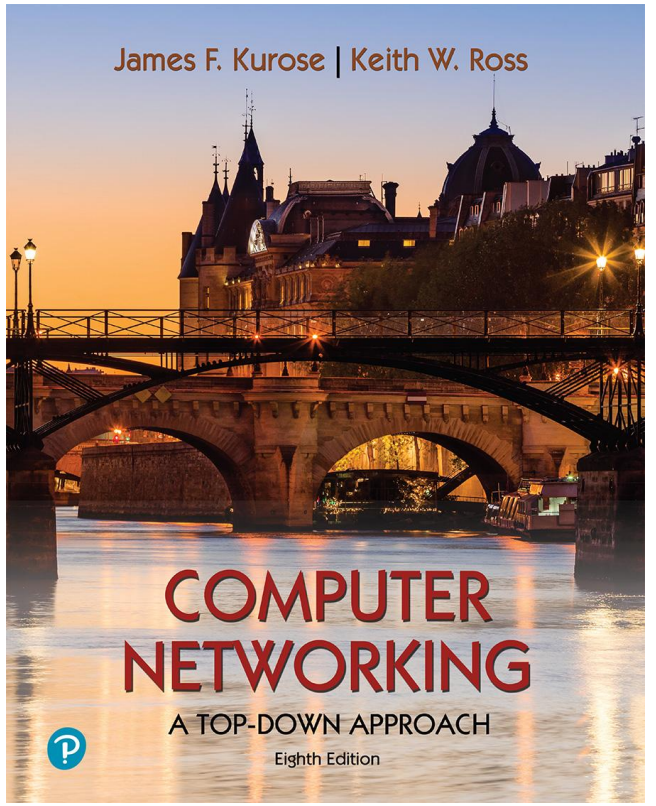
Chapter 1 Introduction

Chuming Qiao/Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Adapted from the slides of the book's authors

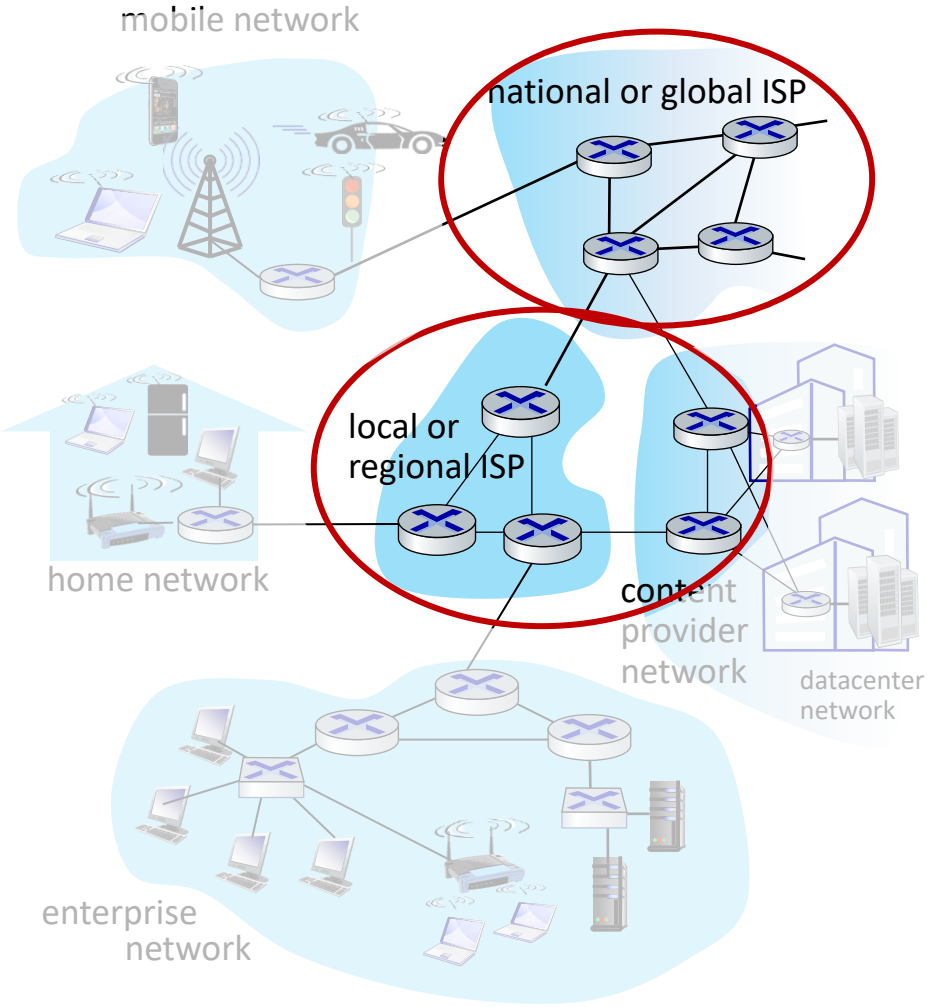
Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- **Network core:** internet structure, routing and forwarding
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

The network core

- mesh of interconnected routers

Ok, but WHY?

