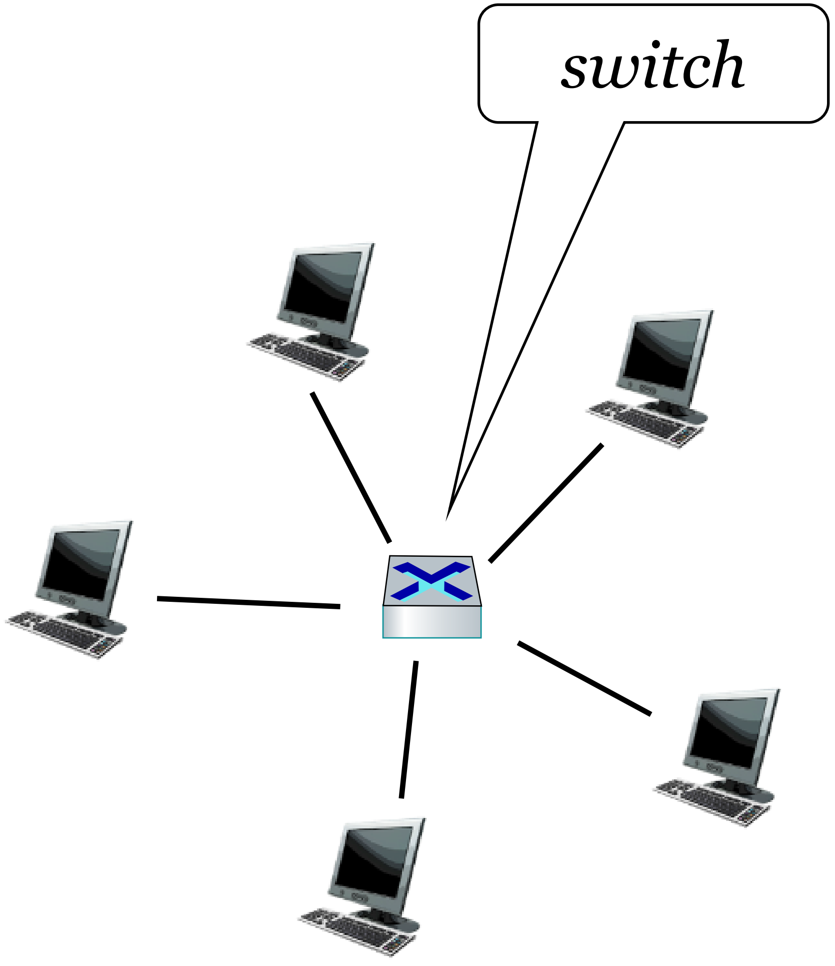
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Creating a network

Everyone connected to everyone?

- We'd probably run out of copper pretty soon ...

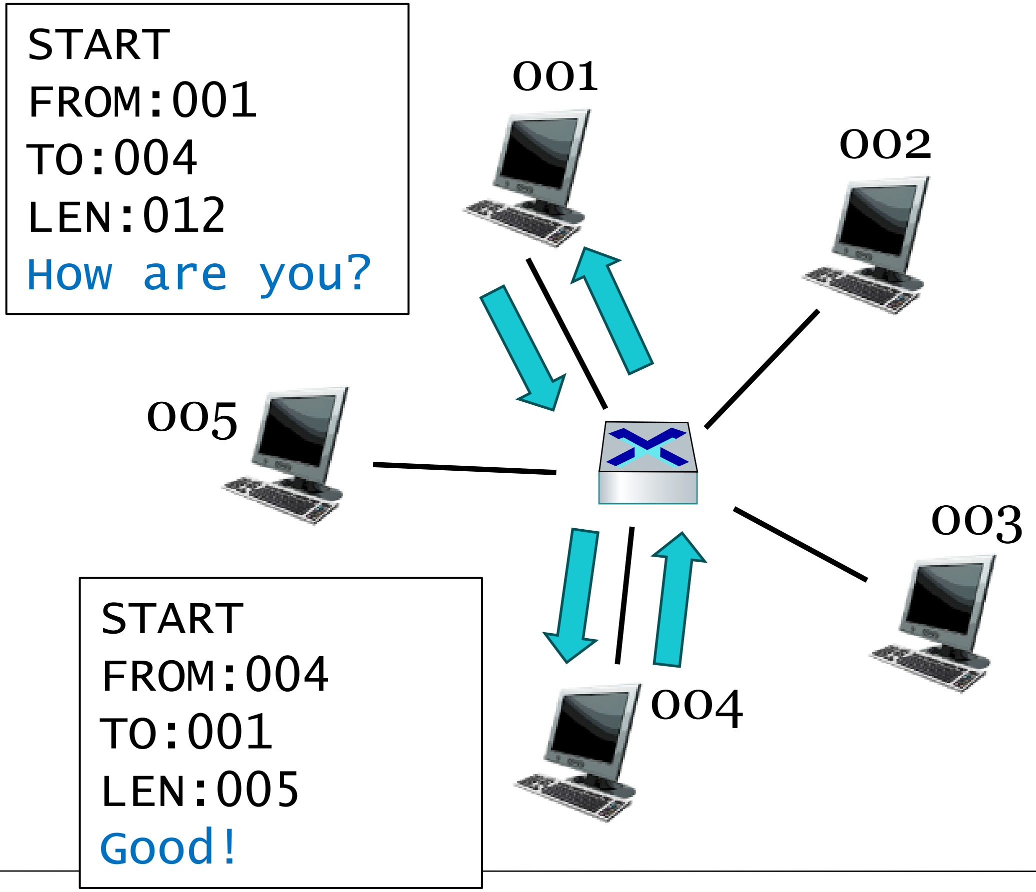




Creating a network

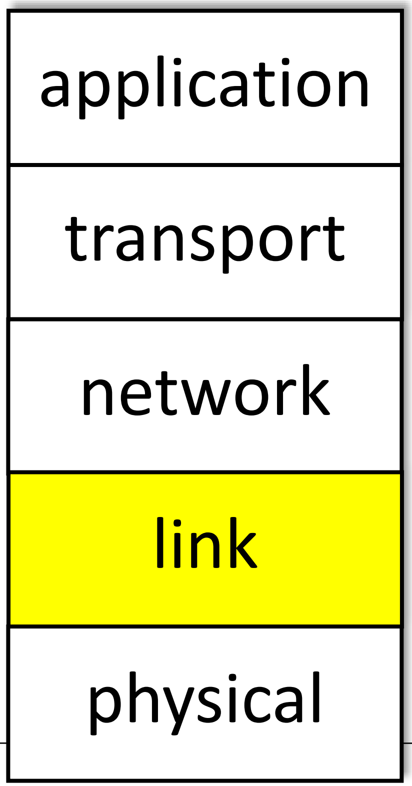
New problem: how to know which message goes where?

- Add source and destination *addresses* to our protocol
- Assign each host an address
- Add addresses to the message format

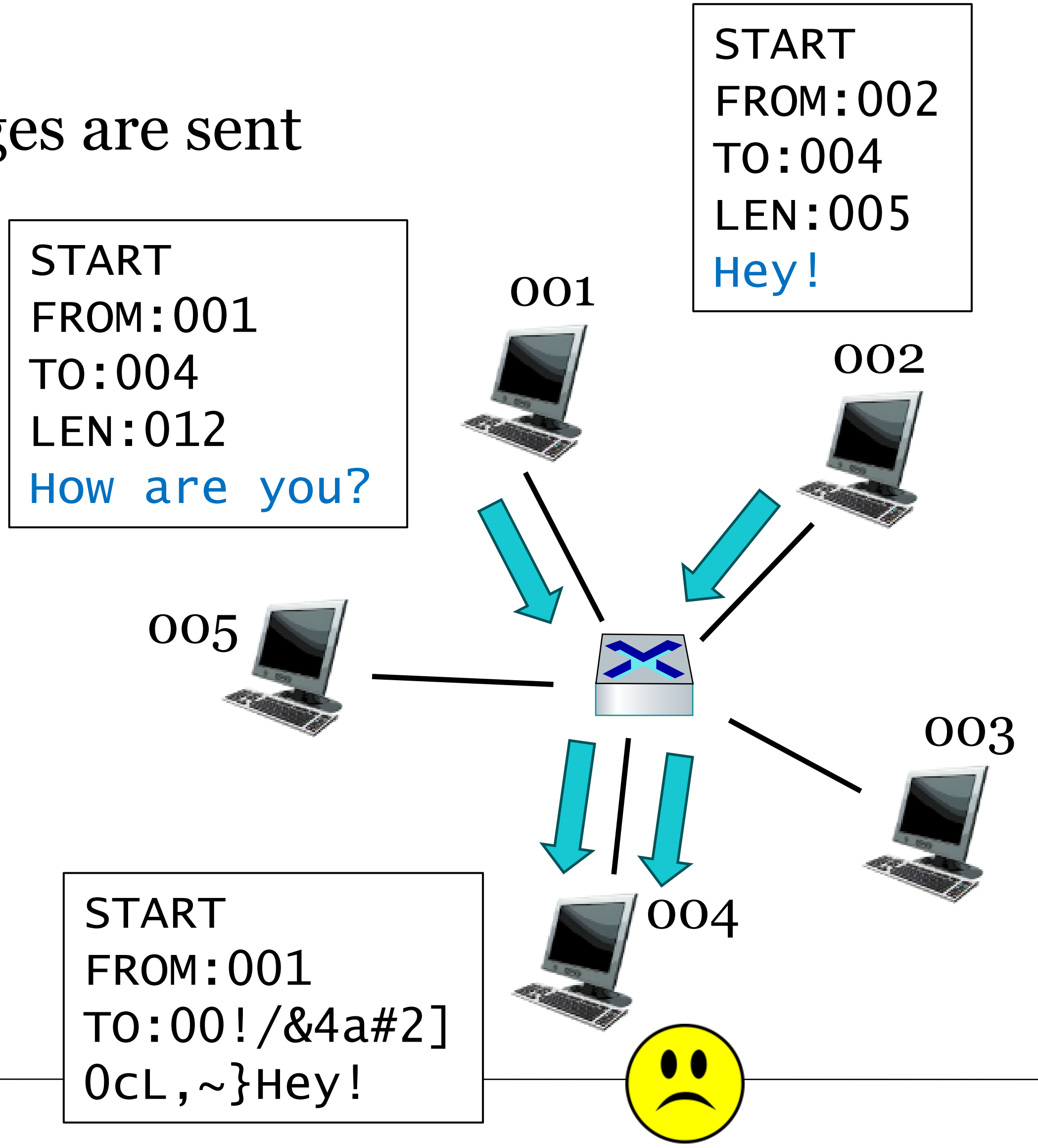


Creating a network

Another problem: what if multiple messages are sent simultaneously to one host?



- Need a mechanism to share physical medium to avoid this
- This is the purpose of the *link layer*

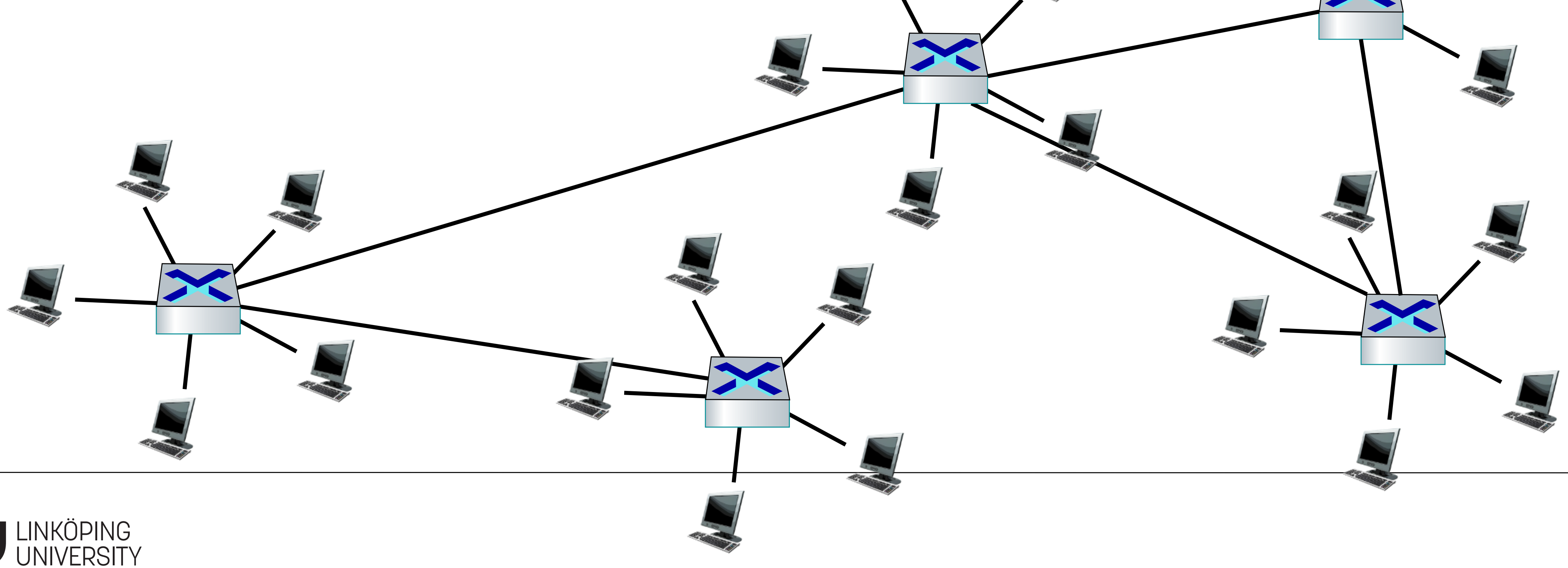


The post-apocalyptic Internet grows…

Everyone connecting to the same switch doesn't scale…
(Again, we'd soon run out of copper)

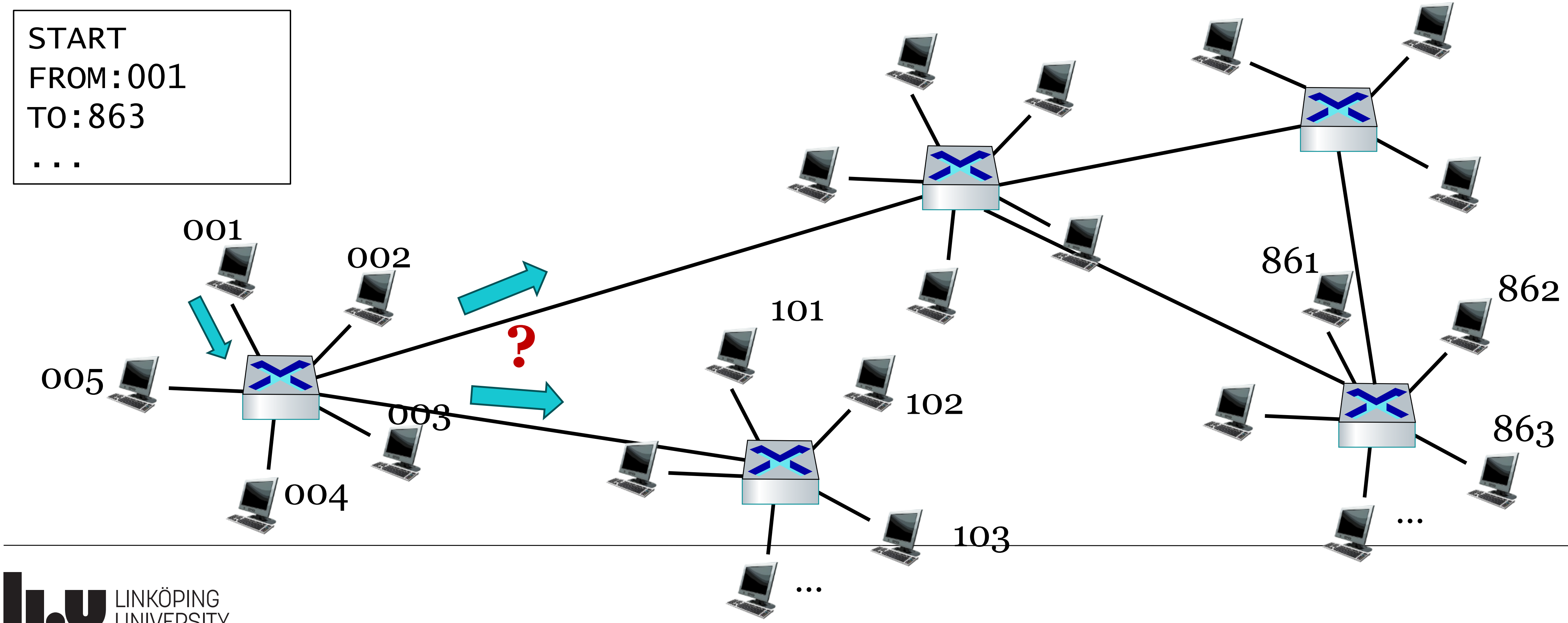
Solution: a network of networks!

But this creates some new problems…



The post-apocalyptic Internet grows…

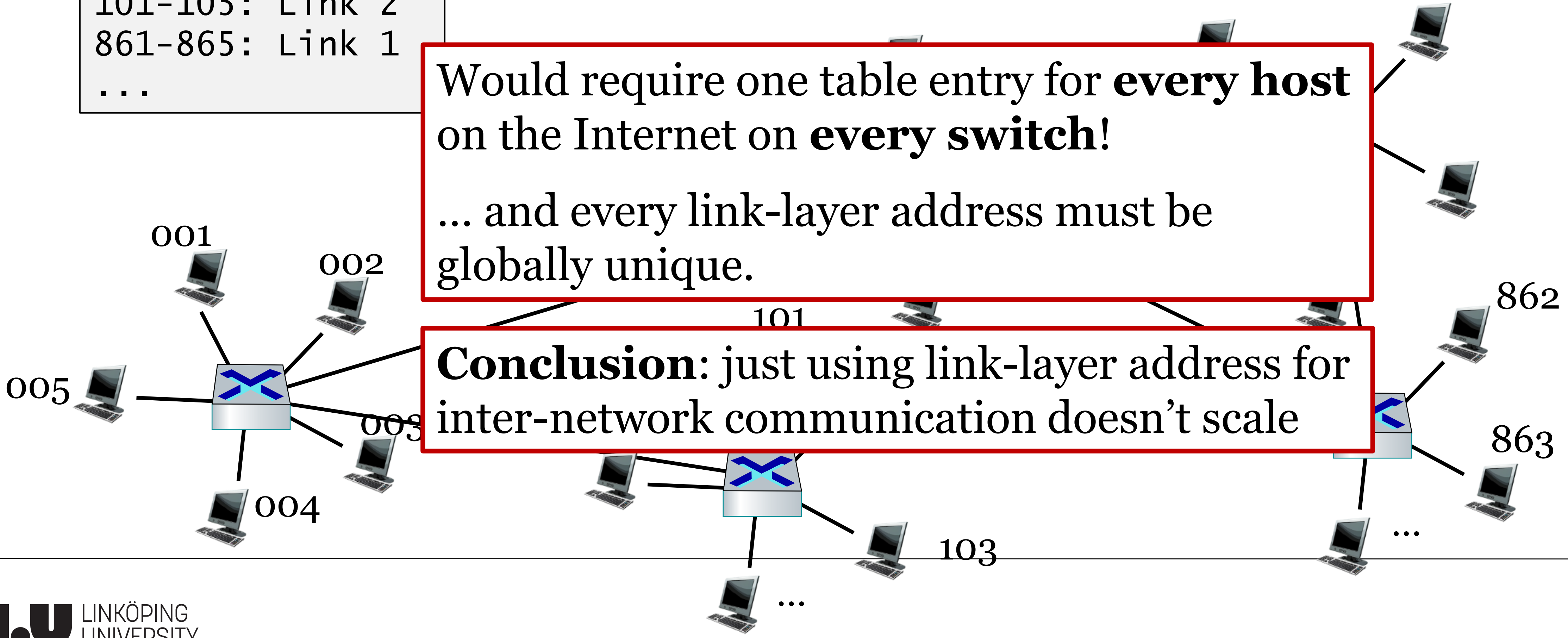
How will switch know on which link to send messages to other networks?



The post-apocalyptic Internet grows…

Potential solution: maintain a table in each switch:

```
001-005: local
101-105: Link 2
861-865: Link 1
...
```



Would require one table entry for **every host** on the Internet on **every switch**!

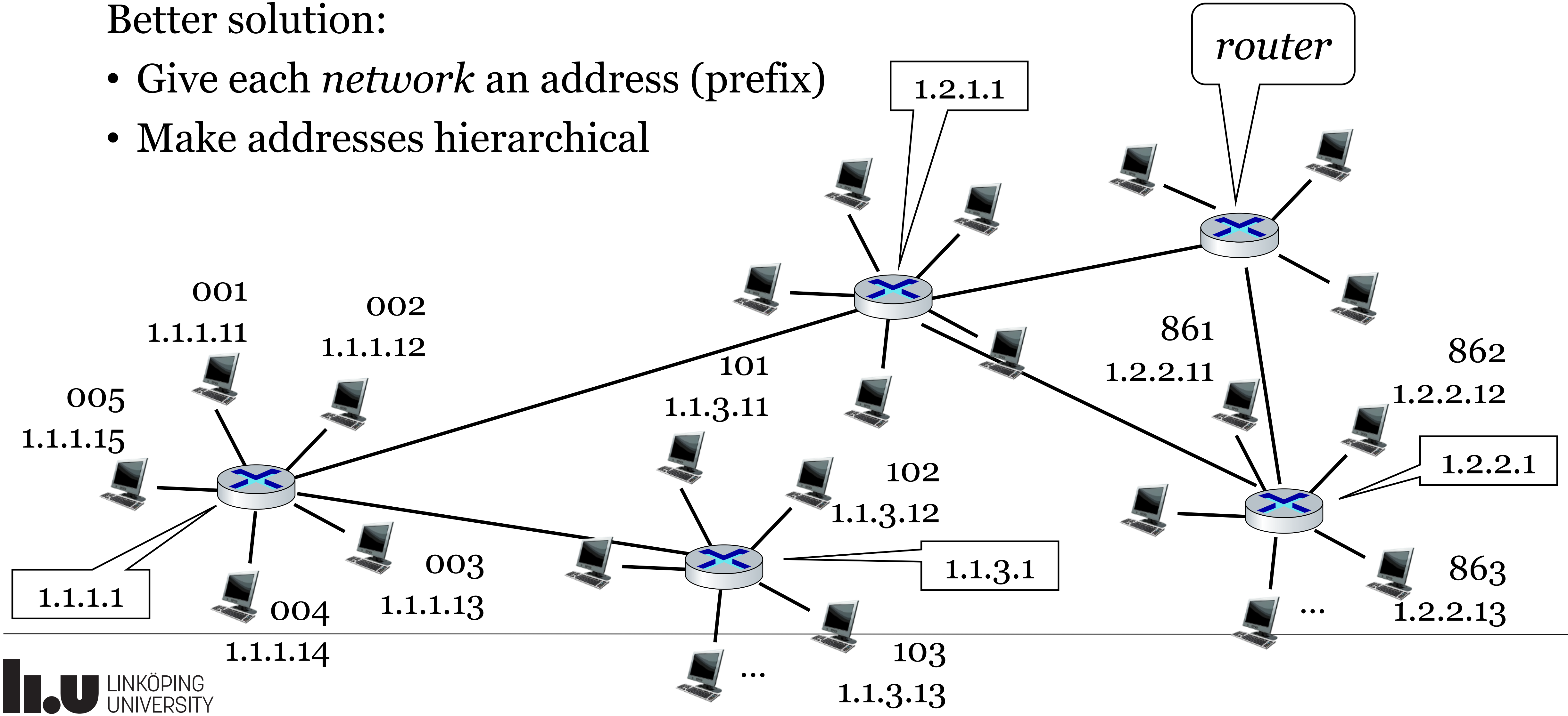
… and every link-layer address must be globally unique.

Conclusion: just using link-layer address for inter-network communication doesn't scale

The post-apocalyptic Internet grows…

Better solution:

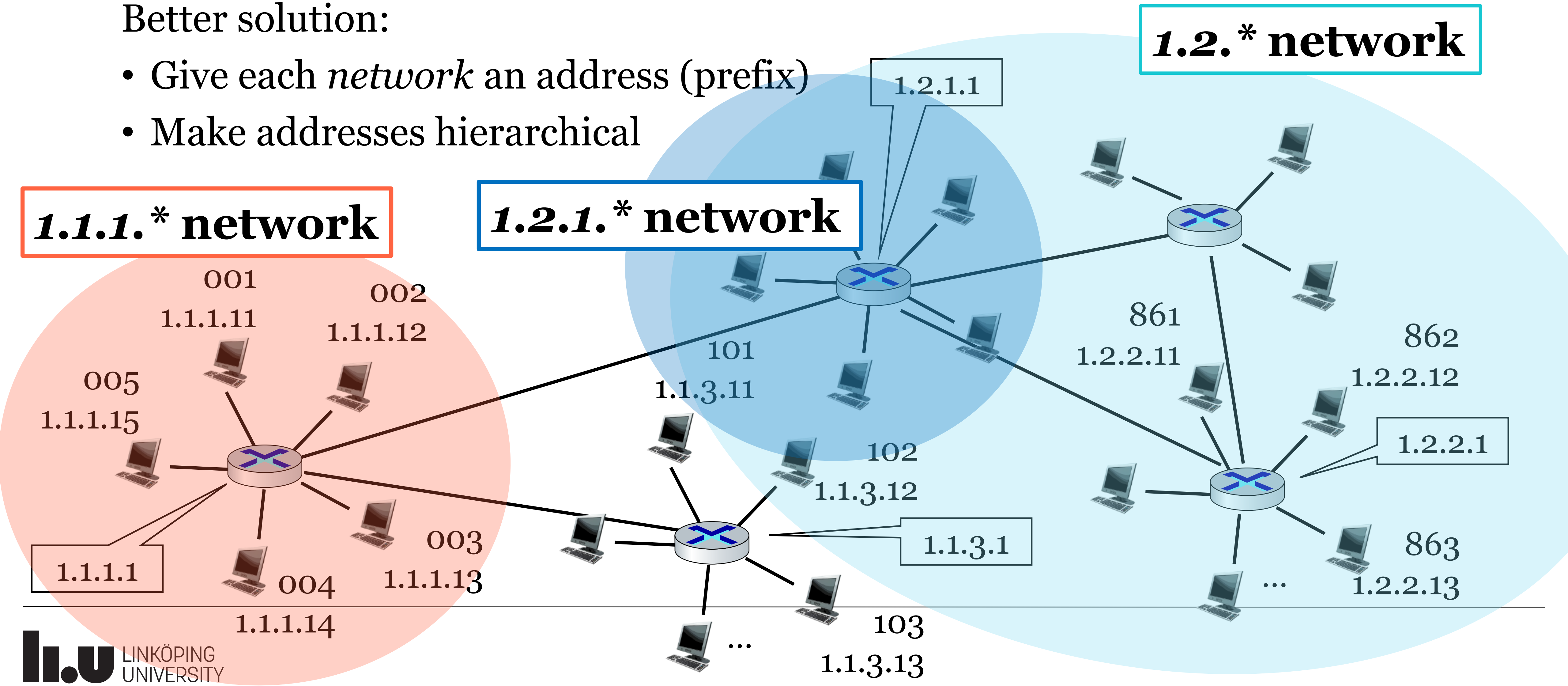
- Give each *network* an address (prefix)
- Make addresses hierarchical



The post-apocalyptic Internet grows…

Better solution:

- Give each *network* an address (prefix)
- Make addresses hierarchical

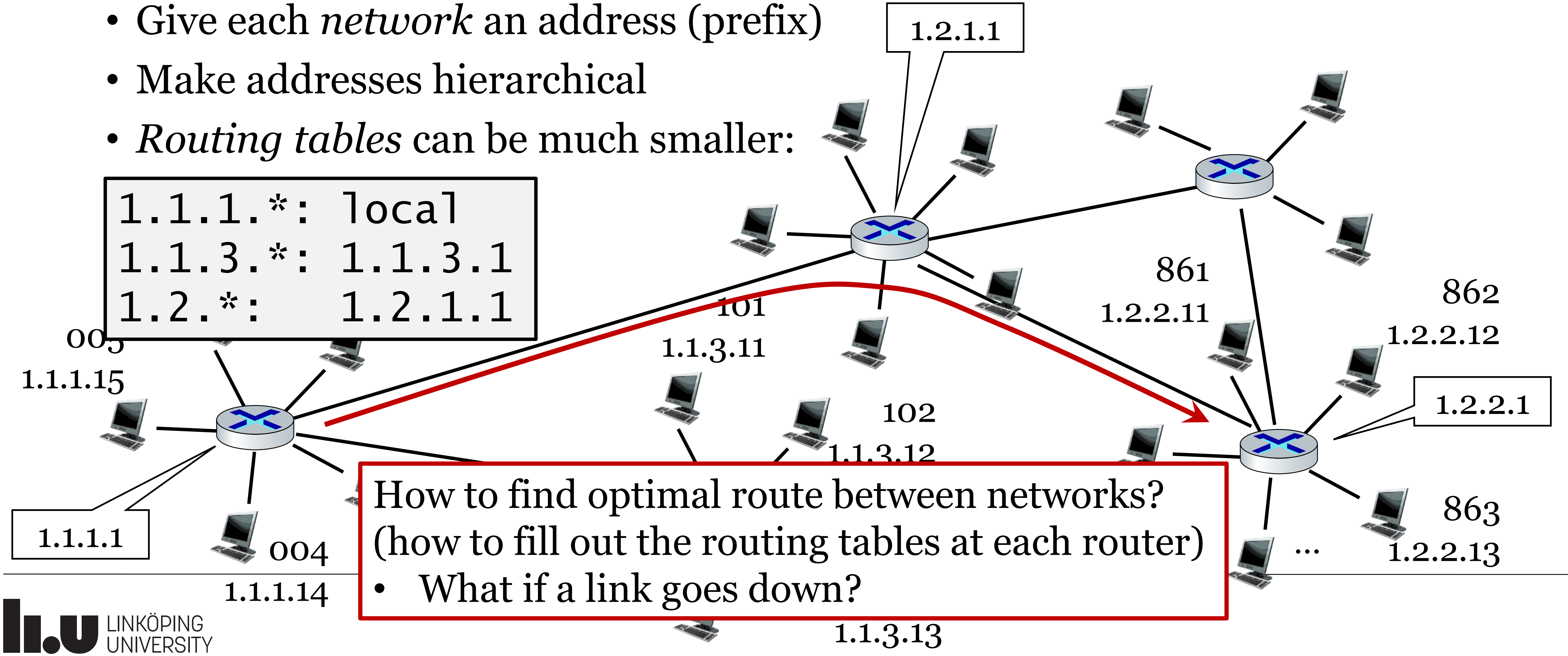


The post-apocalyptic Internet grows…

Better solution:

- Give each *network* an address (prefix)
- Make addresses hierarchical
- *Routing tables* can be much smaller:

```
1.1.1.*: local
1.1.3.*: 1.1.3.1
1.2.*:   1.2.1.1
```



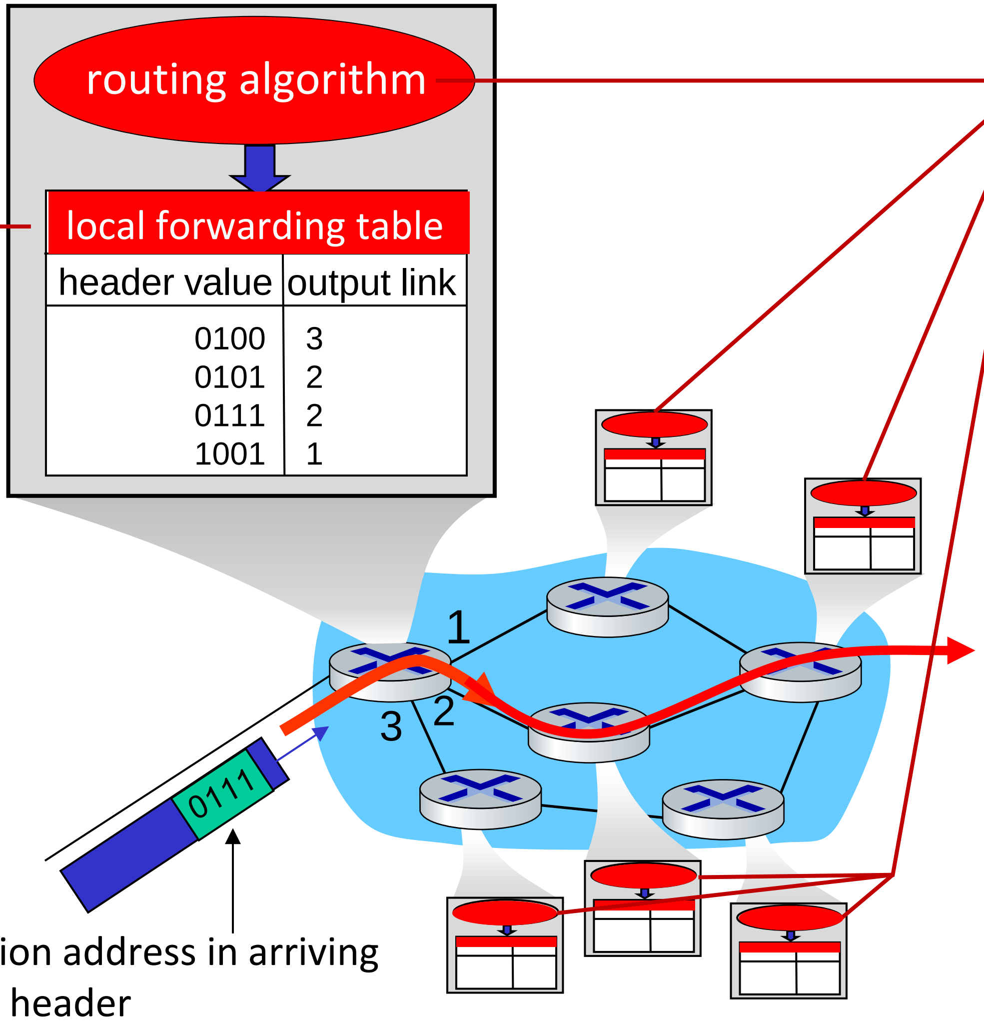
How to find optimal route between networks?
(how to fill out the routing tables at each router)

- What if a link goes down?