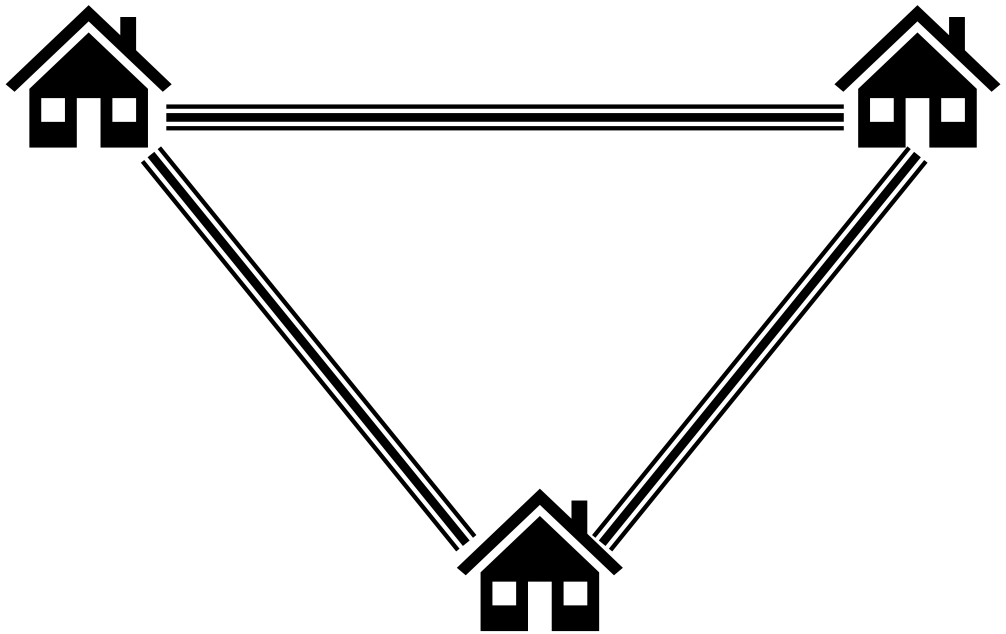


The network core

- mesh of interconnected routers

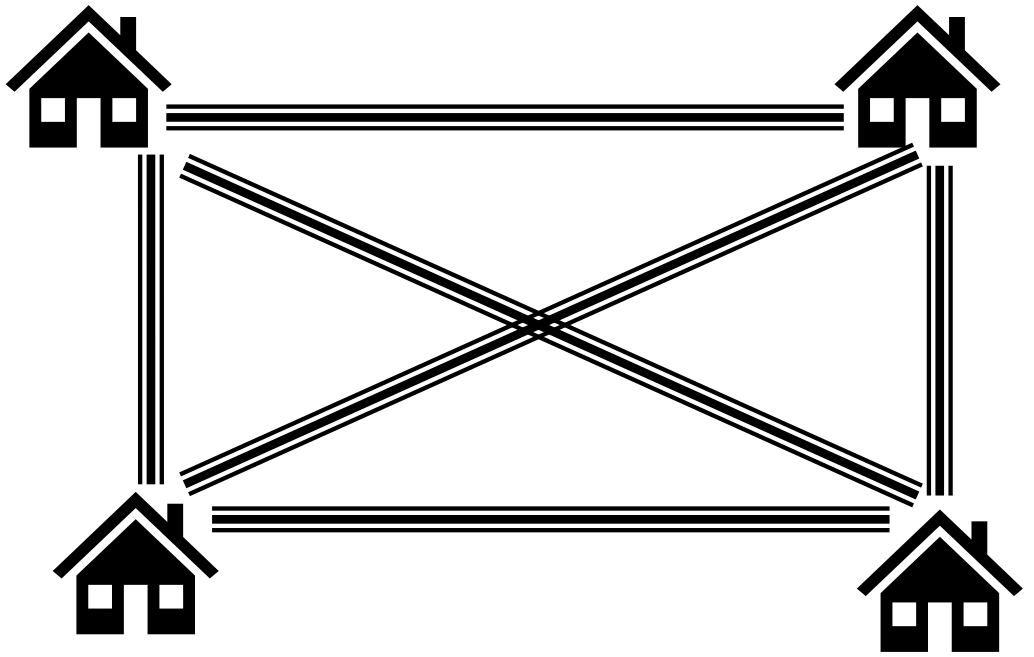
The network core

- mesh of interconnected routers



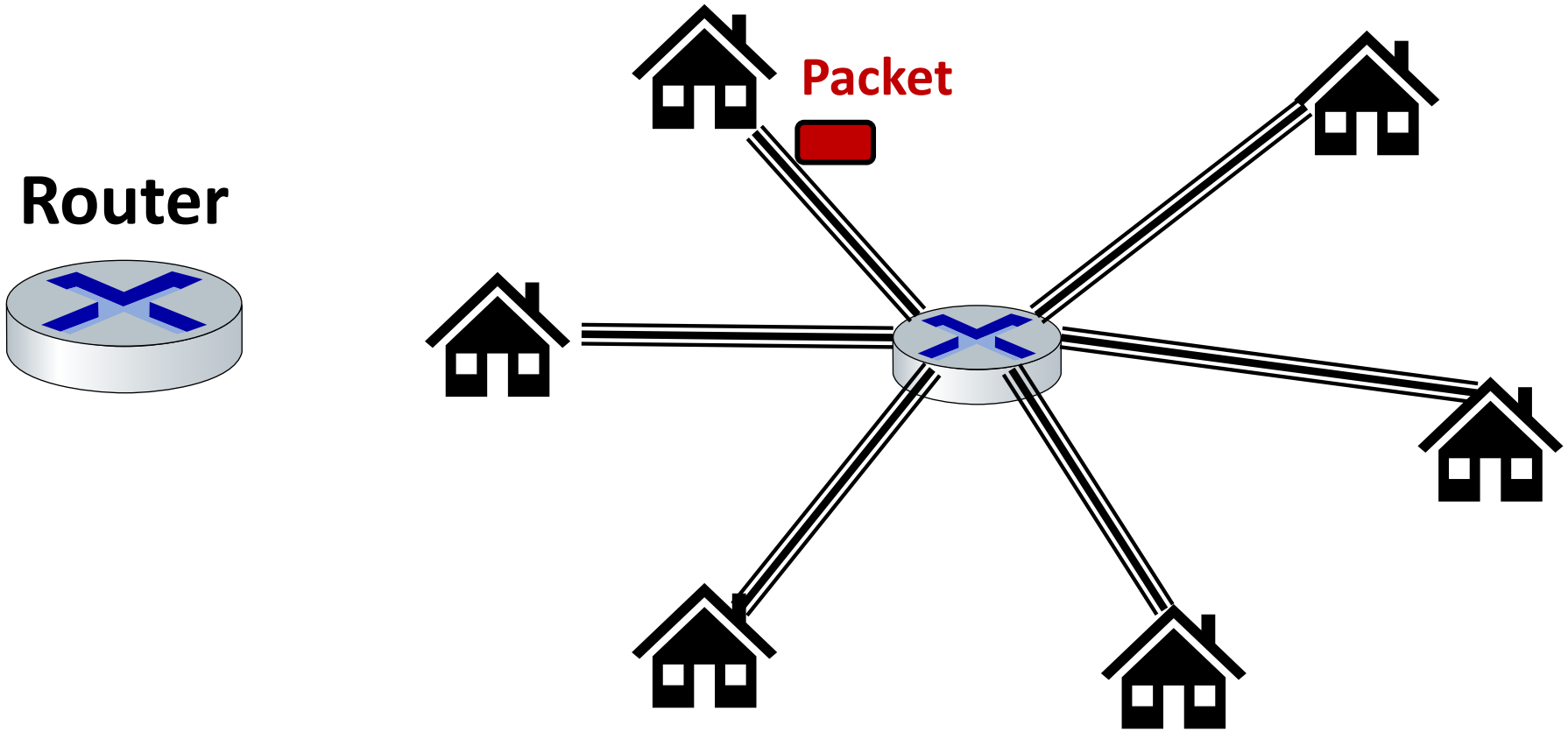
The network core

- mesh of interconnected routers



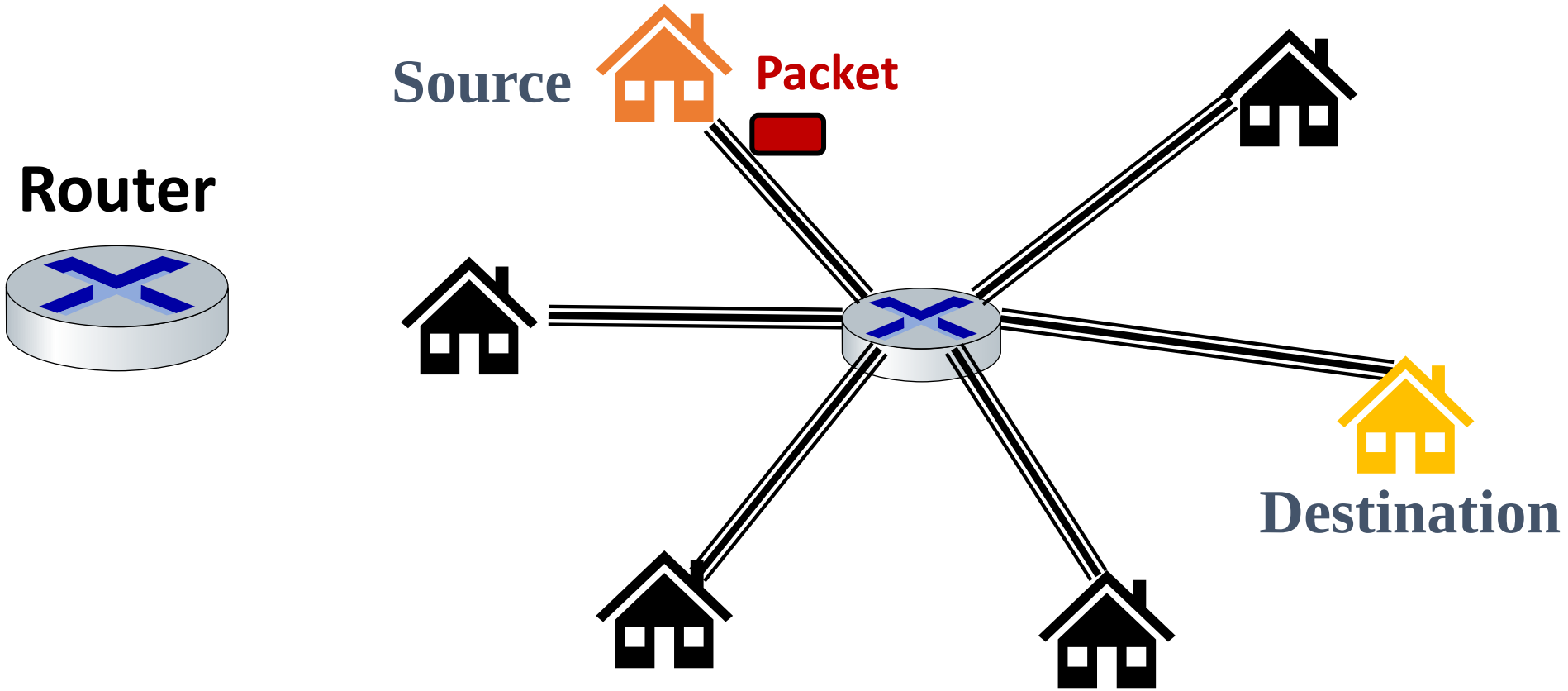
The network core

- mesh of interconnected routers



The network core

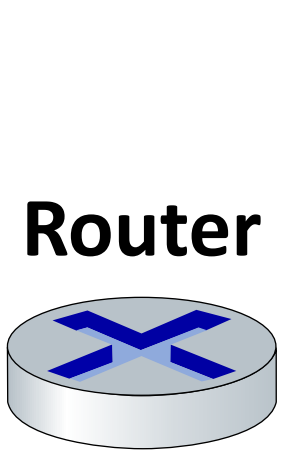