Routing vs switching/forwarding

Forwarding:

- aka “switching”
- *local* action: move arriving packets from router’s input link to appropriate router output link



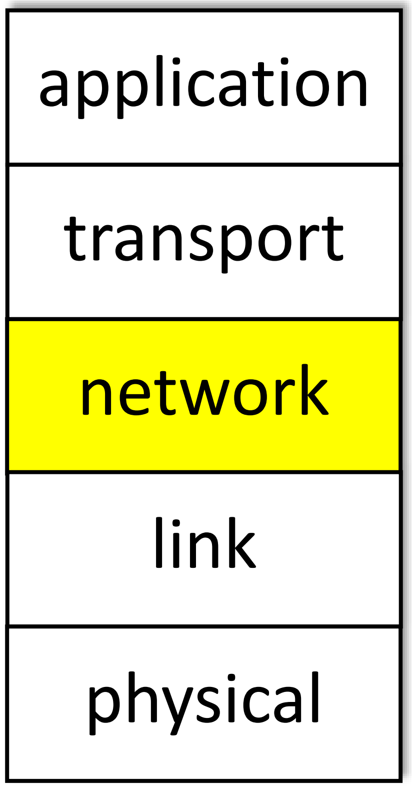
Routing:

- *global* action: determine source-destination paths taken by packets
- routing algorithms


45 h
3,053 miles
48 h
3,309 miles
United States
routing
San Jose
Northampton


forwarding
forwarding
45 h
3,053 miles
48 h
3,309 miles
United States
Seattle
Victoria
WASHINGTON
Portland
San Francisco
San Jose
CALIFORNIA
Las Vegas
Los Angeles
San Diego
Phoenix
Tucson
El Paso
ARIZONA
NEW MEXICO
COLORADO
WYOMING
NORTH DAKOTA
SOUTH DAKOTA
MINNESOTA
Minneapolis
WISCONSIN
IOWA
KANSAS
Kansas City
MISSOURI
OKLAHOMA
TEXAS
Dallas
Austin
Houston
San Antonio
ARKANSAS
LOUISIANA
New Orleans
MISSISSIPPI
ALABAMA
Atlanta
TENNESSEE
Nashville
KENTUCKY
ILLINOIS
INDIANA
Indianapolis
Chicago
MICHIGAN
Detroit
OHIO
Toronto
Ottawa
Montreal
Quebec City
MAINE
VERMONT
Northampton
Tampa
FLORIDA
BAJA CALIFORNIA
SONORA
CHIHUAHUA
COAHUILA
Sacramento
Cleveland
DOWNTOWN
OHIO CITY
CENTRAL

The post-apocalyptic Internet grows…

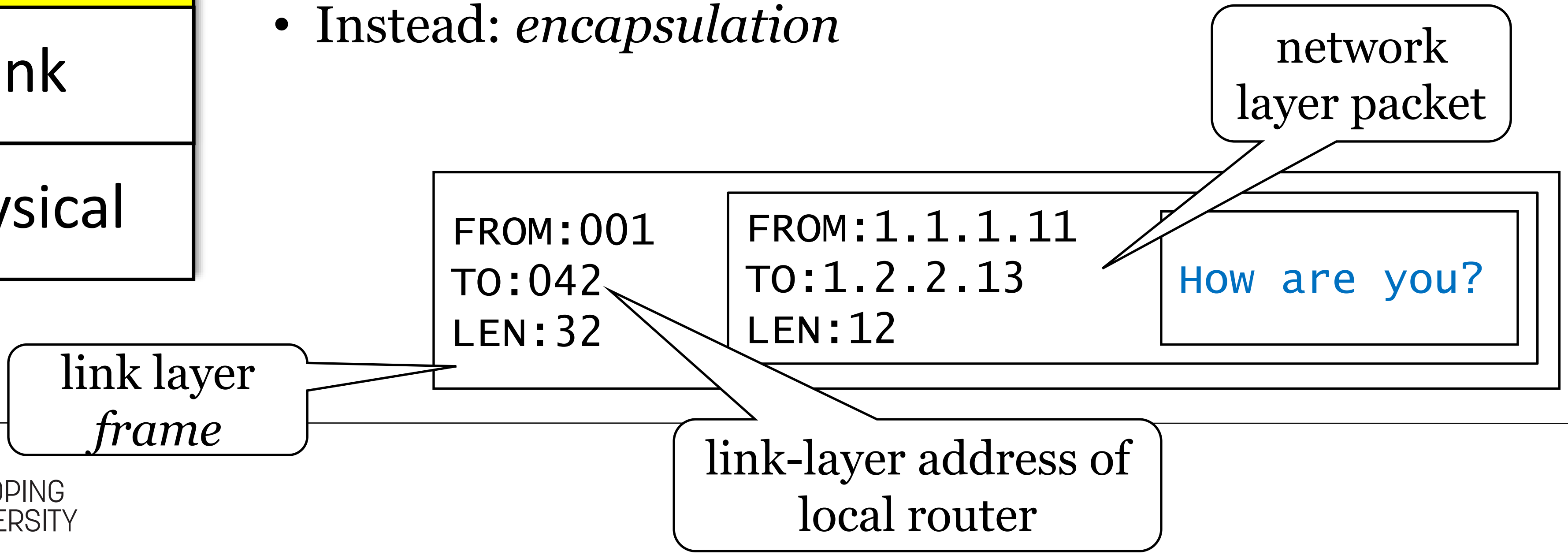


Both inter-network data transfer and routing is handled at the *network layer*

- Data plane
- Control plane

Note: network layer does not “replace” link layer.

- Instead: *encapsulation*



Packet loss and delay

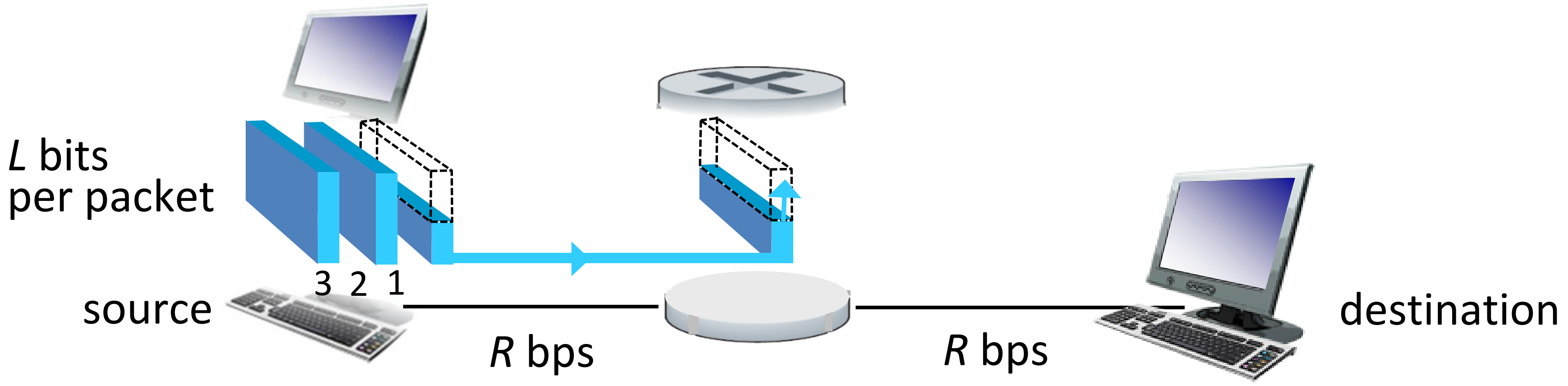
Some new implications of a large complex network

- Packets must now pass along many intermediate routers
- Router must do some processing of each packet
- Sometimes input rate > output link throughput

(the last two points:) need to temporarily *cache* packets

Leads to some new problems…

Packet-switching: store-and-forward

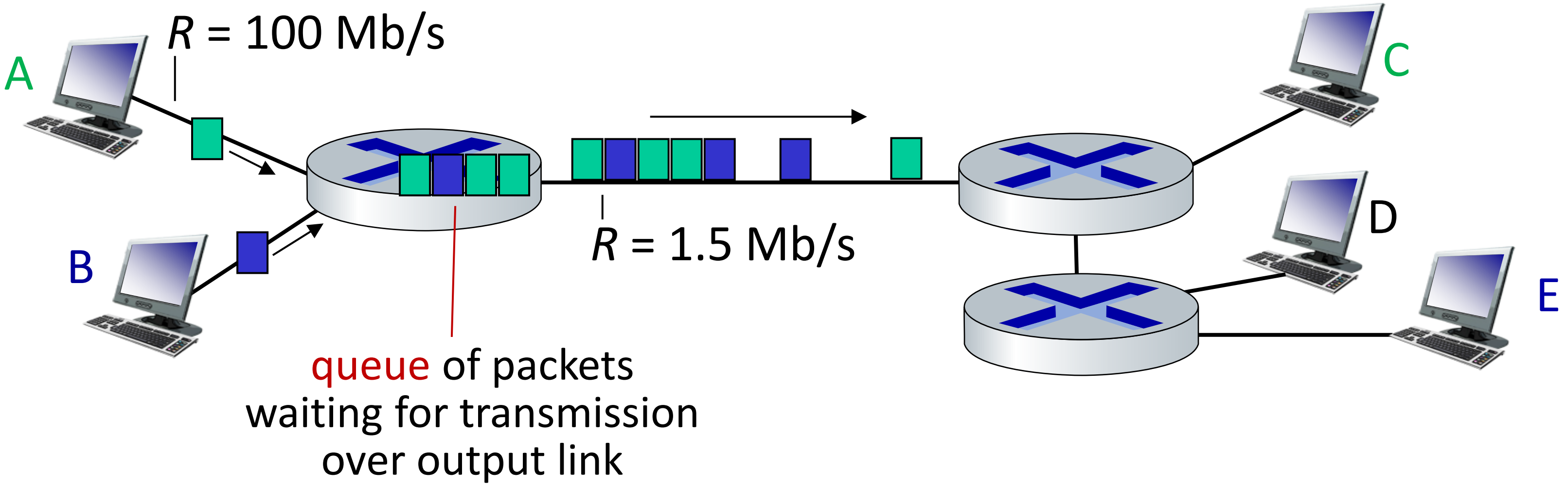


- packet transmission delay: takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

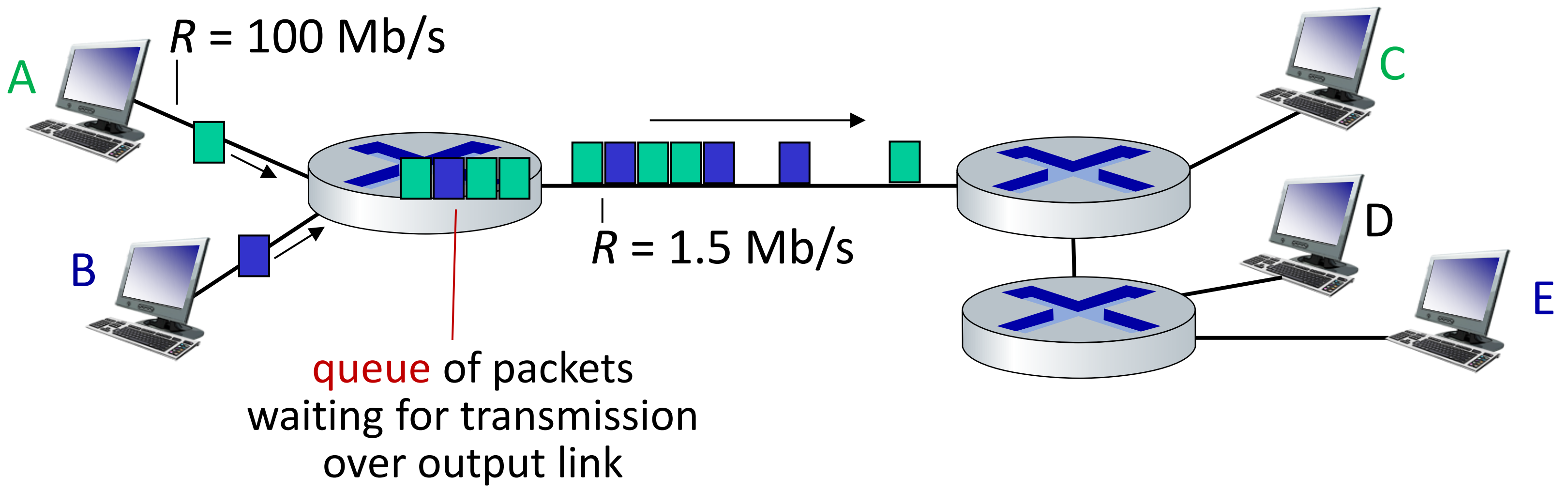
- *L* = 10 Kbits
- *R* = 100 Mbps
- one-hop transmission delay = 0.1 msec

Packet-switching: queueing



Queueing occurs when work arrives faster than it can be serviced:

Packet-switching: queueing



Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

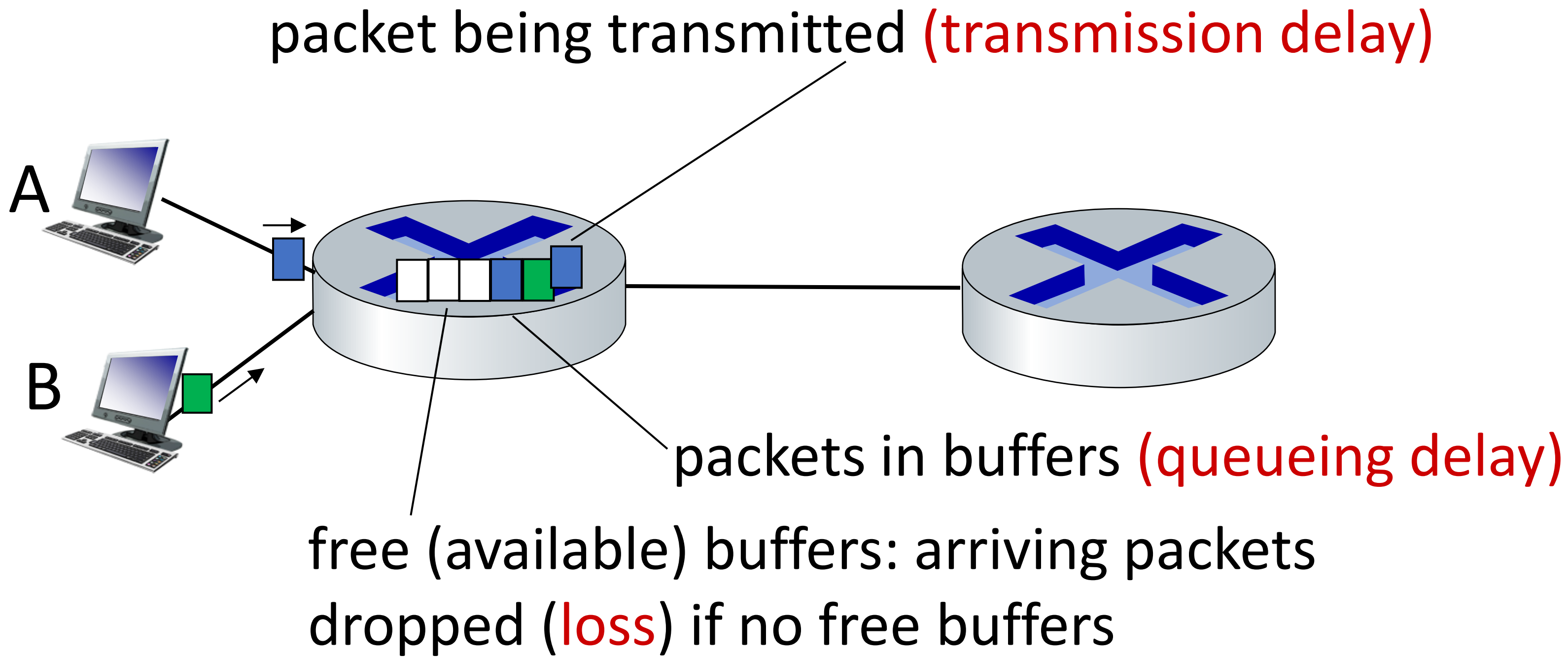
- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Interactive Animation: Queuing and Loss

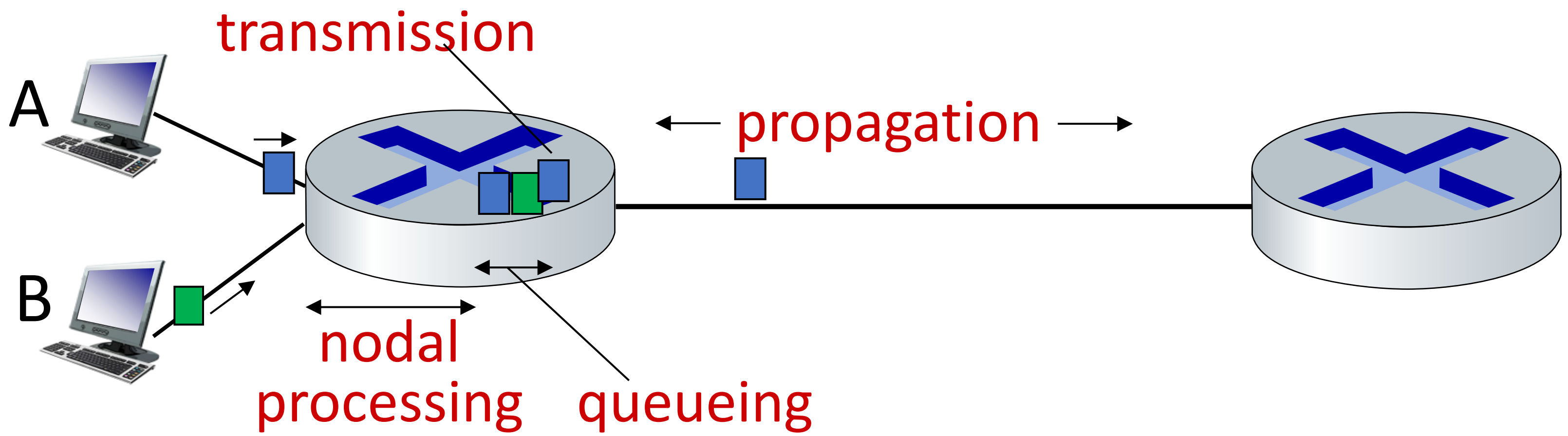
https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/queuing-loss-applet/index.html

How do packet delay and loss occur?

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet *loss* occurs when memory to hold queued packets fills up



Packet delay: four sources



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

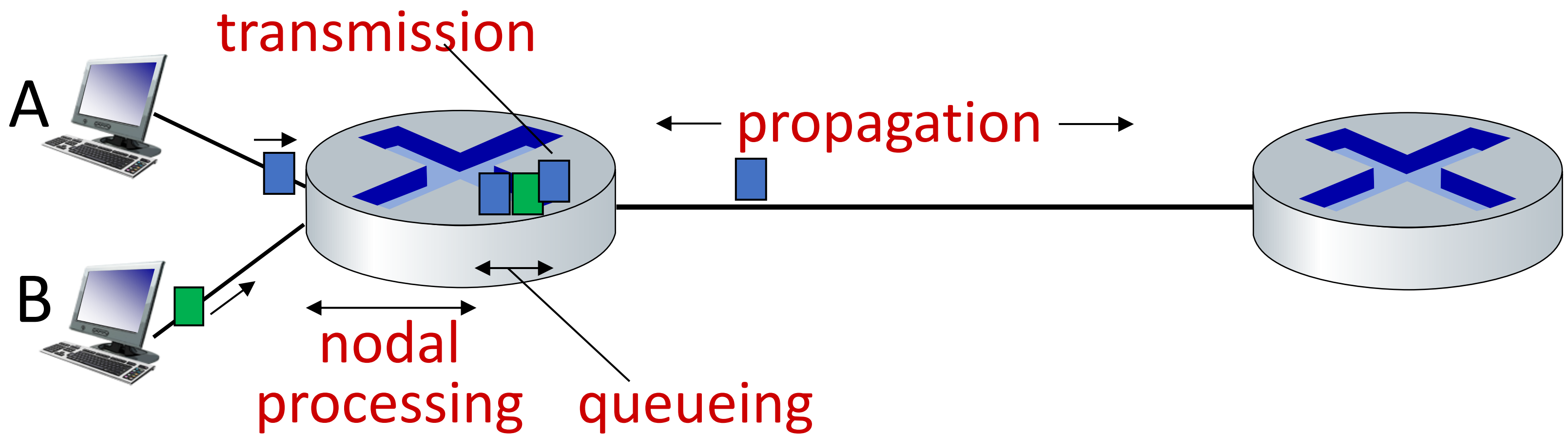
d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < microsecs

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Interactive Animation: Transmission versus Propagation Delay

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/transmission-vs-propogation-delay/transmission-propagation-delay-ch1/index.html

Packet delay: four sources



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

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:

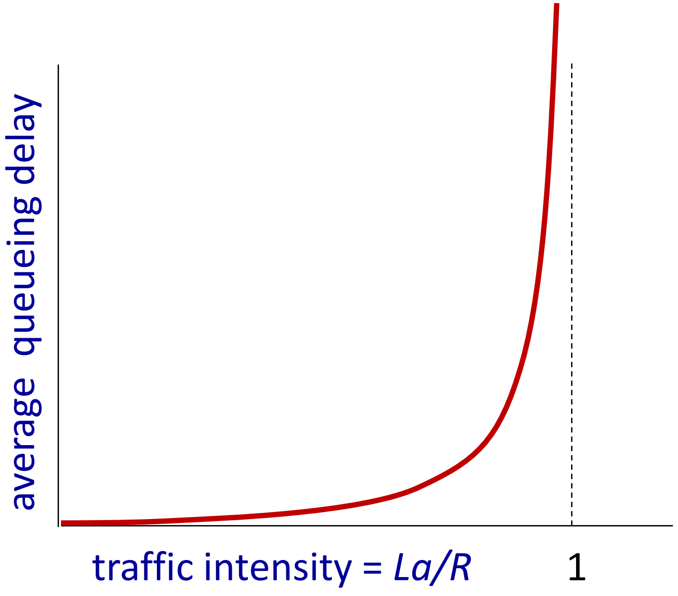
- d: length of physical link
- s: propagation speed (~2x10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop}
very different

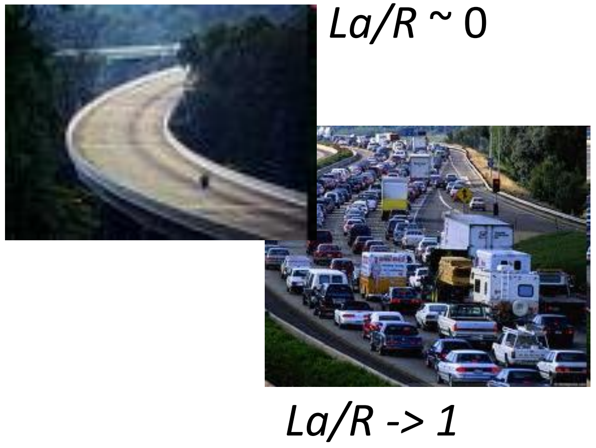
Packet queueing delay (revisited)

- *a:* average packet arrival rate
- *L:* packet length (bits)
- *R:* link bandwidth (bit transmission rate)

$$\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}}$$ *"traffic intensity"*

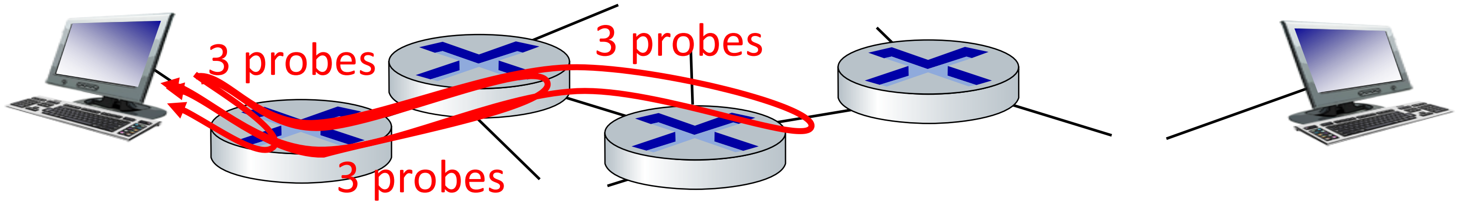


- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!



“Real” Internet delays and routes

- what do “real” Internet delay & loss look like?
- **traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination (with time-to-live field value of *i*)
  - router *i* will return packets to sender
  - sender measures time interval between transmission and reply



Real Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

3 delay measurements to border1-rt-fa5-1-0.gw.umass.edu

trans-oceanic link

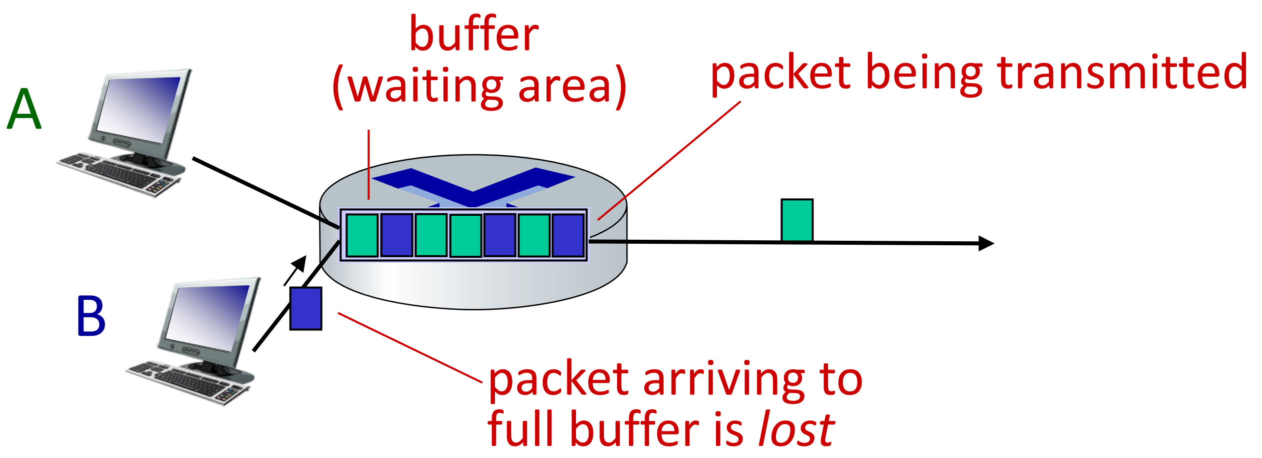
looks like delays *decrease*! Why?

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

Packet loss

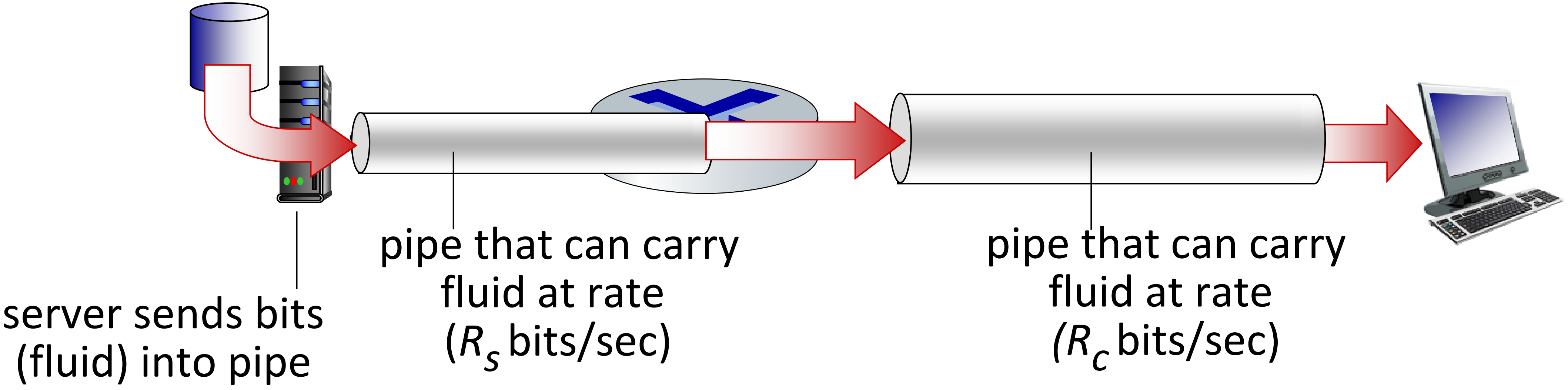
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation (on publisher's website) of queuing and loss

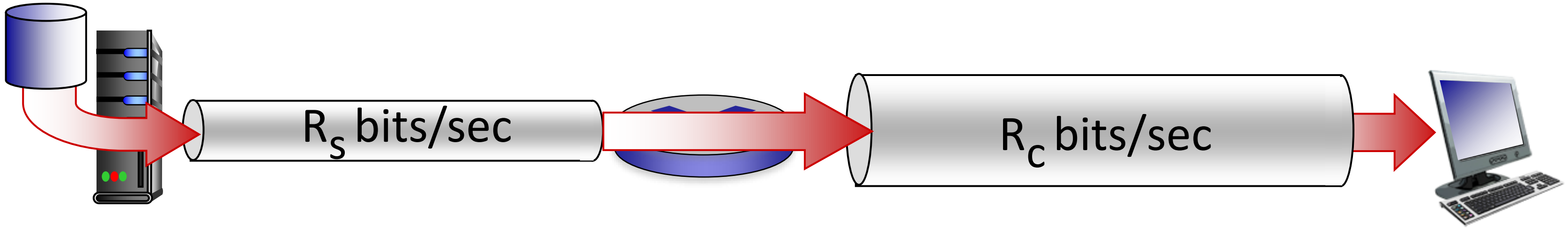
Throughput

- *throughput:* rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

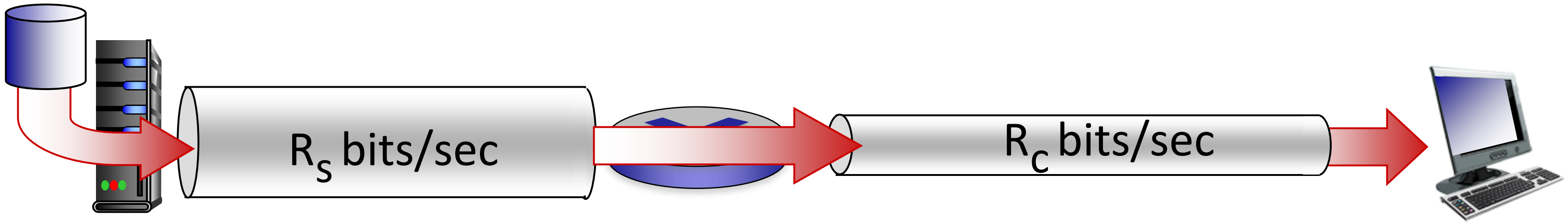


Throughput

$R_s < R_c$ What is average end-end throughput?



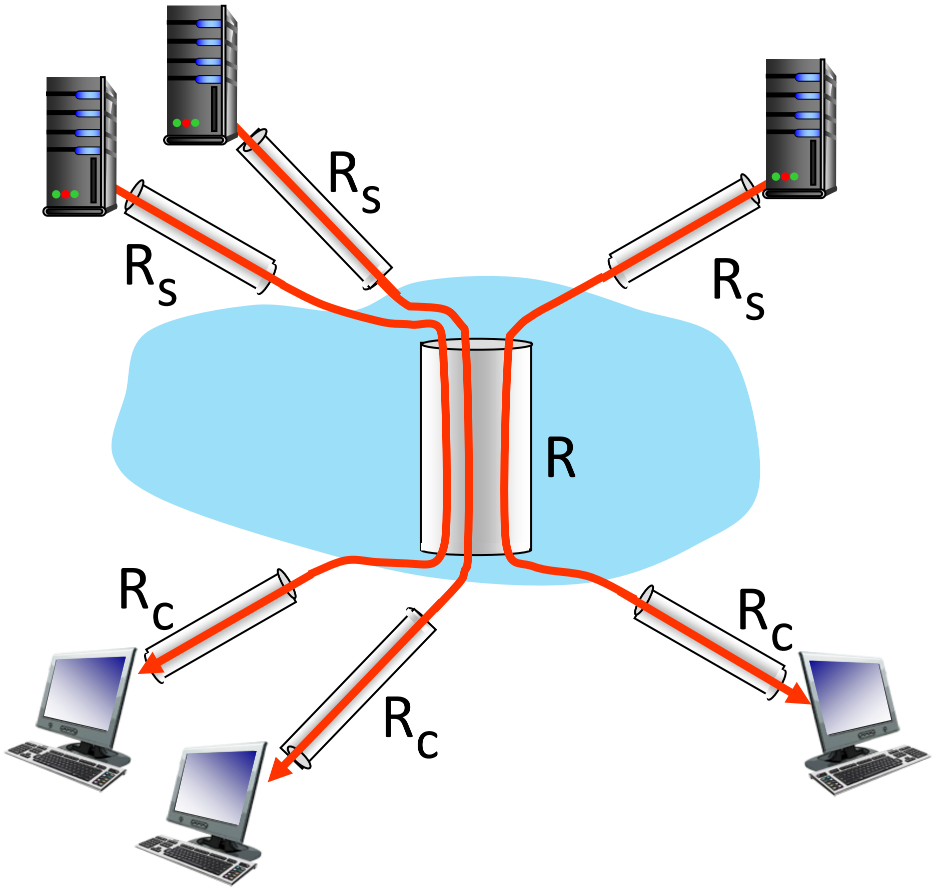
$R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput

Throughput: network scenario



10 connections (fairly) share backbone bottleneck link *R* bits/sec

- per-connection end-end throughput: $min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/

Store-and-forward delay

Very large packet sizes leads to poor performance

- Must wait at every router for whole packet to be received before passing it on

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/message-segmentation/index.html

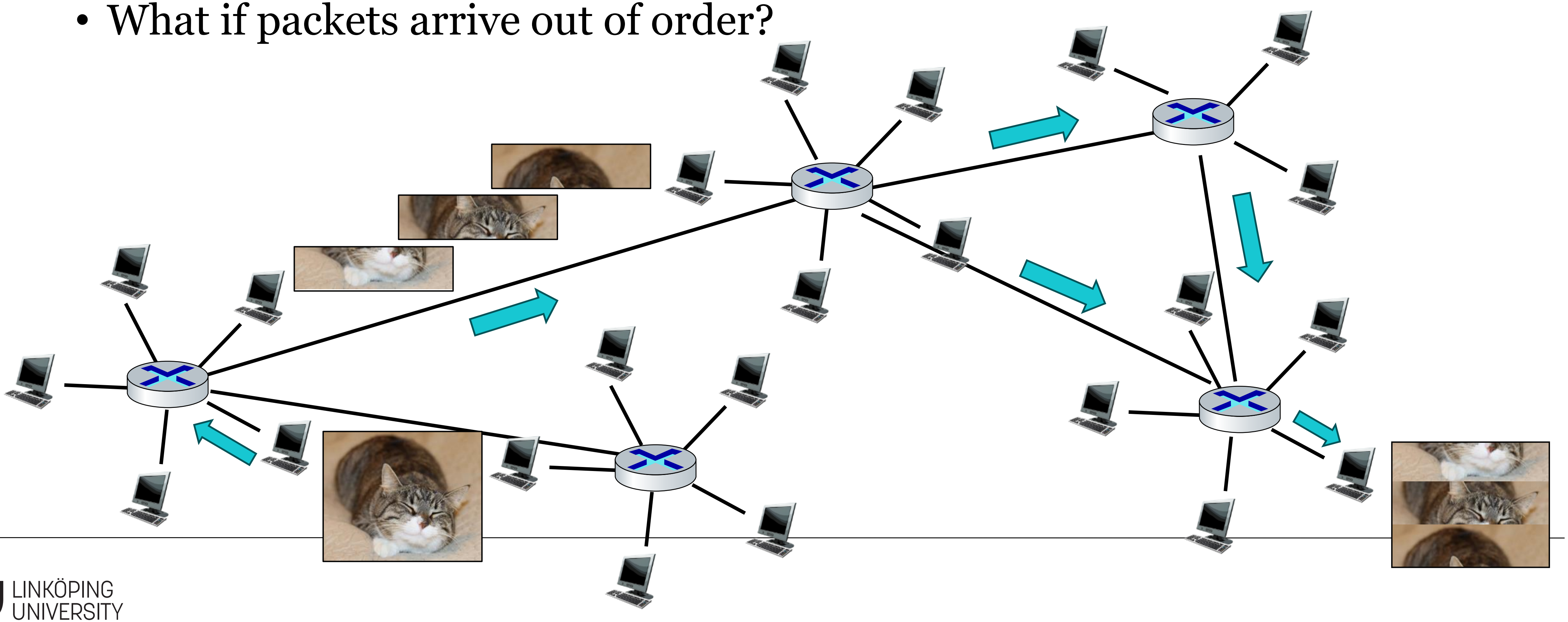
Split large messages into smaller packets

- Brings some new challenges…

Problems with packets...