- mesh of interconnected routers

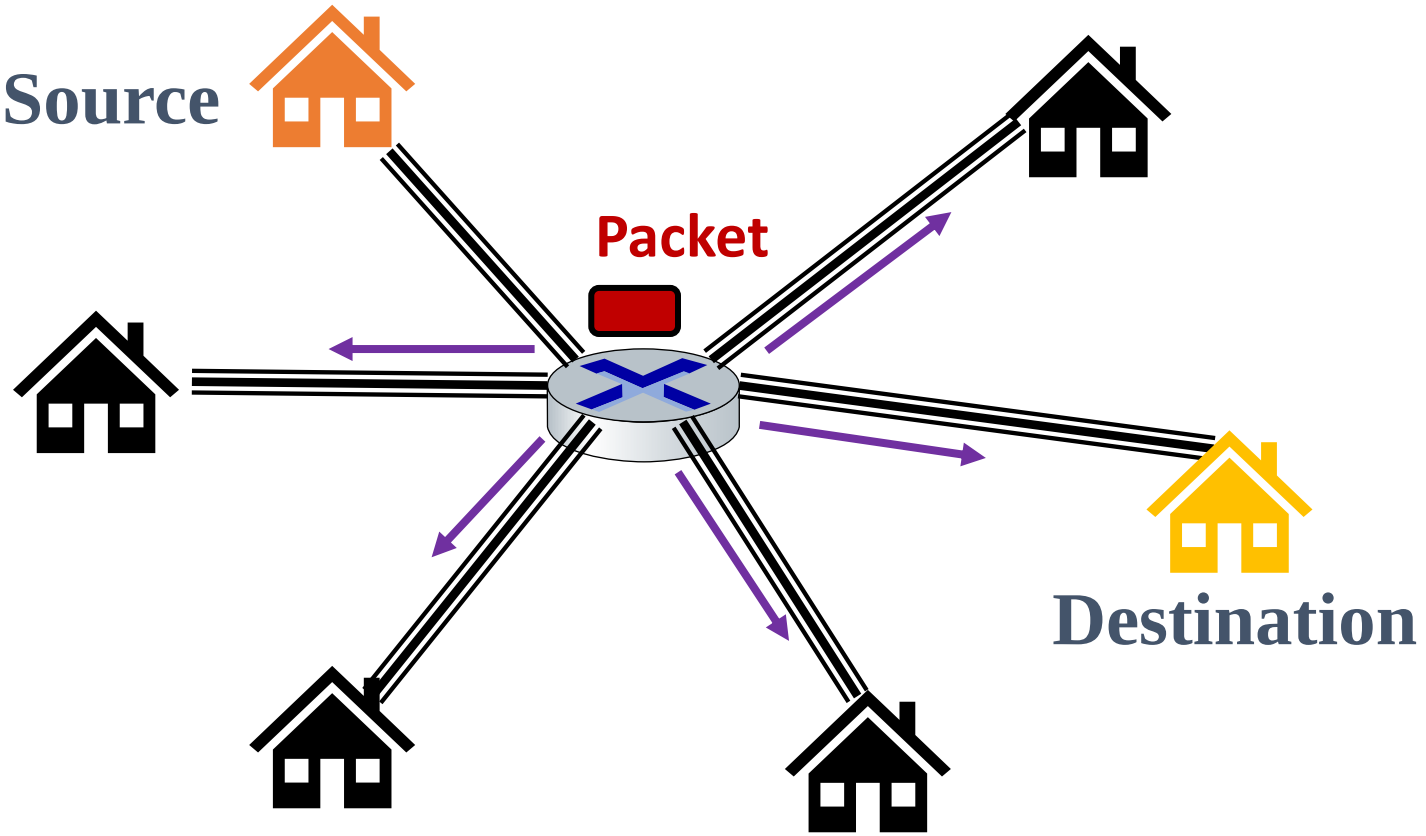


The network core

- mesh of interconnected routers

Router



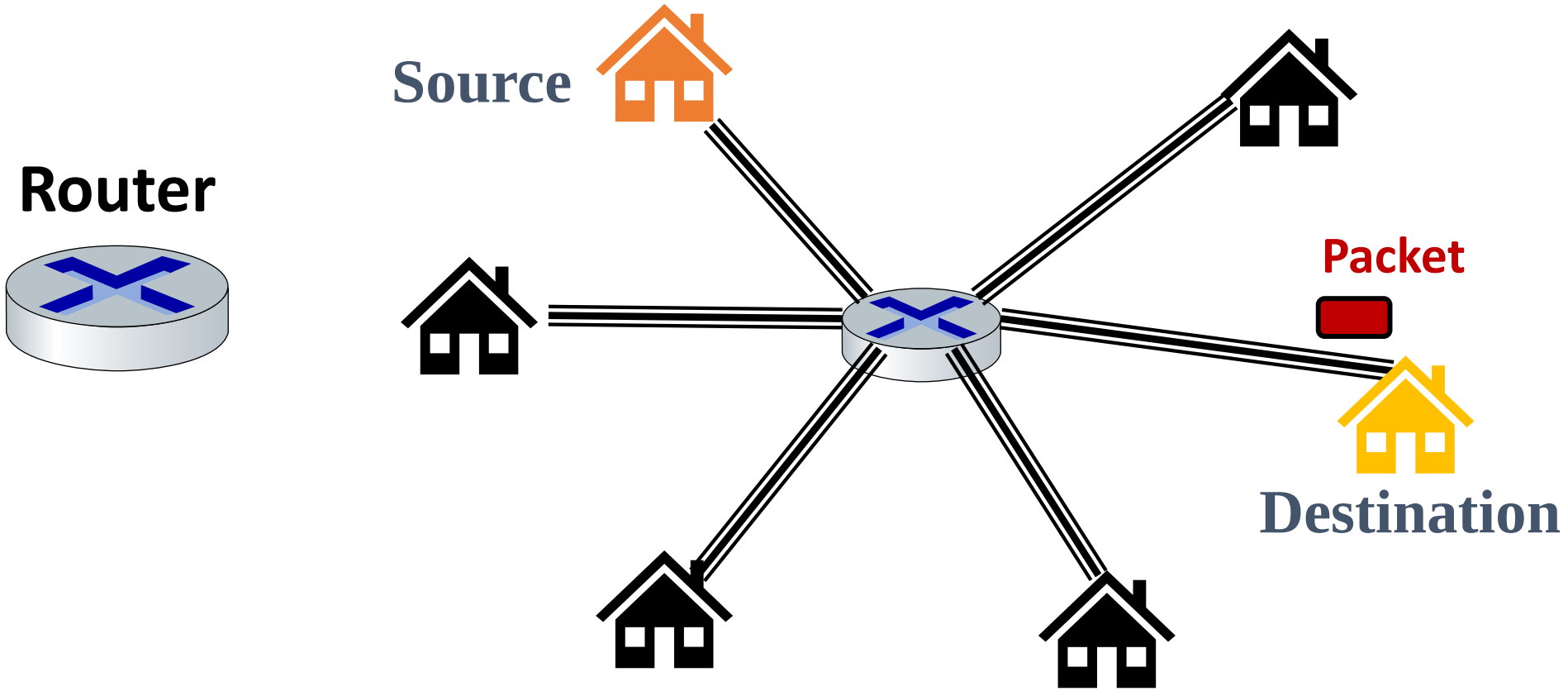


Forwarding:

- aka "switching"
- *local* action: move arriving packets from router's input link to appropriate router output link

The network core

- mesh of interconnected routers

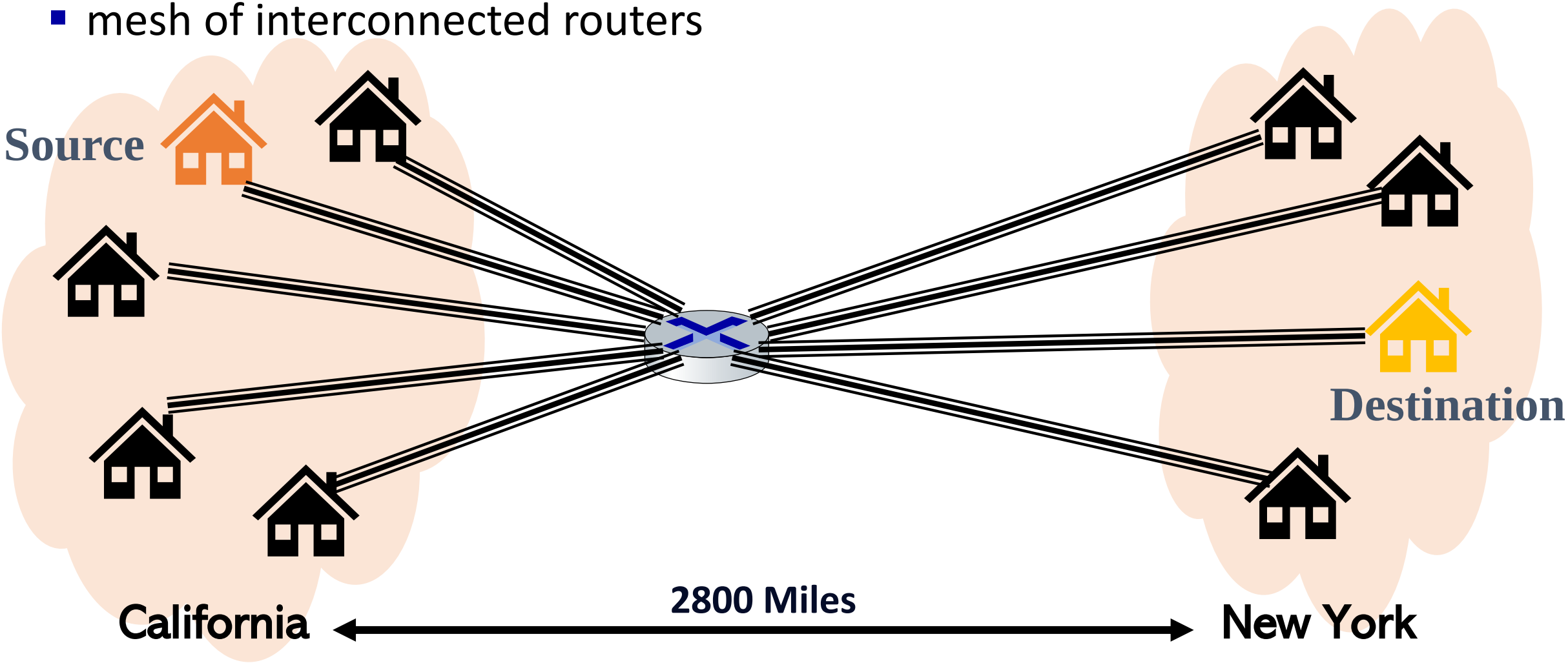


Forwarding:

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

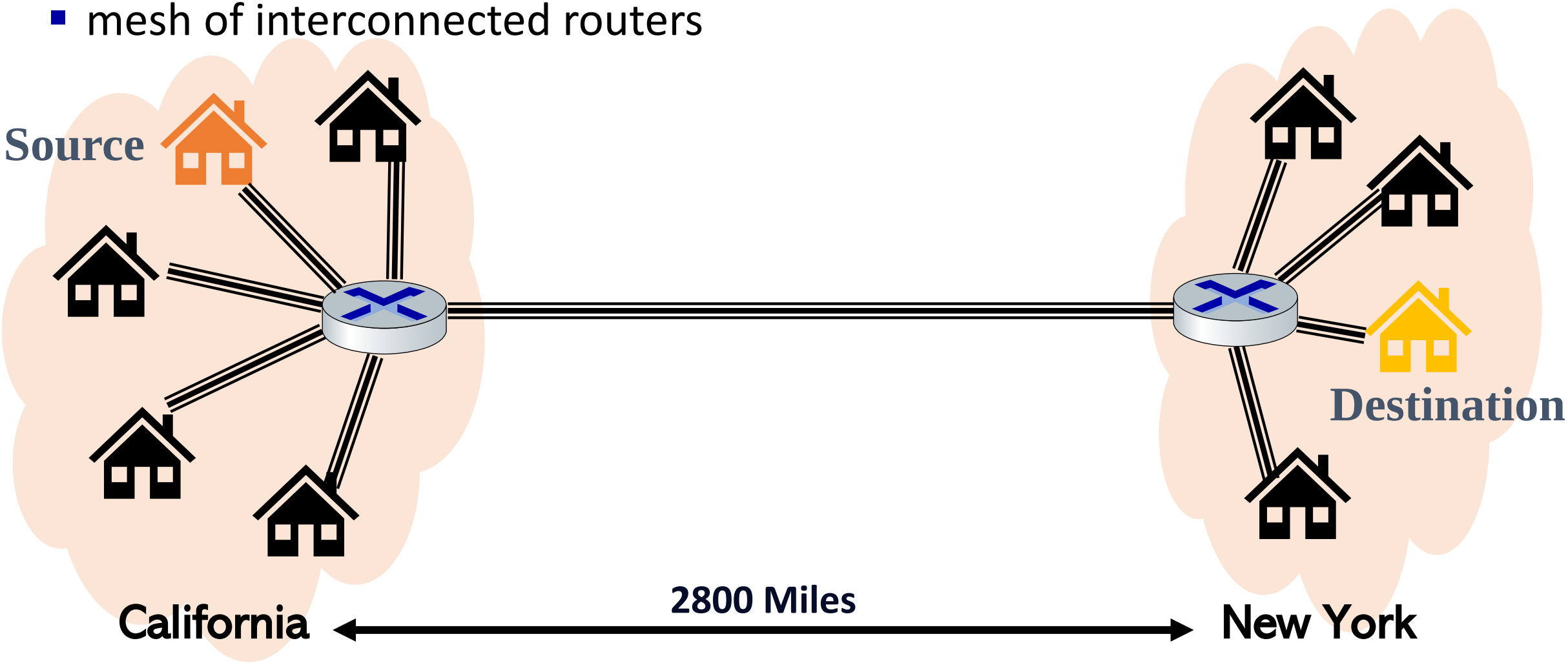
The network core

- mesh of interconnected routers



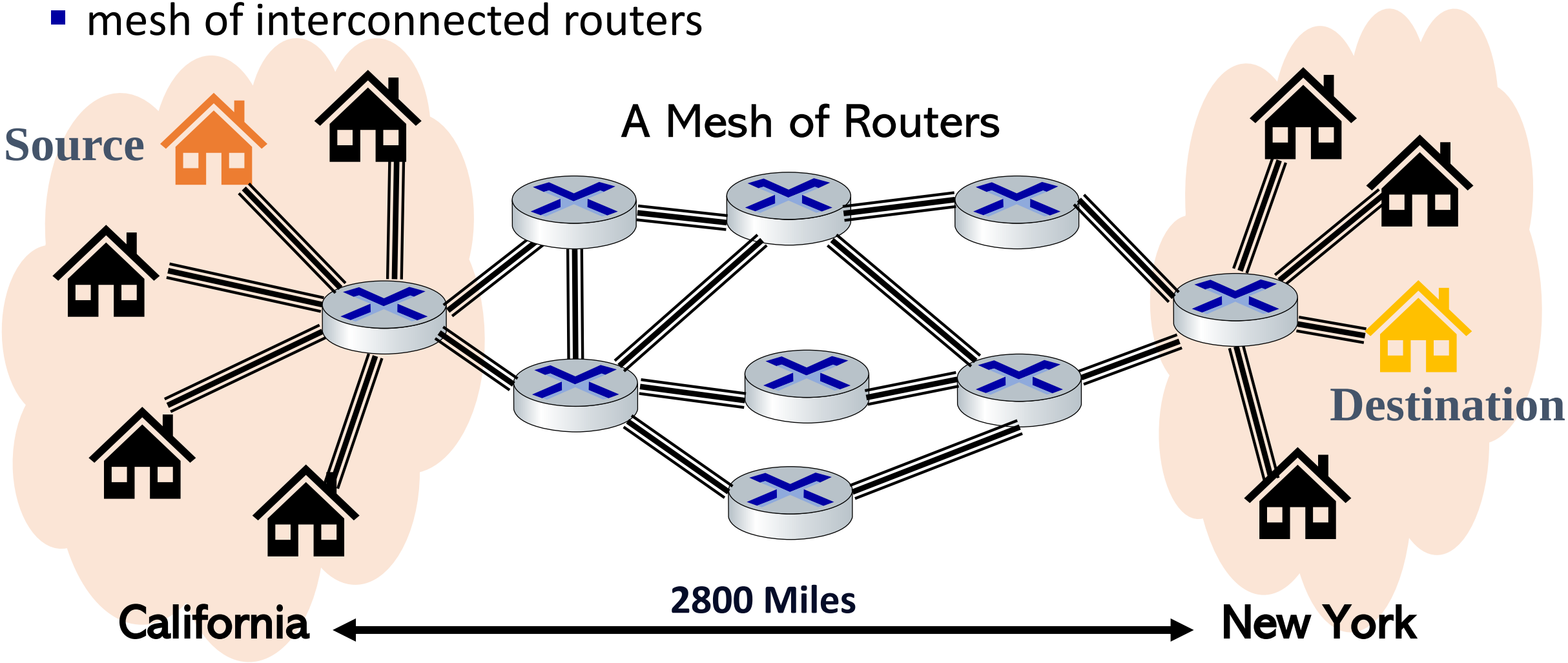
The network core

- mesh of interconnected routers



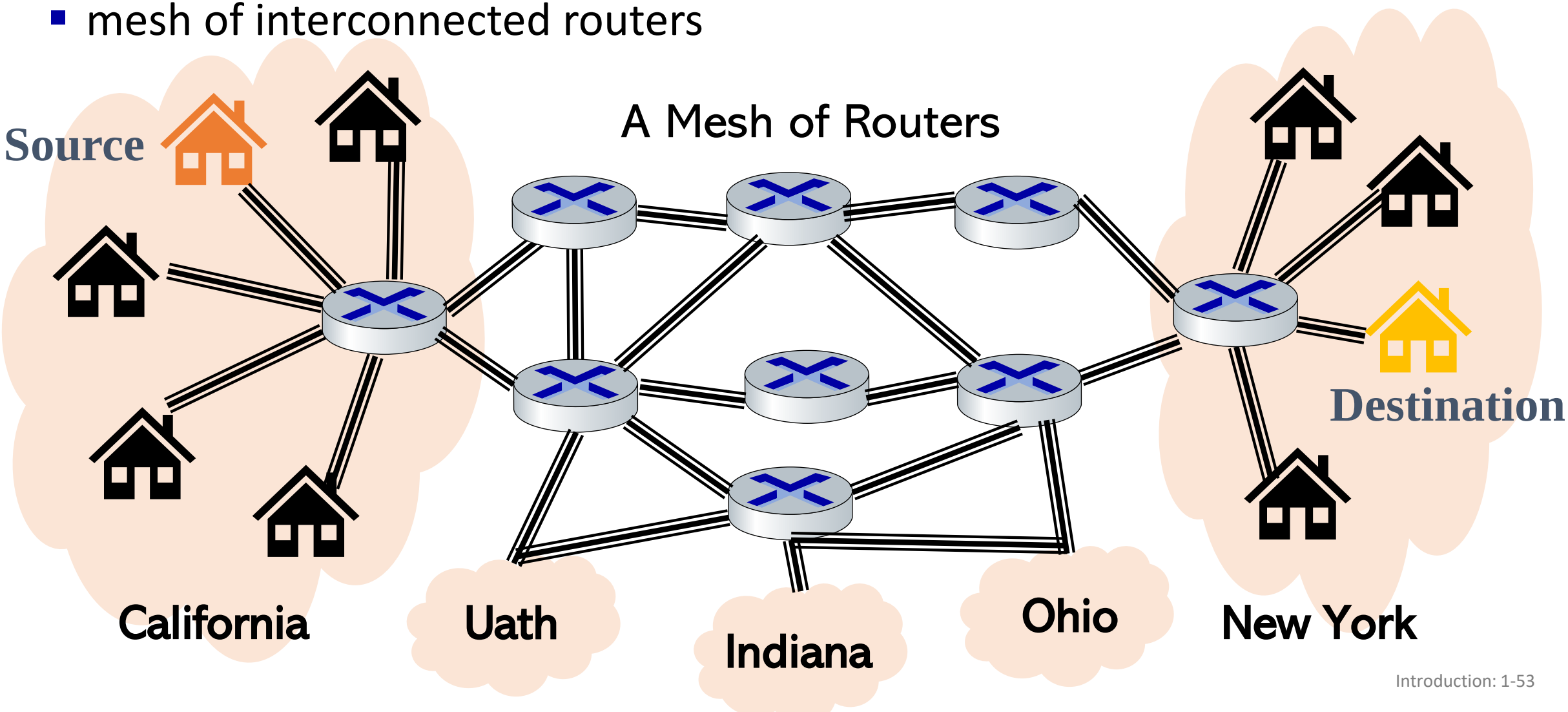
The network core

- mesh of interconnected routers



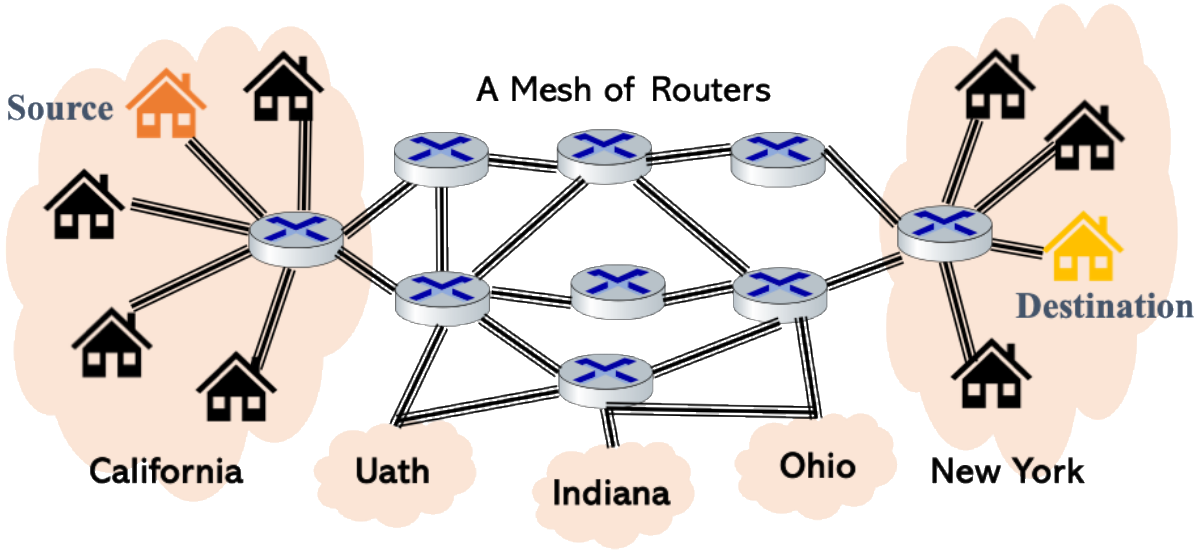
The network core

- mesh of interconnected routers



The network core

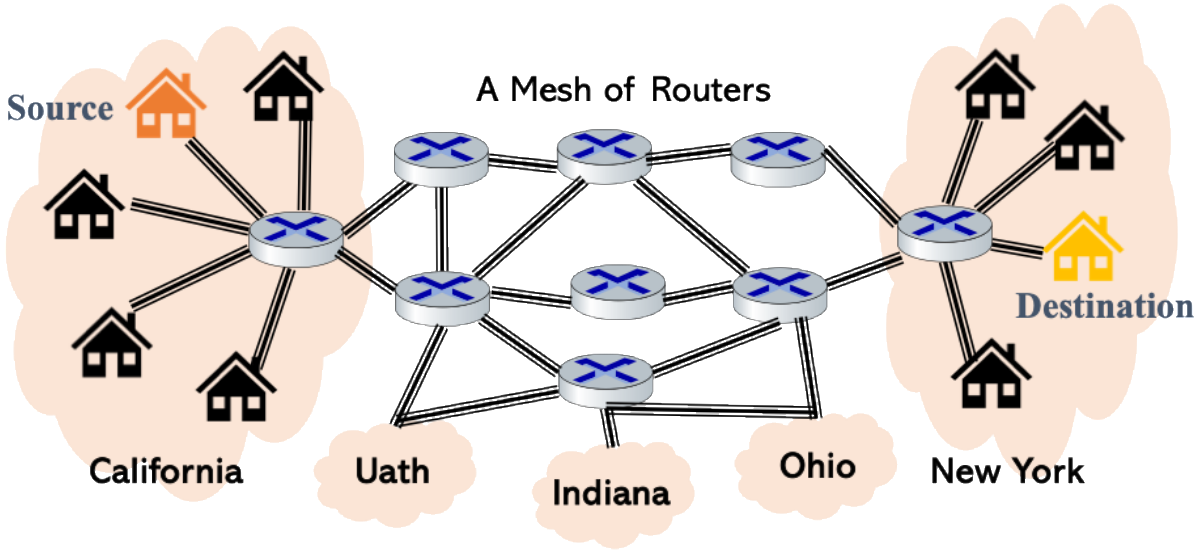
- mesh of interconnected routers



Aren't they very similar to each other?

The network core – Routing

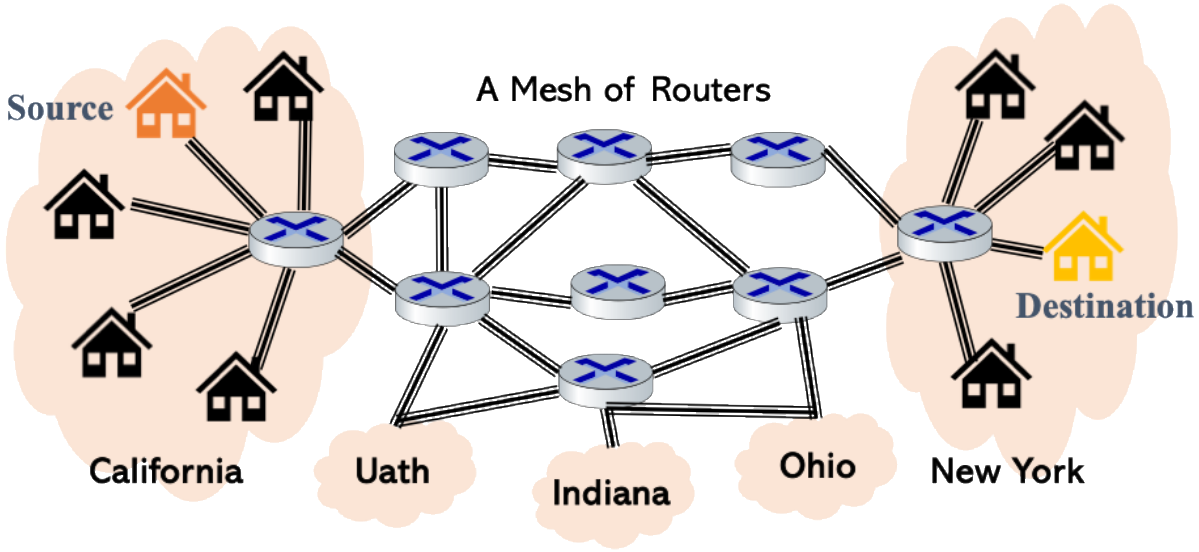
- Routing: Finding the Correct/Optimal path from source to destination



What's a correct/optimal path?