Say we want to send a large chunk of data – like a cute cat picture

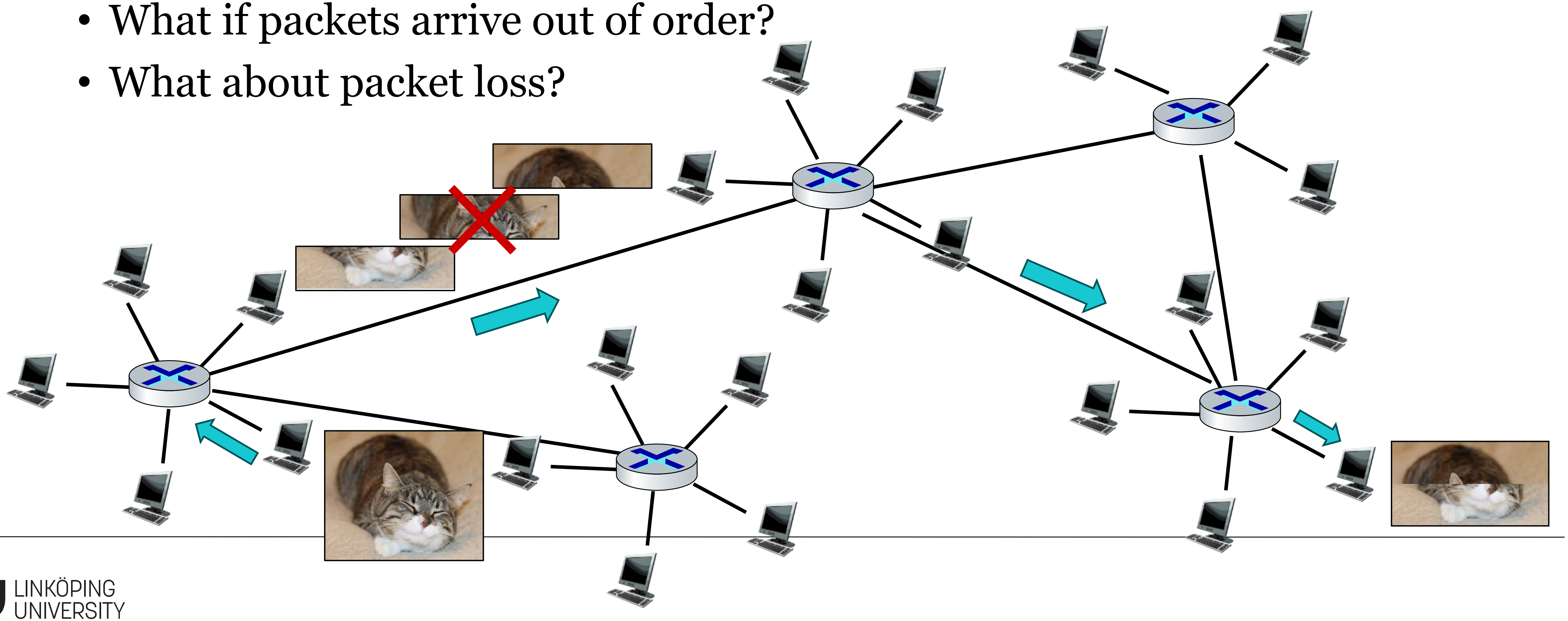
- Split into multiple network-layer packets to improve speed
  - What if packets arrive out of order?



Problems with packets…

Say we want to send a large chunk of data – like a cute cat picture

- Split into multiple network-layer packets to improve speed
  - What if packets arrive out of order?
  - What about packet loss?



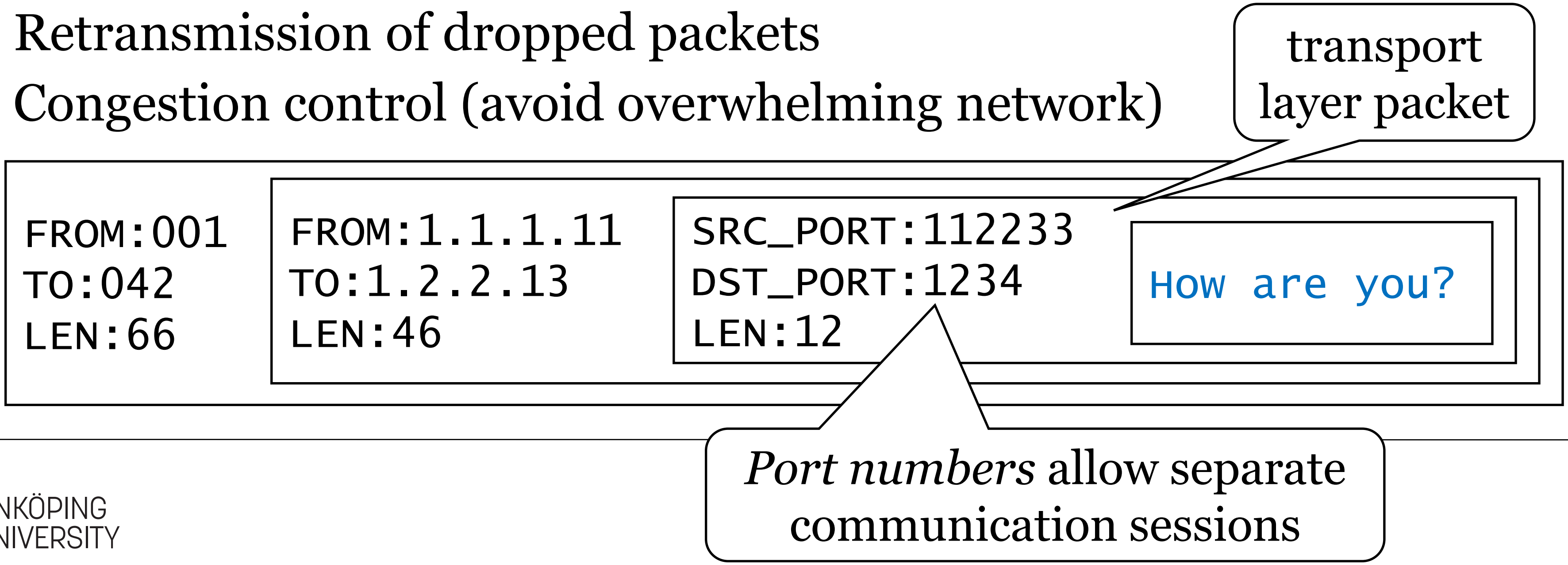
Problems with packets…

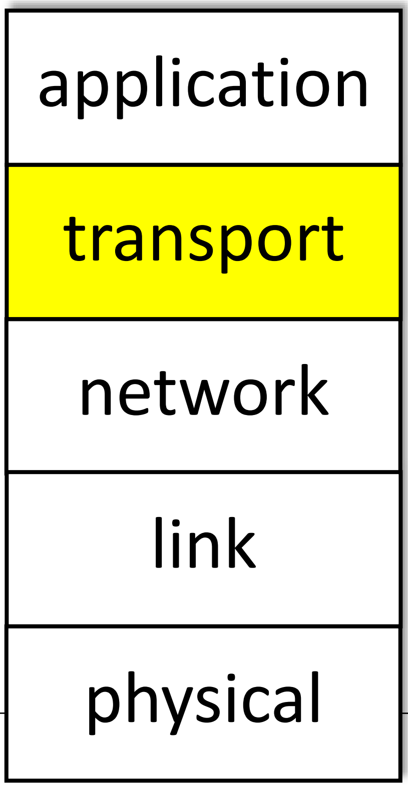
What if we want multiple networking apps running on the same machine

- For example: one for sending cat pictures and another for text messages?
- … or two concurrent chat sessions?

The *transport layer* handles all these problems (and others)

- Separate communication sessions on one machine
- In-order delivery of packets
- Retransmission of dropped packets
- Congestion control (avoid overwhelming network)

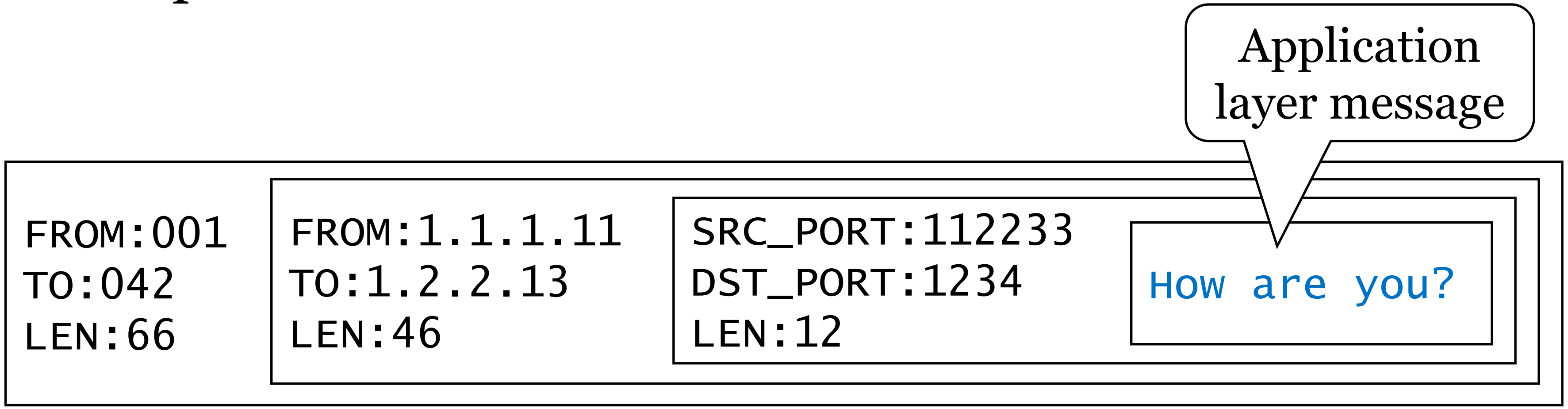


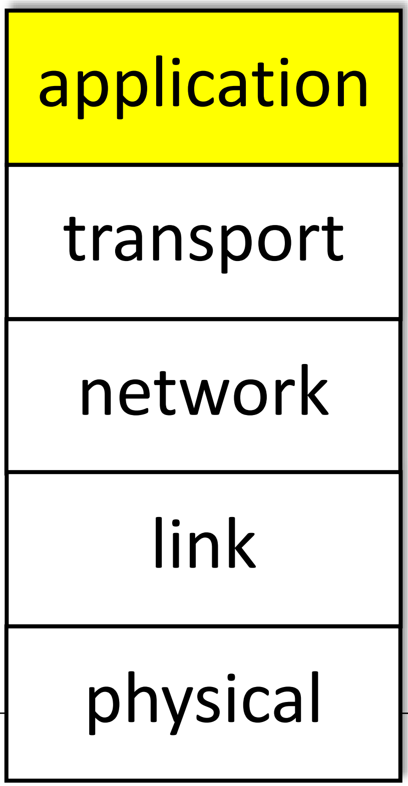


The final layer

The physical through transport layers are only concerned with getting data from point A to point B

- All applications of network communication are at (you guessed it): the *application layer*
- HTTP, DNS, IMAP (email), video streaming, etc.
- The topic of next lecture…





Putting it together

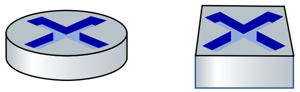
Now that we've seen the purpose of each layer, let's take a look at our current (pre-apocalypse) internet

application	HTTP, DNS, IMAP, SMTP, SSH, …
transport	TCP, UDP
network	IP, ICMP, routing protocols (OSPF, BGP, etc.)
link	MAC, ARP, …
physical	Ethernet, WiFi, Bluetooth, …

The Internet: a "nuts and bolts" view

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's "edge"



Packet switches: forward packets (chunks of data)

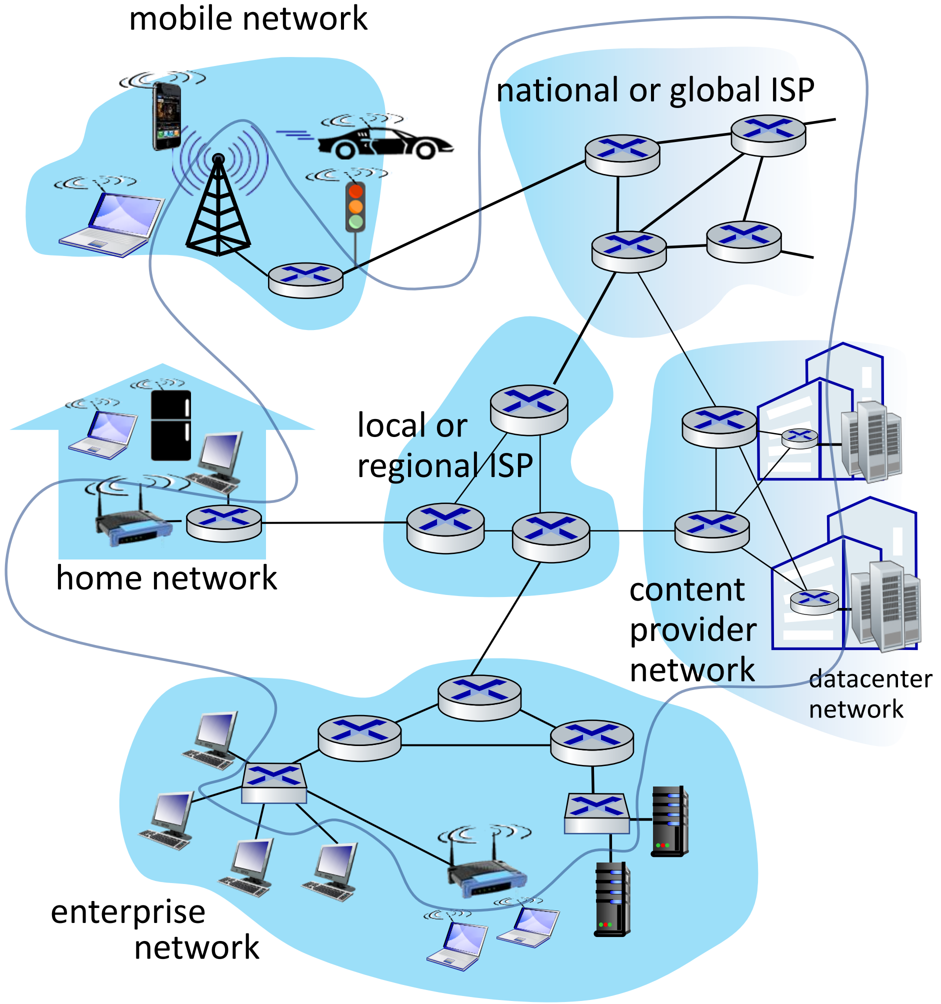
- *routers, switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