The network core – Routing

- Routing: Finding the Correct/Optimal path from source to destination



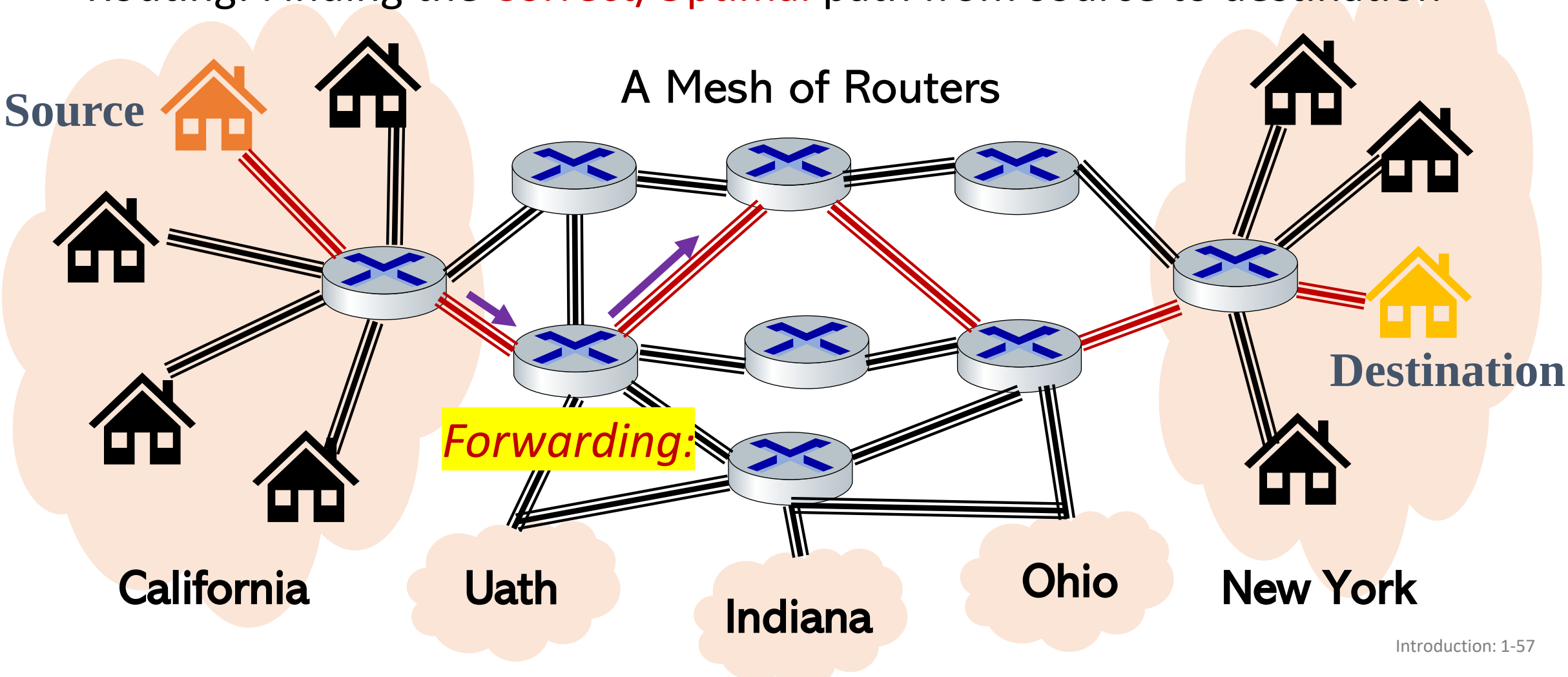
Routing Algorithm

- ❖Shortest distance?

- ❖Shortest distance?
- ❖Cheapest without tolls?
- ❖Best views?

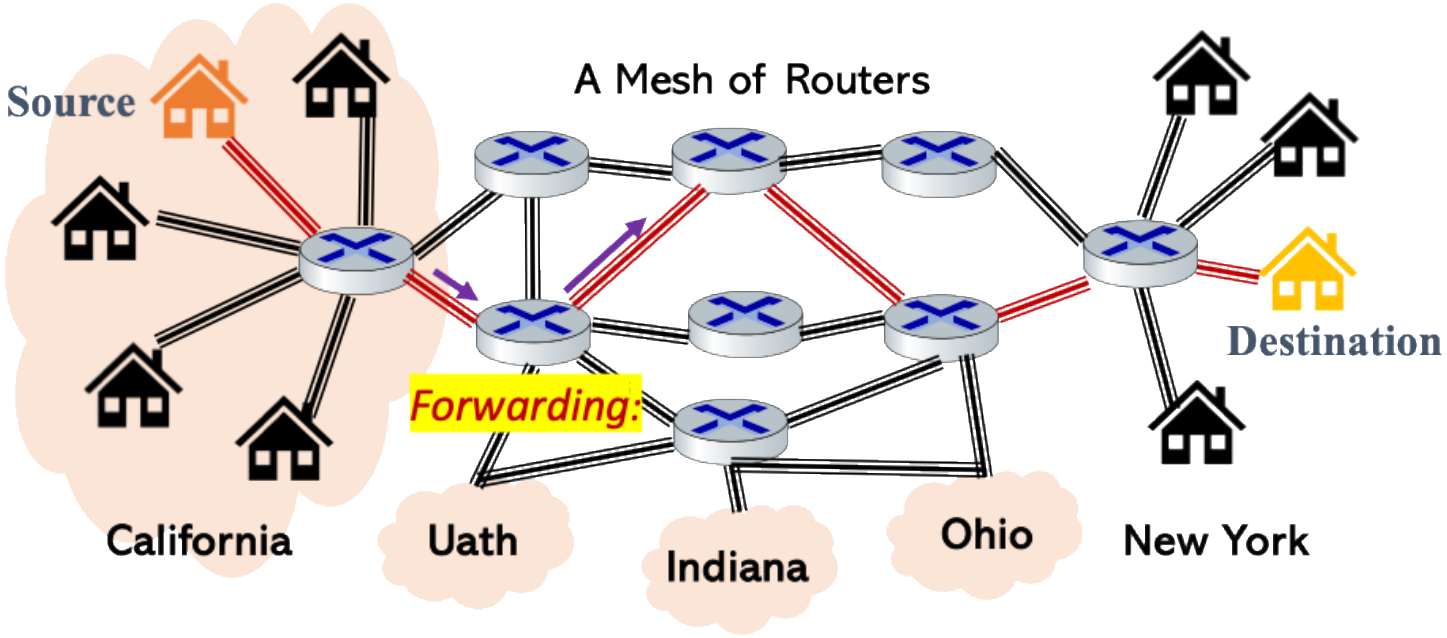
The network core – Routing

- Routing: Finding the Correct/Optimal path from source to destination



The network core – Routing

- Routing: Finding the Correct/Optimal path from source to destination



Forwarding:

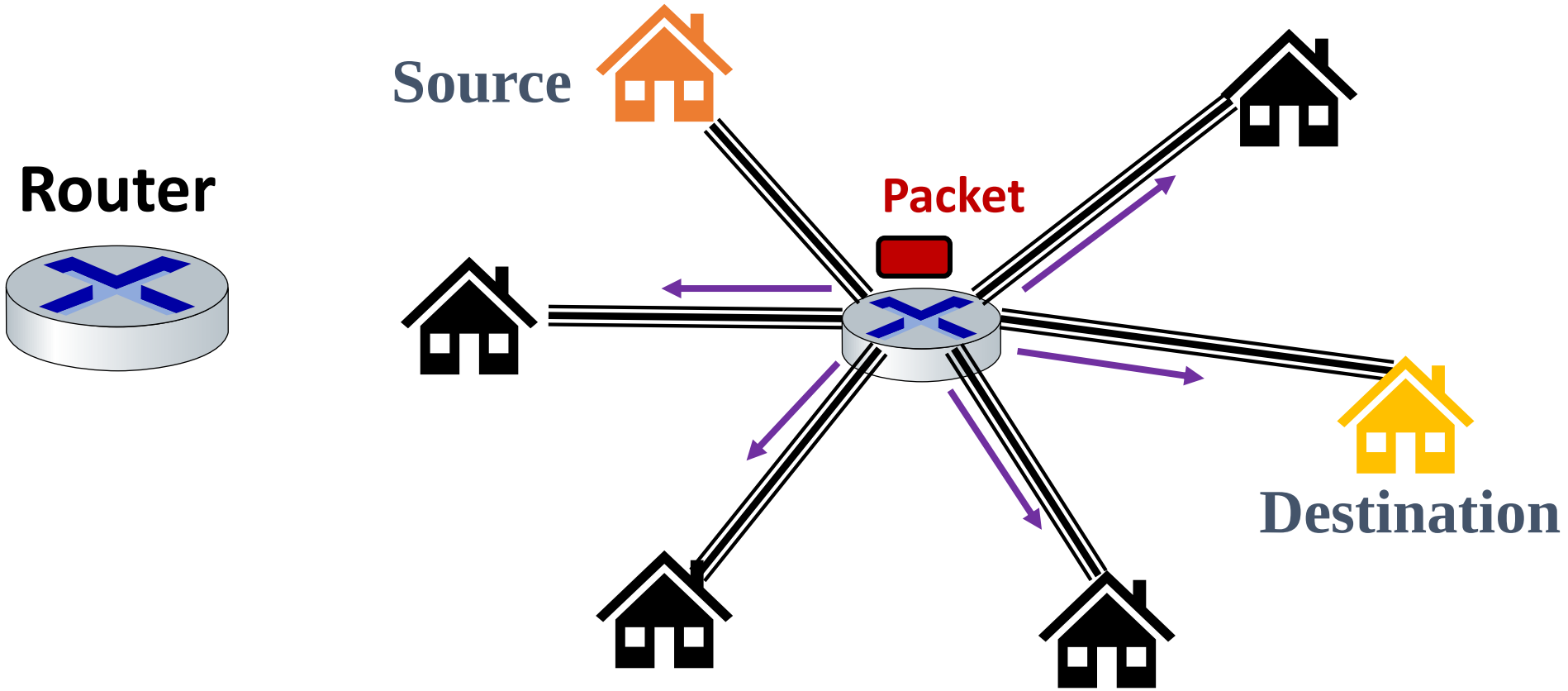
- *local* action: move arriving packets from router's input link to appropriate router output link

Routing:

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

Packet Switching VS Circuit Switching

- Forward is also called switching

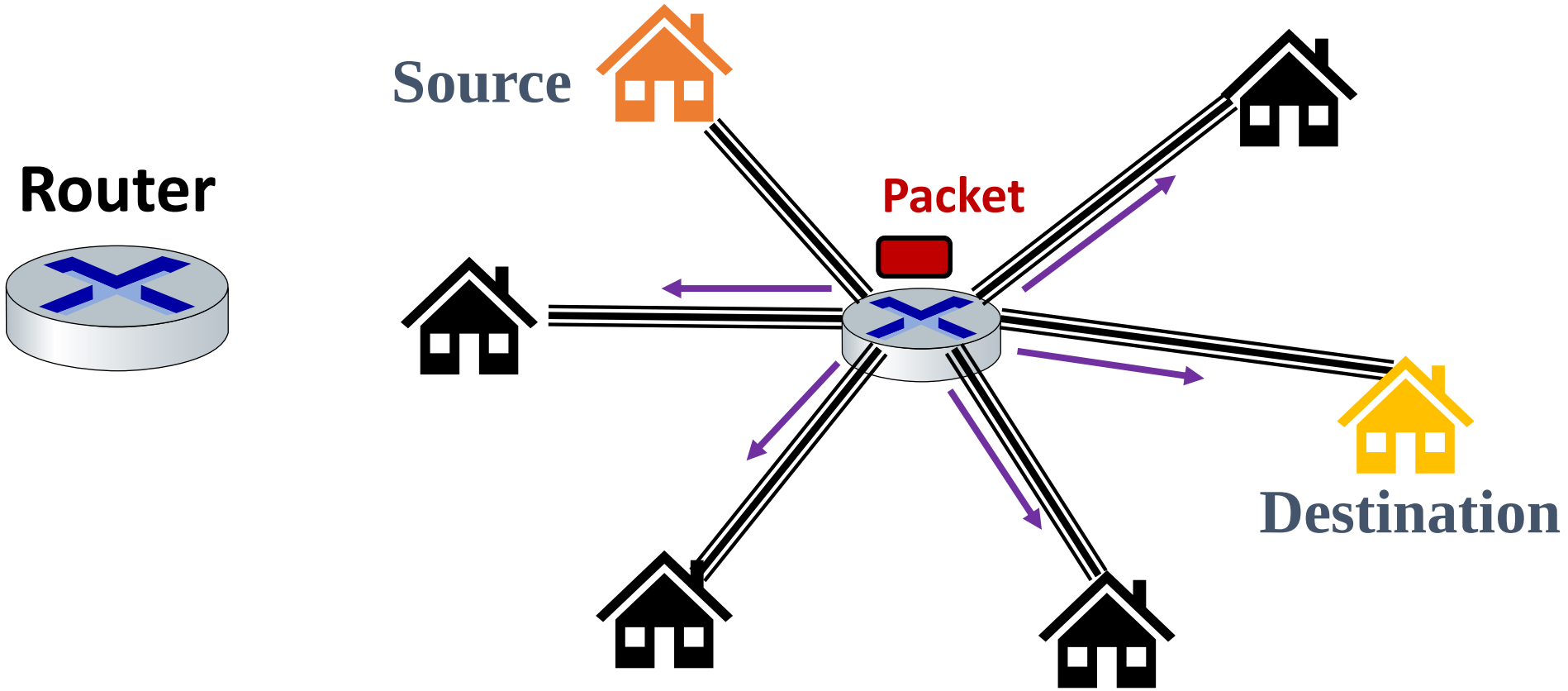


Forwarding:

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

Packet Switching

- Forward is also called switching

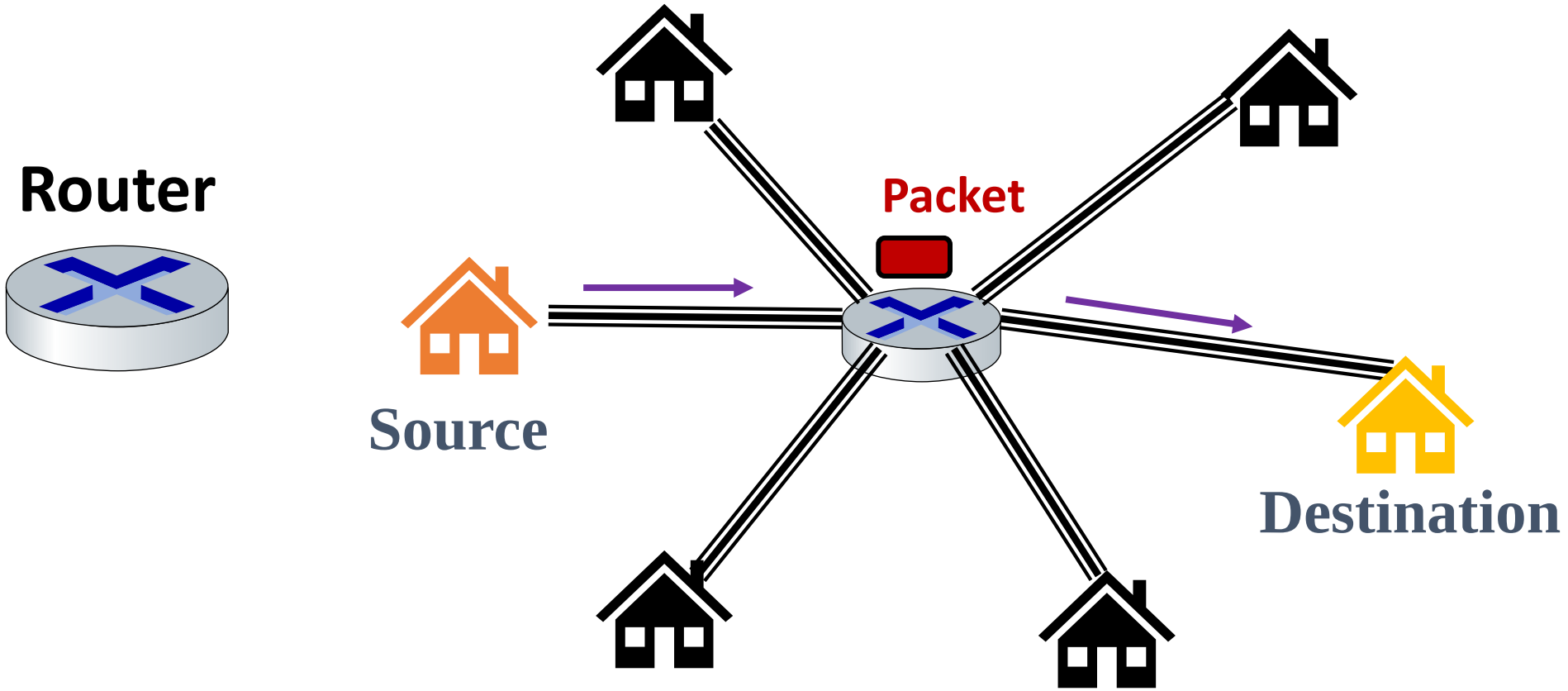


Forwarding:

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

Packet Switching

- Forward is also called switching

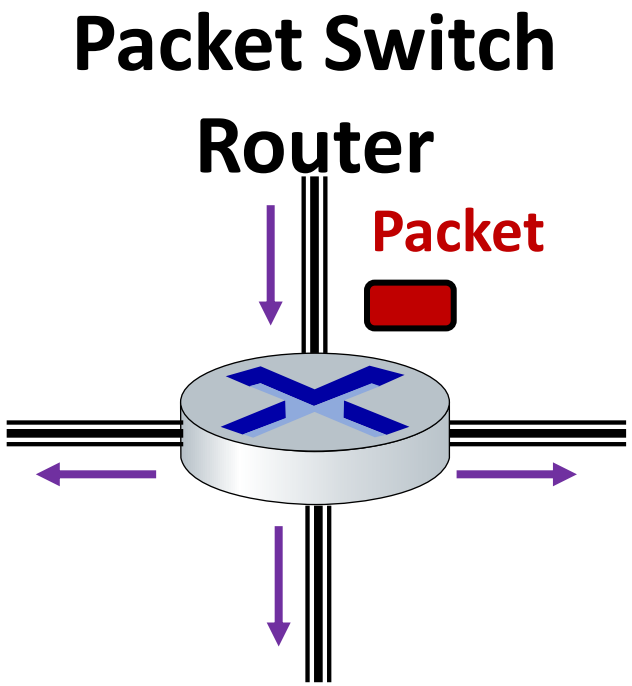


Forwarding:

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

Packet-switching: store-and-forward

- Forward is also called switching
  - *store and forward: entire* packet must arrive at router before it can be transmitted on next link