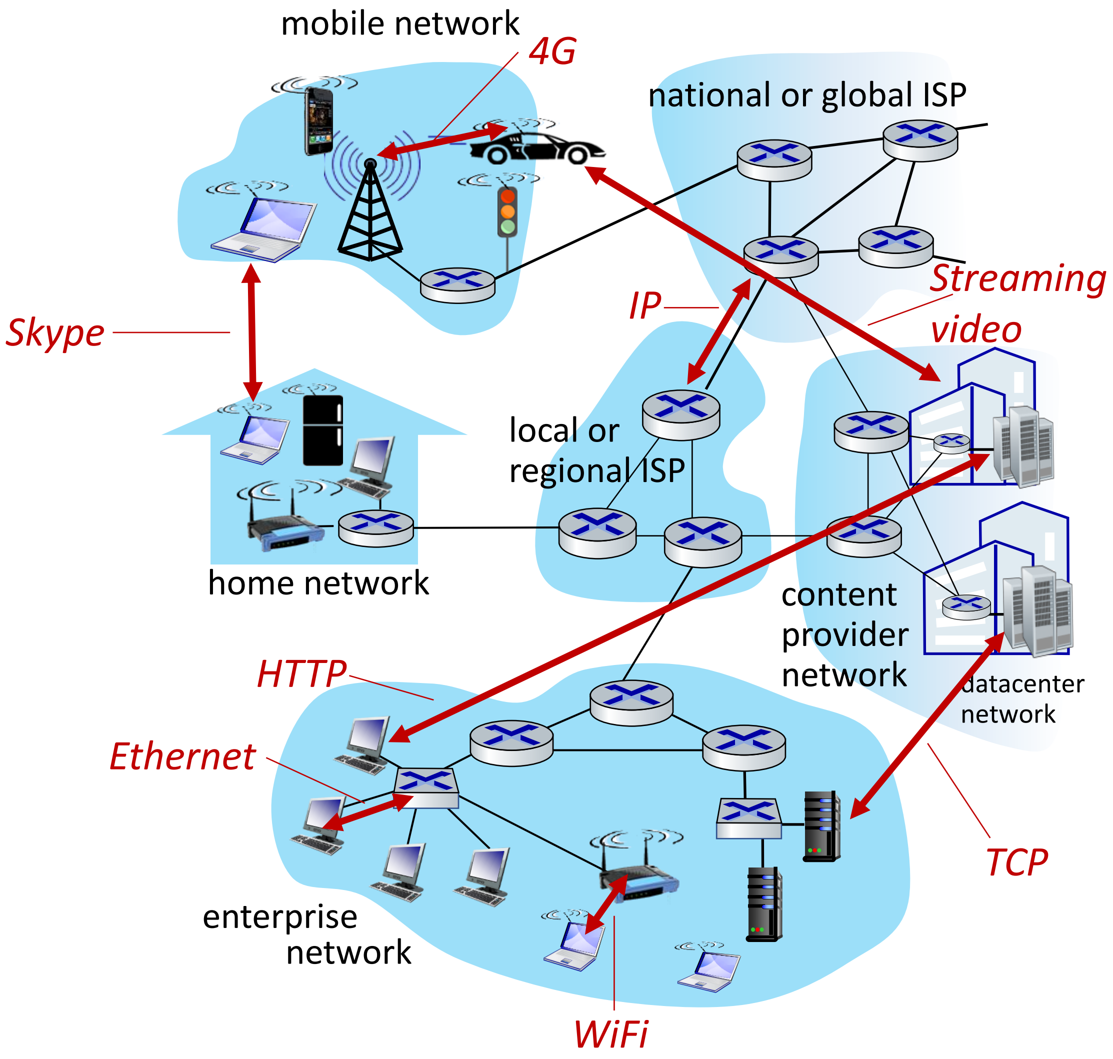
Networks

- collection of devices, routers, links: managed by an organization



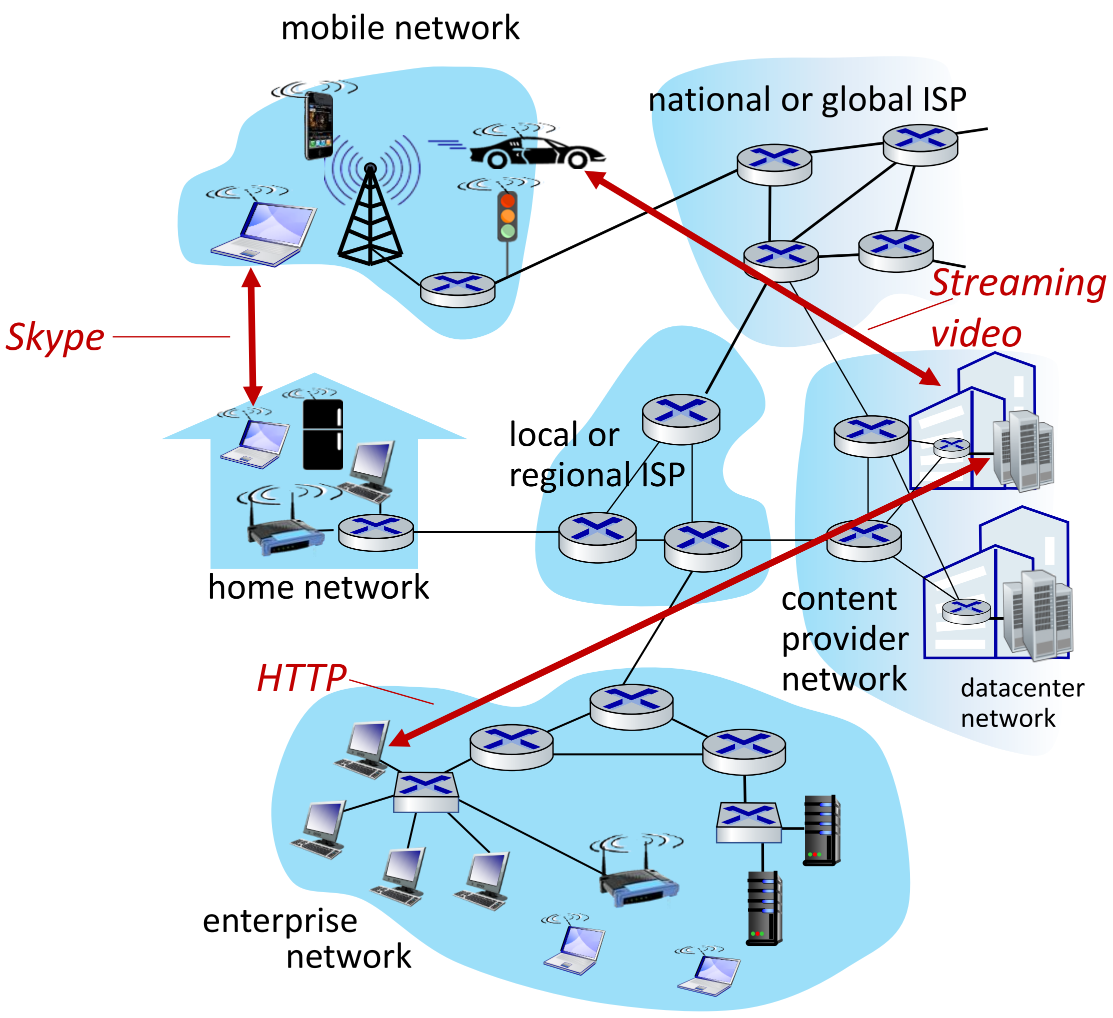
The Internet: a "nuts and bolts" view

- *Internet: "network of networks"*
  - Interconnected ISPs
- *protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet
- *Internet standards*
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force



The Internet: a "services" view

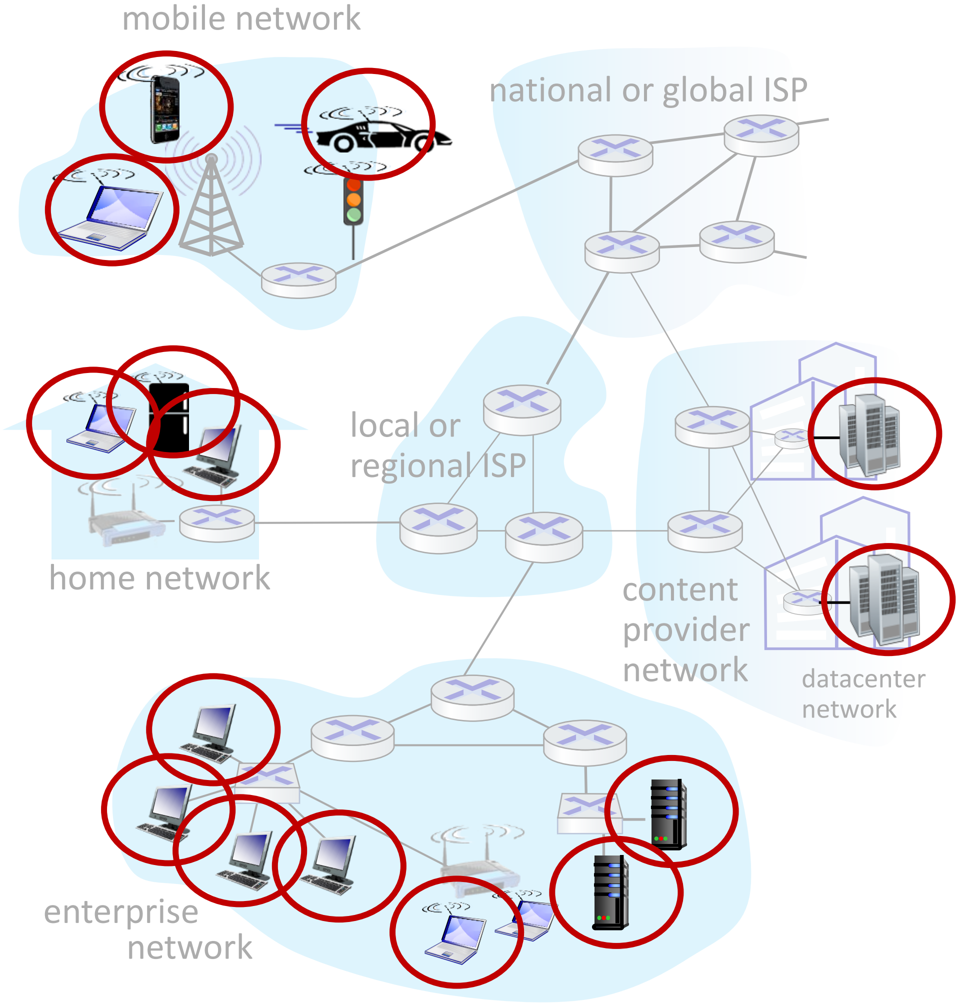
- *Infrastructure* that provides services to applications:
  - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, inter-connected appliances, …
- provides *programming interface* to distributed applications:
  - "hooks" allowing sending/receiving apps to "connect" to, use Internet transport service
  - provides service options, analogous to postal service



A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers



A closer look at Internet structure

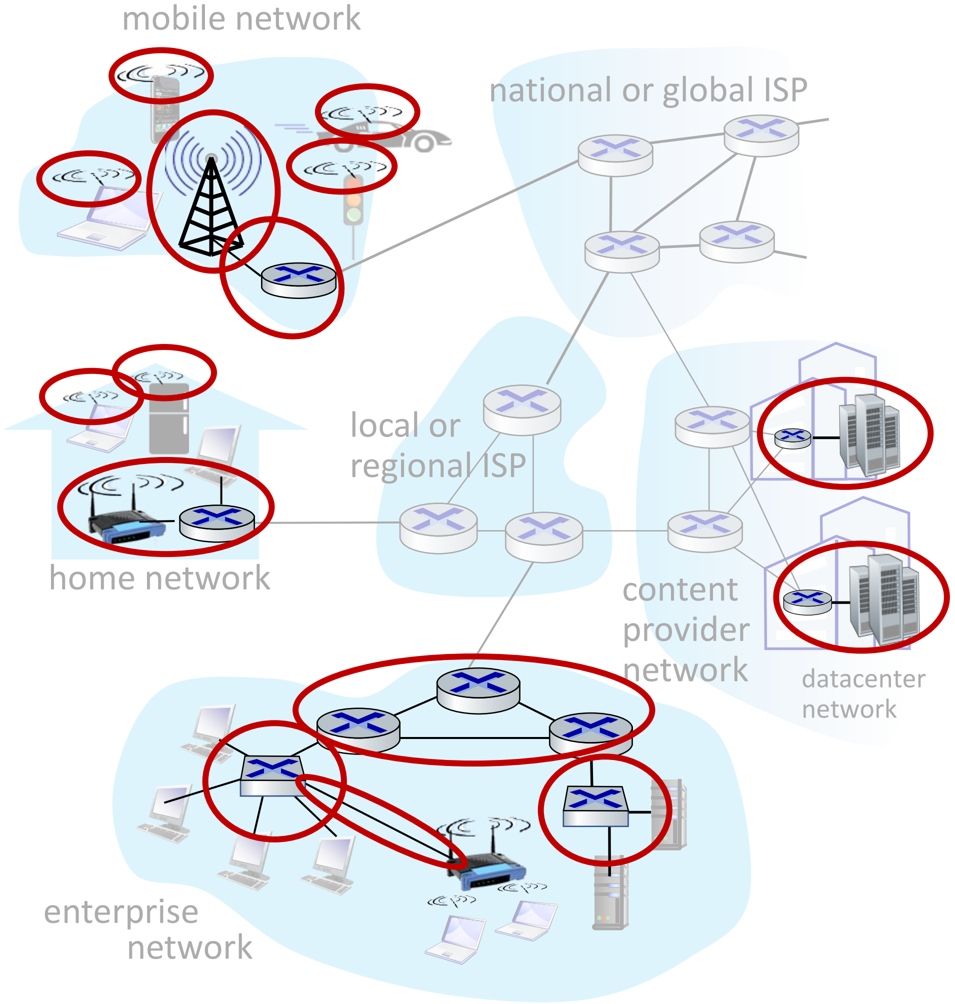
Network edge:

- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

Connects user devices to edge router
- Home WiFi router,
- 4G/5G connection,
- etc.



A closer look at Internet structure

Network edge:

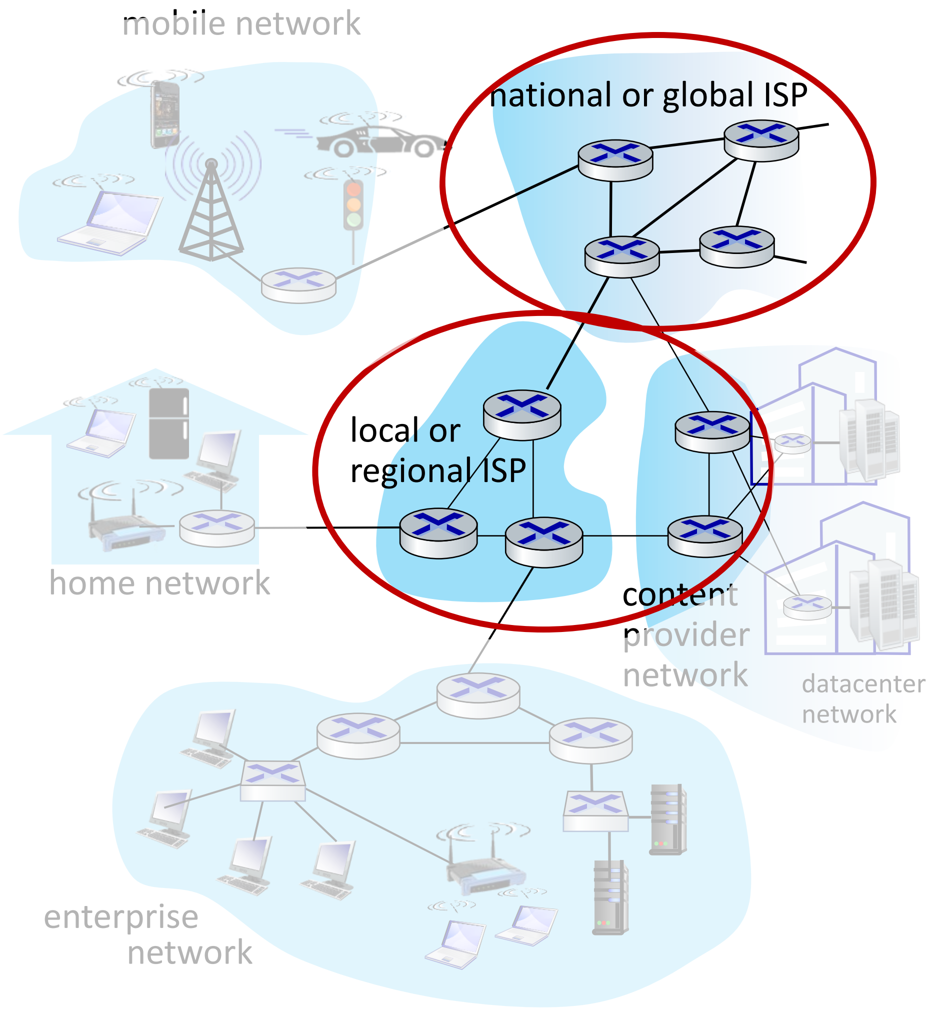
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

Network core:

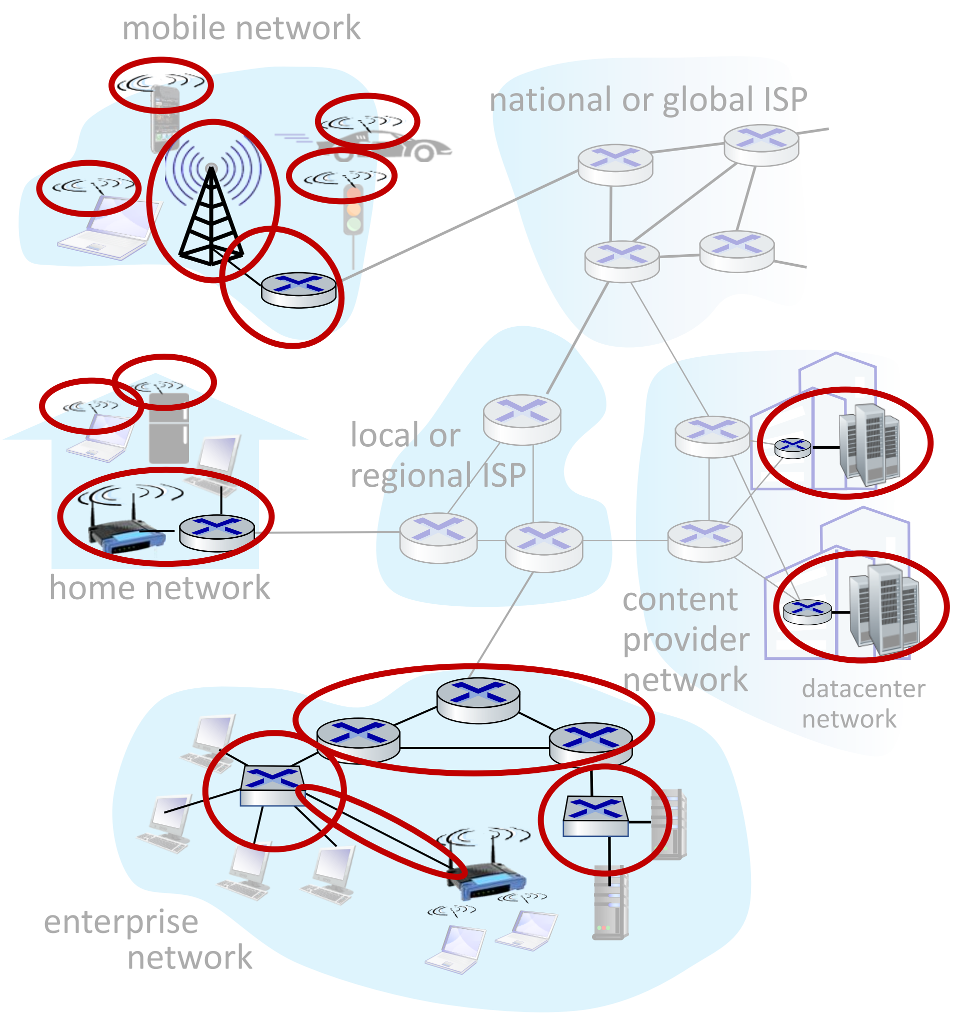
- interconnected routers
- network of networks



Access networks and physical media

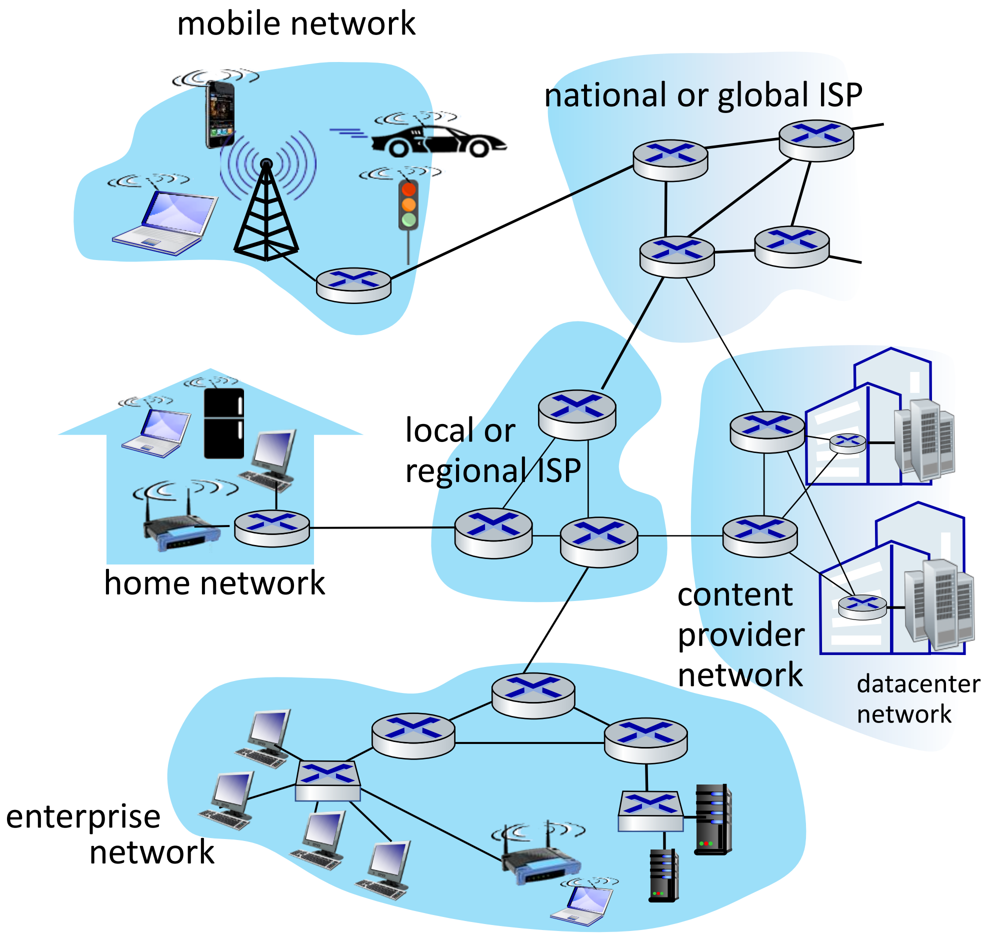
Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks (WiFi, 4G/5G)



Internet structure: a "network of networks"

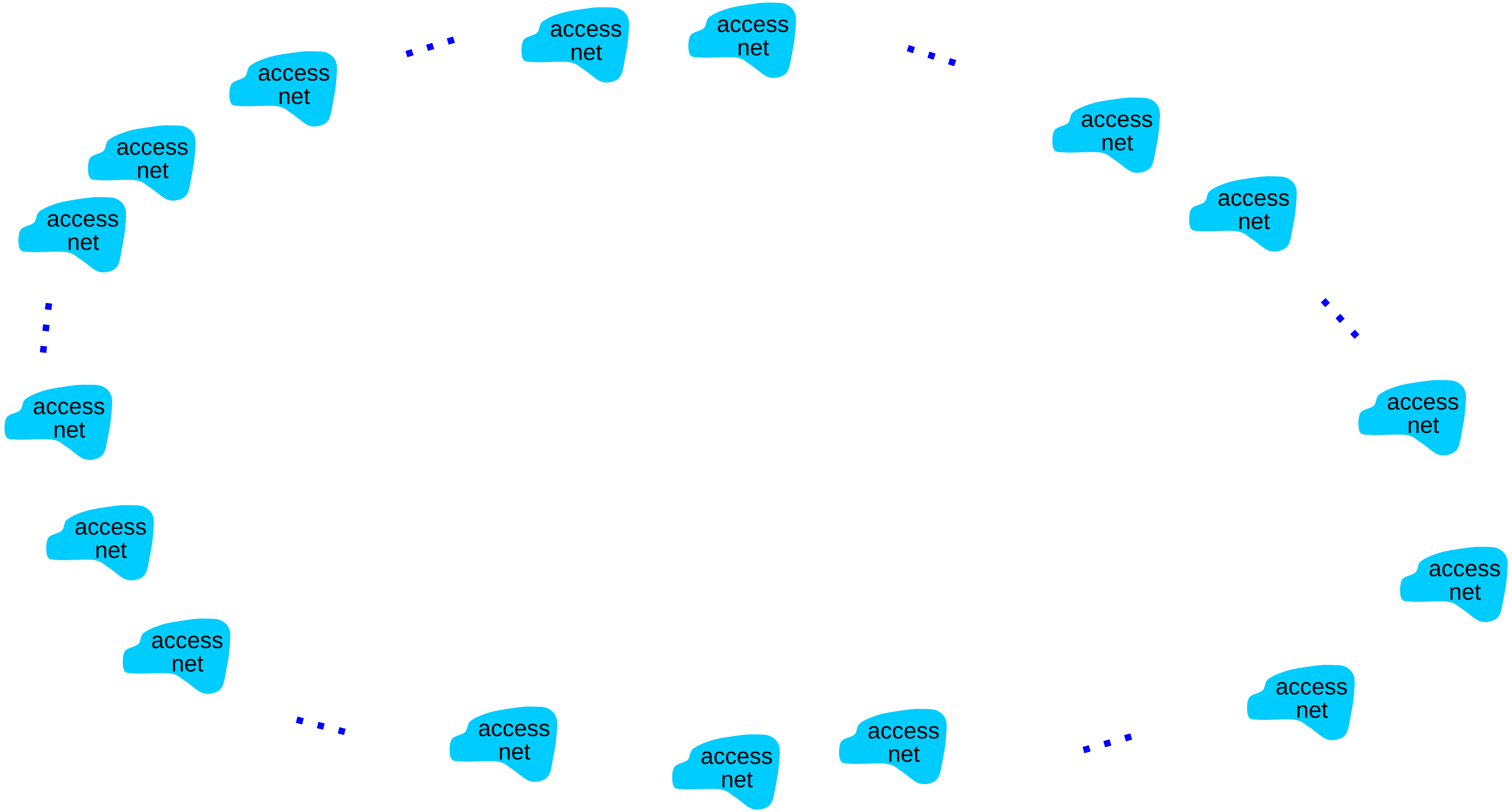
- hosts connect to Internet via access Internet Service Providers (ISPs)
- access ISPs in turn must be interconnected
  - so that *any* two hosts *(anywhere!)* can send packets to each other
- resulting network of networks is very complex
  - evolution driven by economics, national policies



Let's take a stepwise approach to describe current Internet structure

Internet structure: a "network of networks"

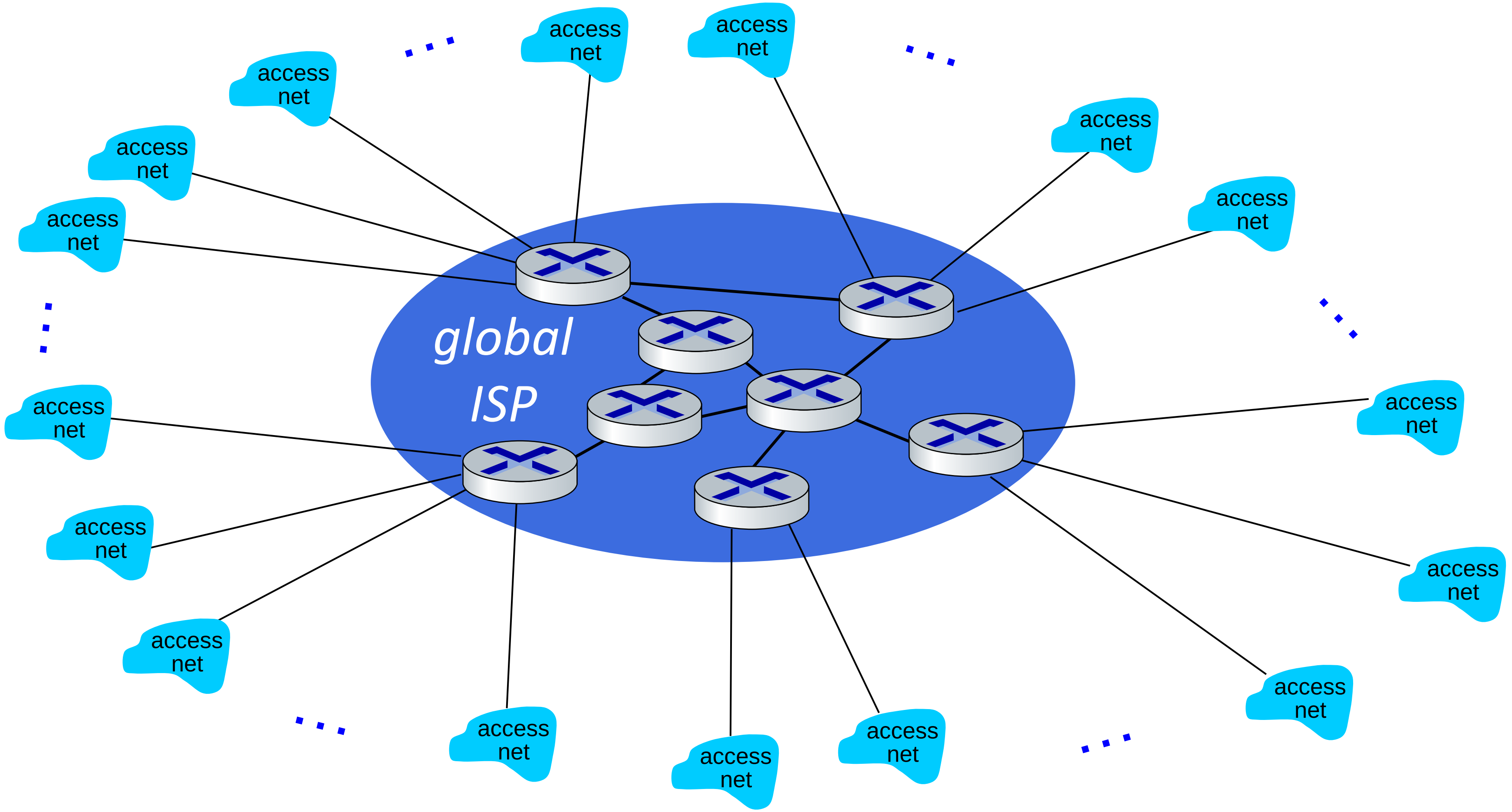
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

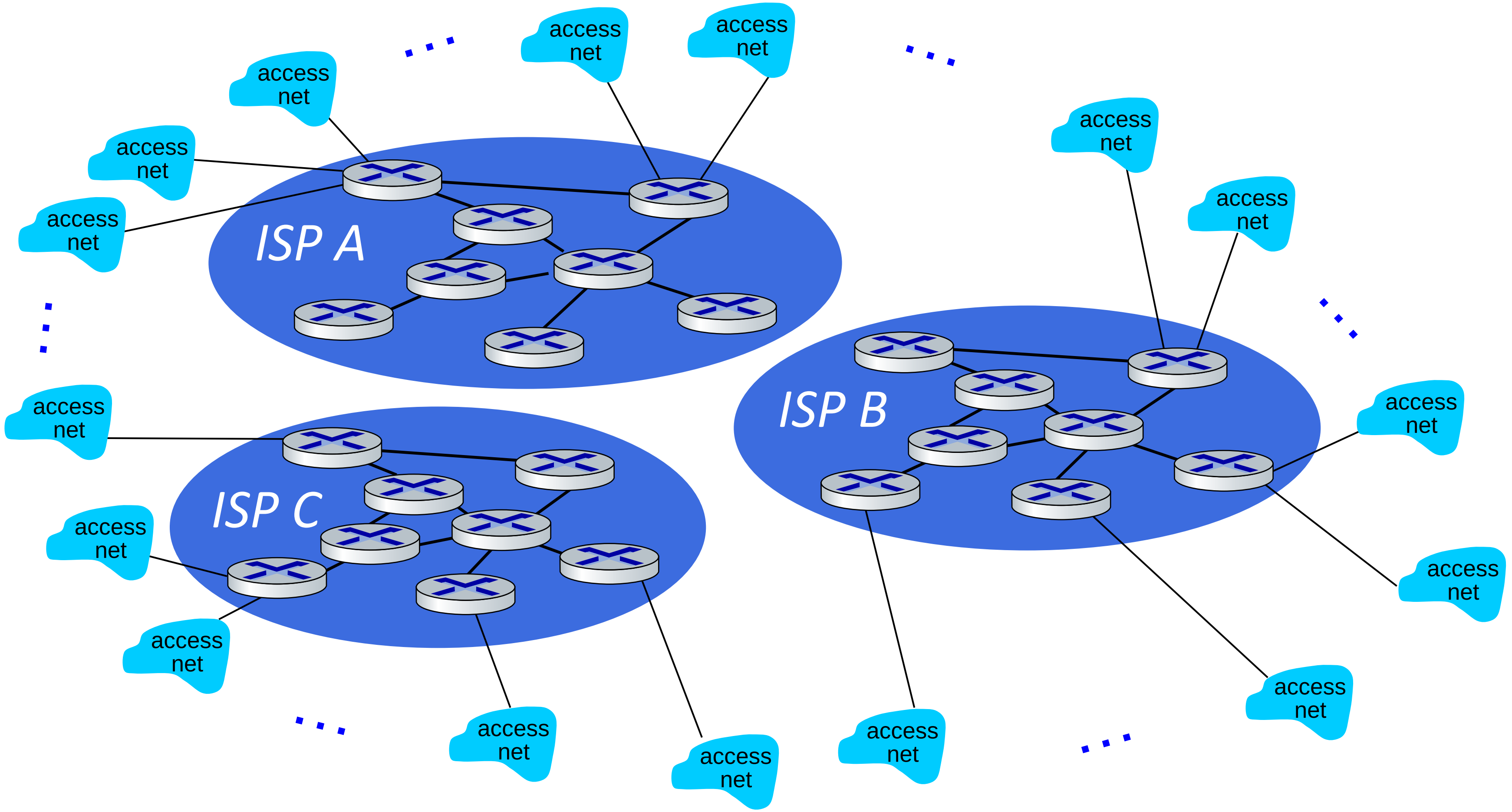
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



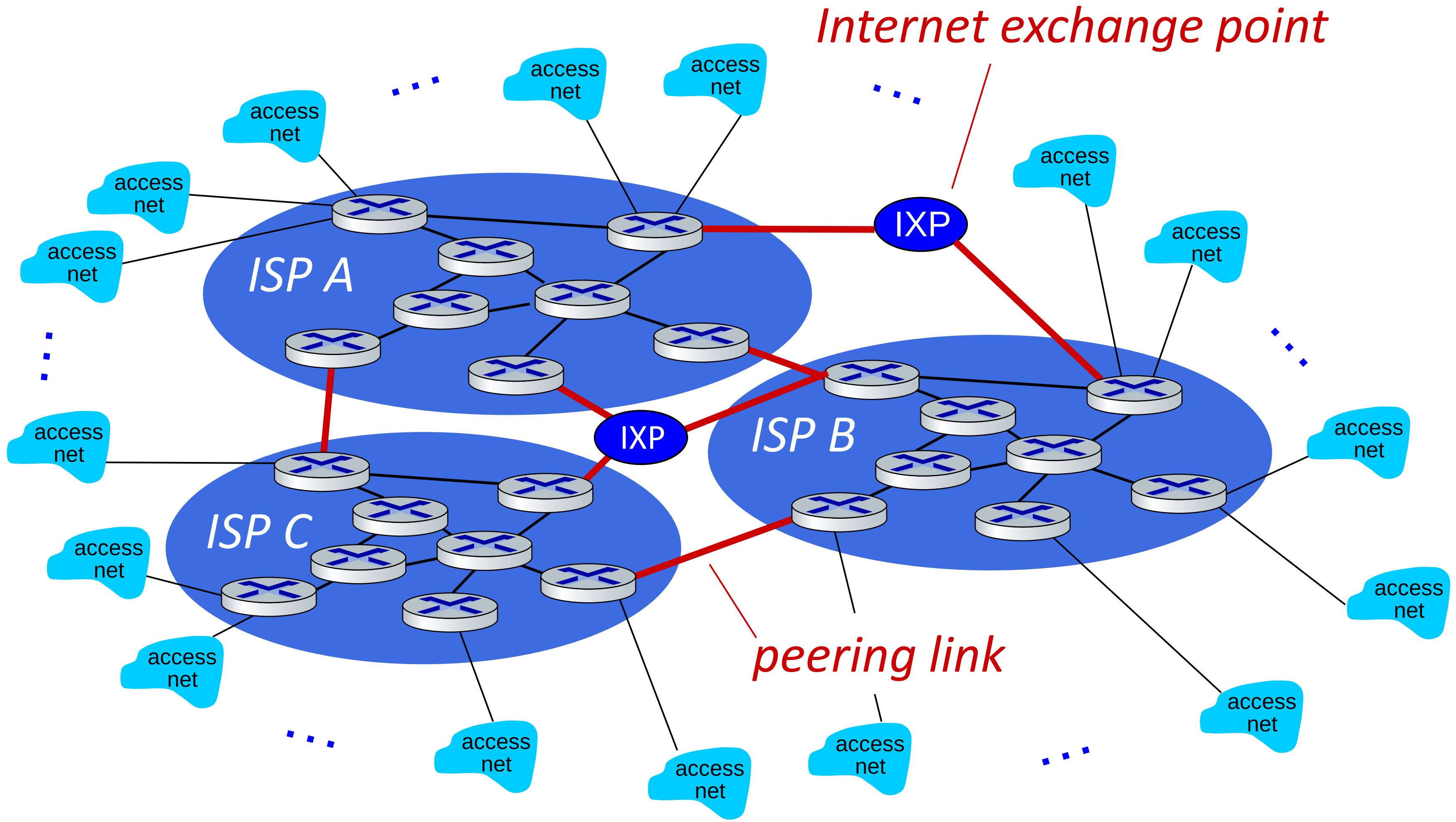
Internet structure: a "network of networks"

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



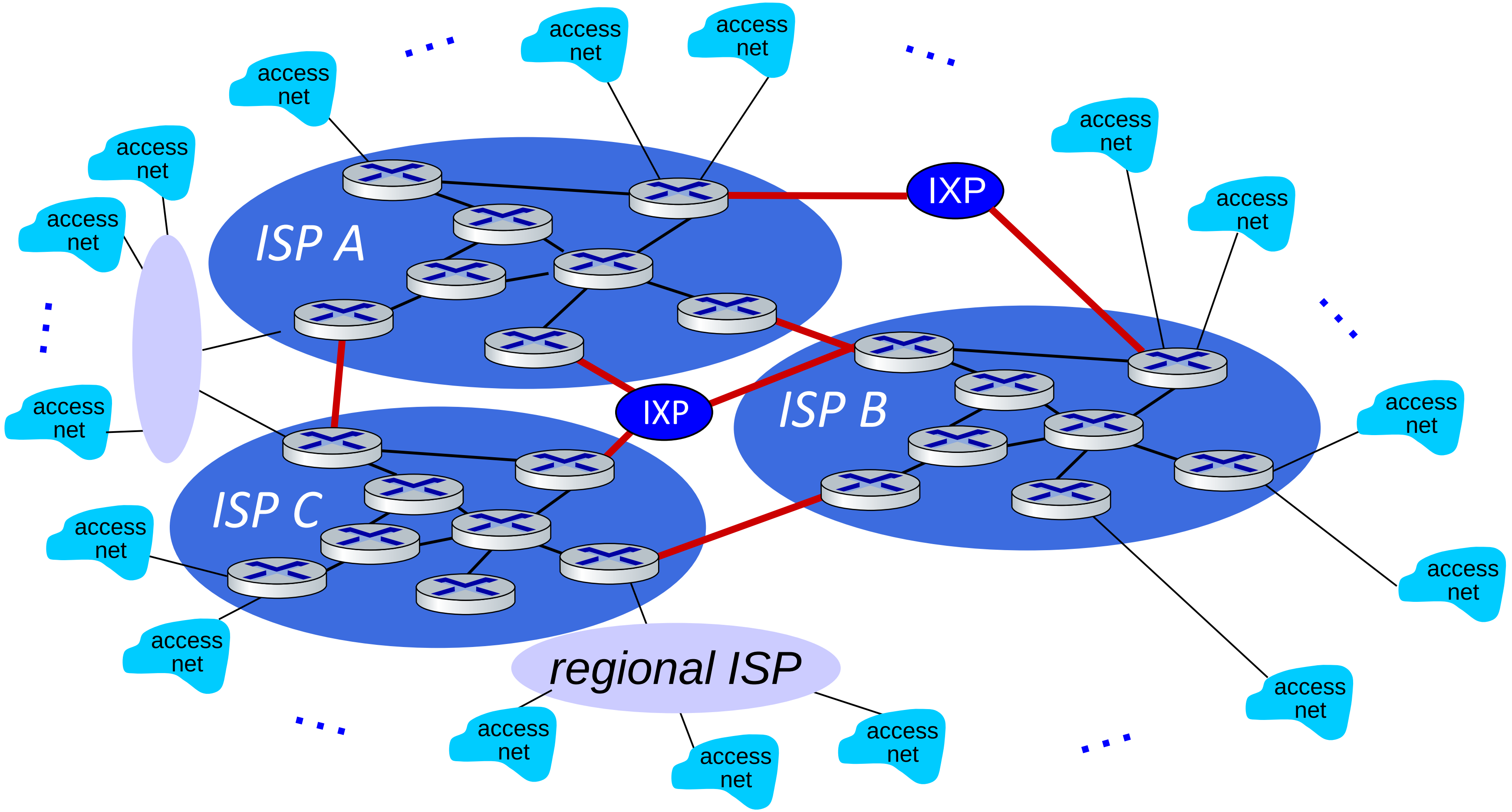
Internet structure: a "network of networks"

But if one global ISP is viable business, there will be competitors …. who will want to be connected



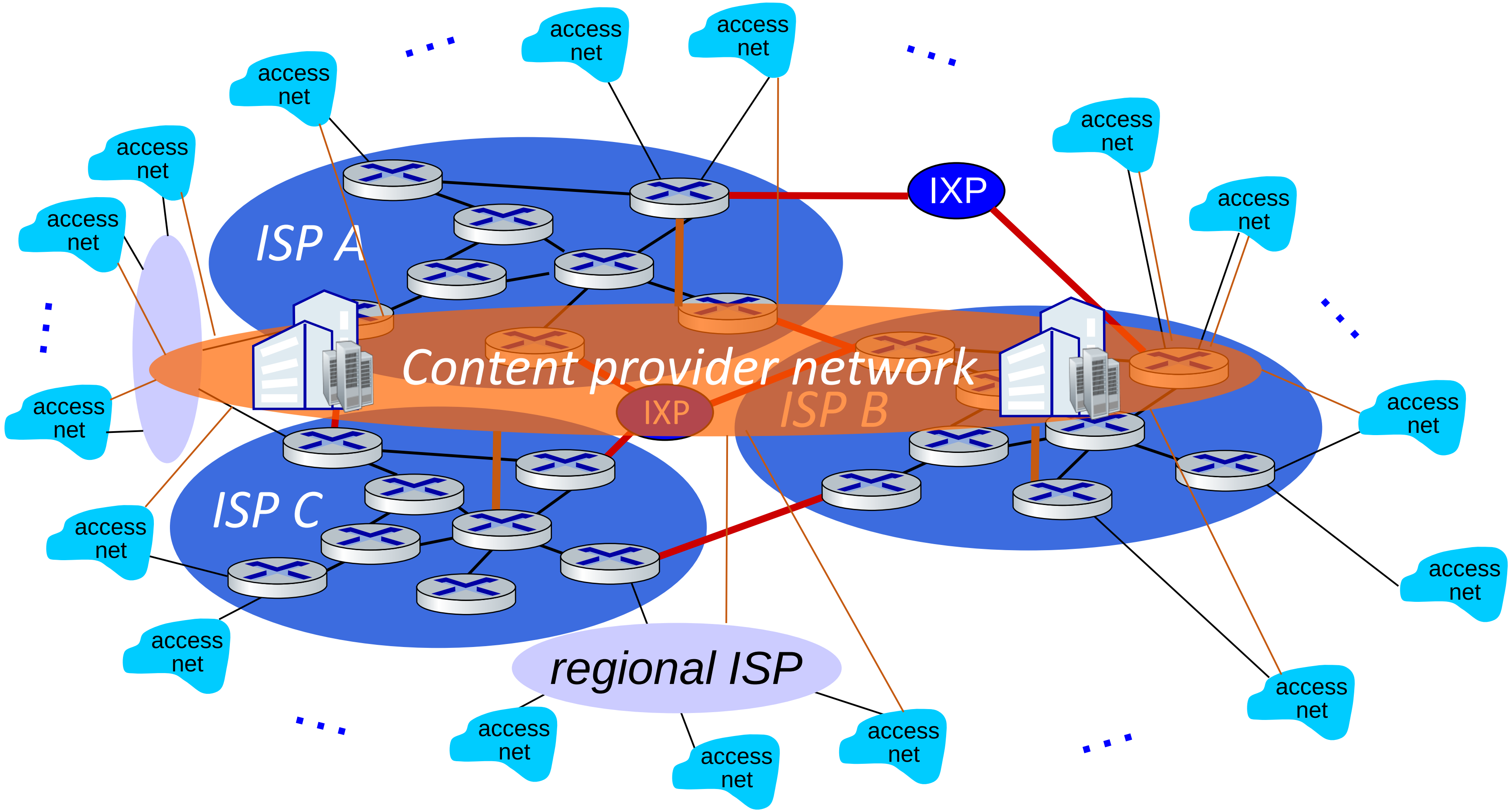
Internet structure: a "network of networks"

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

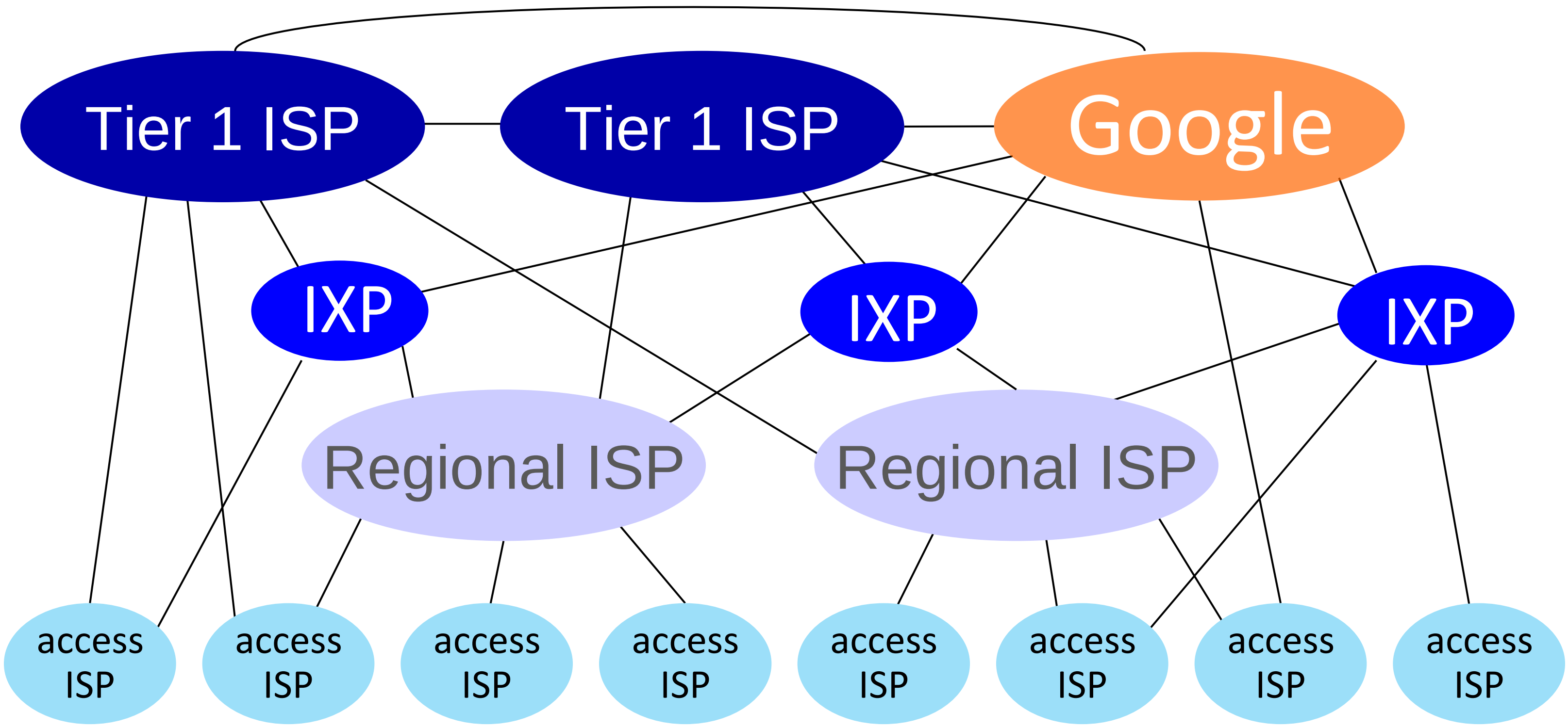


Internet structure: a "network of networks"

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



Internet structure: a "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, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

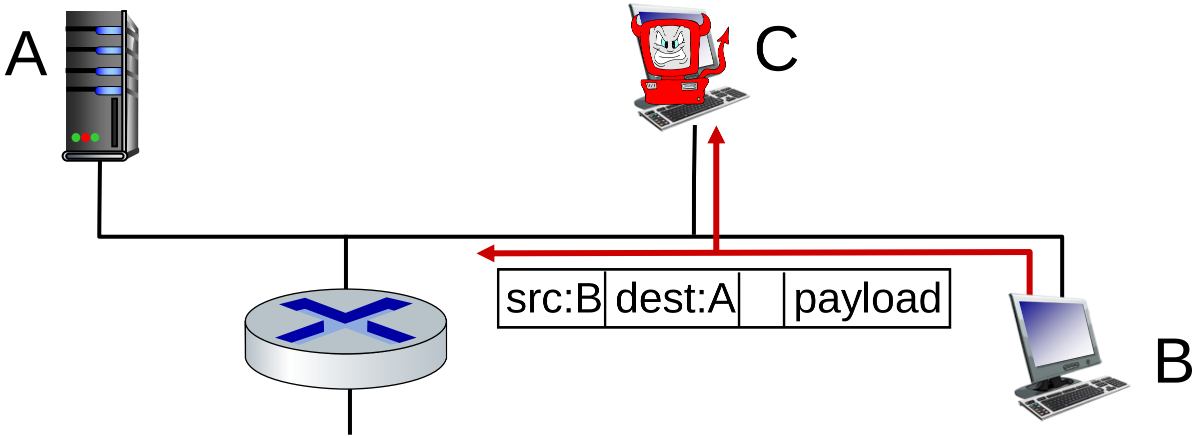
Network security

- Internet not originally designed with (much) security in mind
  - *original vision:* "a group of mutually trusting users attached to a transparent network" ☺
  - Internet protocol designers playing "catch-up"
  - security considerations in all layers!
- We now need to think about:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks

Bad guys: packet interception

packet "sniffing":

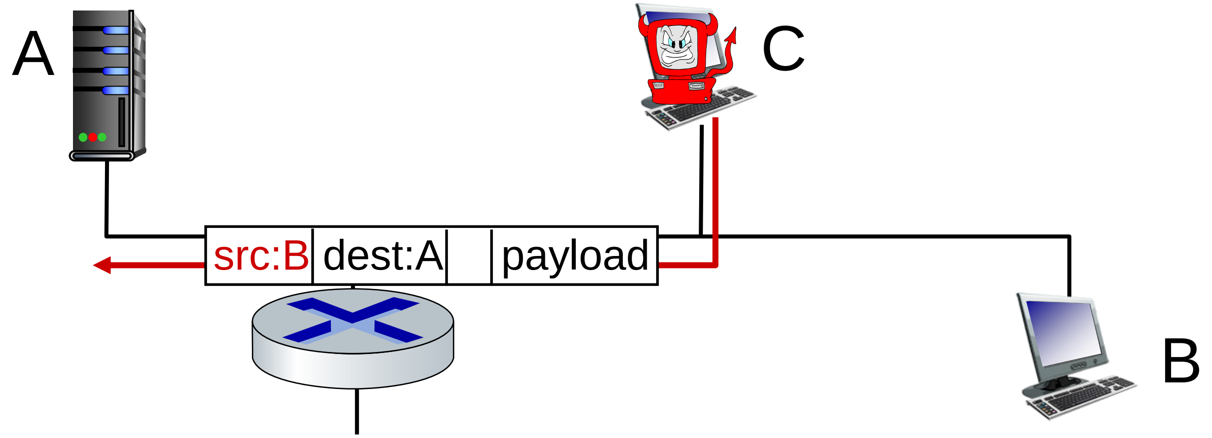
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



Wireshark software used for the labs is a (free) packet-sniffer

Bad guys: fake identity

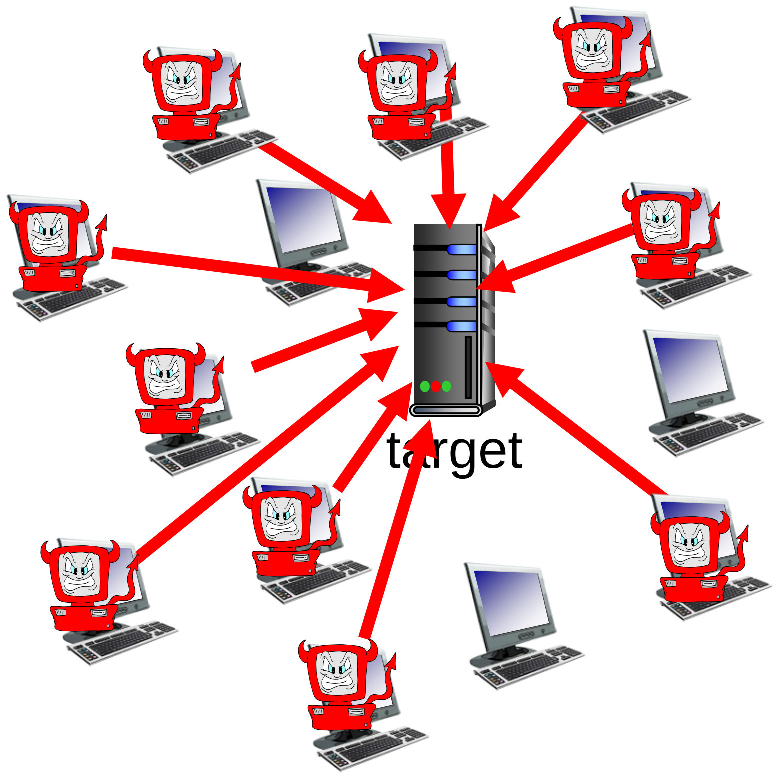
IP spoofing: injection of packet with false source address



Bad guys: denial of service

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Lines of defense:

- authentication: proving you are who you say you are
  - cellular networks provides hardware identity via SIM card; no such hardware assist in traditional Internet
- confidentiality: via encryption
- integrity checks: digital signatures prevent/detect tampering
- access restrictions: password-protected VPNs
- firewalls: specialized "middleboxes" in access and core networks:
  - off-by-default: filter incoming packets to restrict senders, receivers, applications
  - detecting/reacting to DOS attacks

… lots more on security (throughout, Chapter 8)