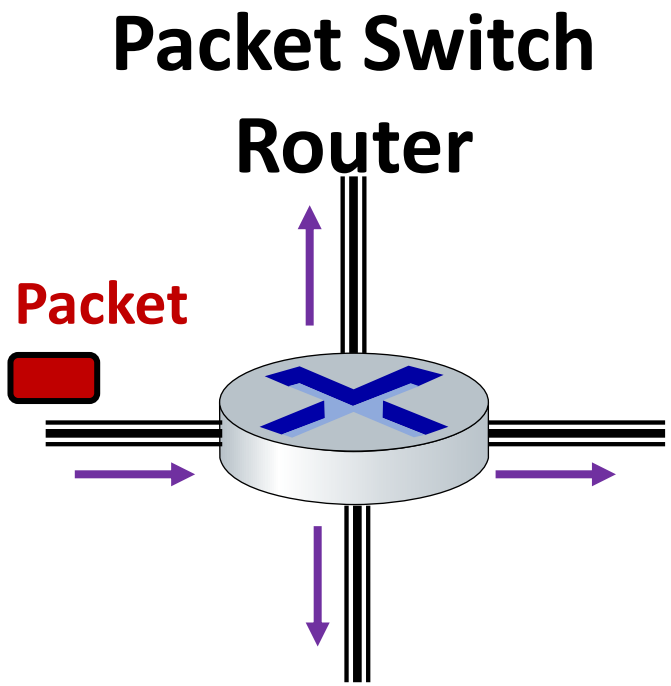


Forwarding:

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

Packet-switching: store-and-forward

- Forward is also called switching
  - *store and forward: entire* packet must arrive at router before it can be transmitted on next link

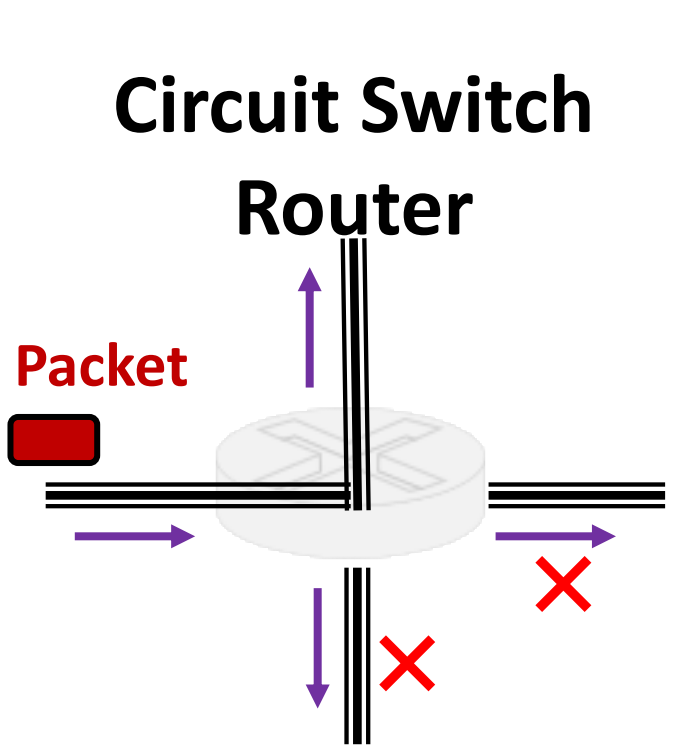


Forwarding:

- aka "switching"
- *local* action: move arriving packets from router's input link to appropriate router output link

Circuit switching

- Forward is also called switching



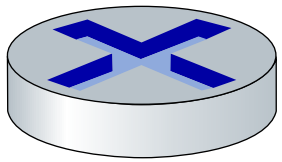
Forwarding:

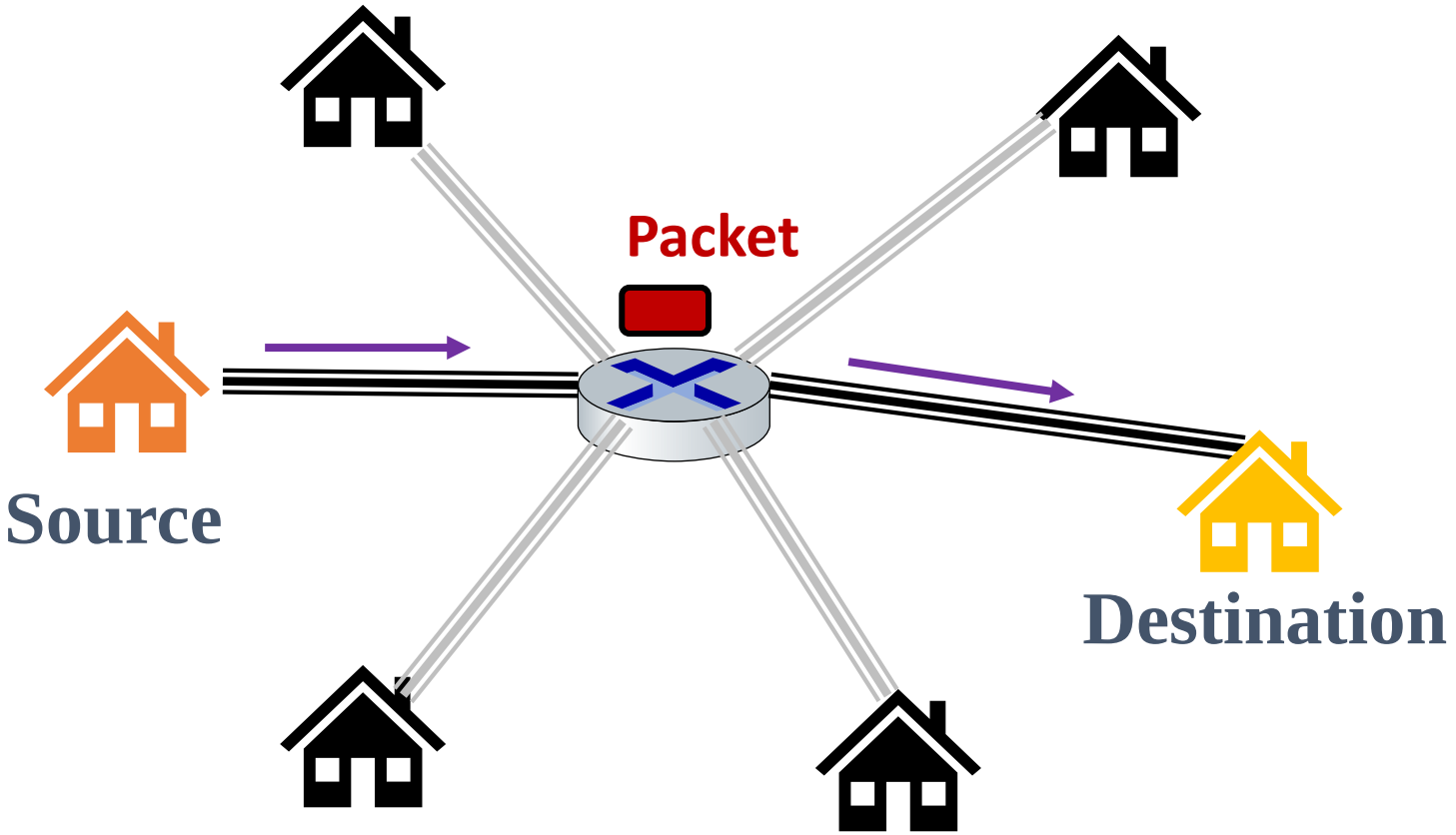
- aka “switching”
- *local* action: move arriving packets from router’s input link to appropriate router output link

Circuit switching

- Forward is also called switching

Circuit Switch Router





Forwarding:

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

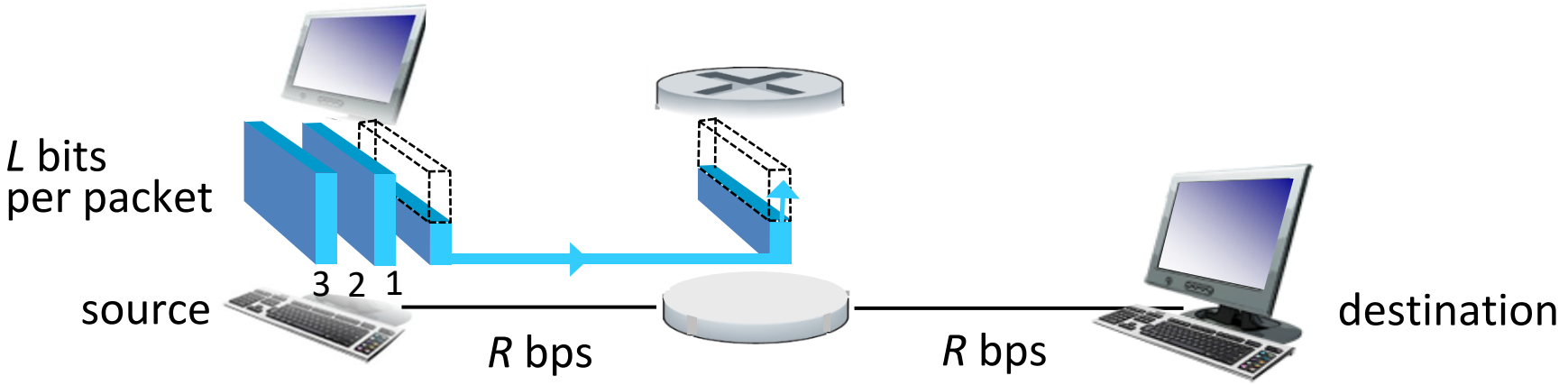
Internet Core: Packet Switching

each end-end data stream divided into *packets*

- users A through C packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

- each packet has a "header" (containing e.g., destination address) in addition to "payload" (data)
- Store and Forward (requires buffer and introduces delay)

Packet-switching: store-and-forward

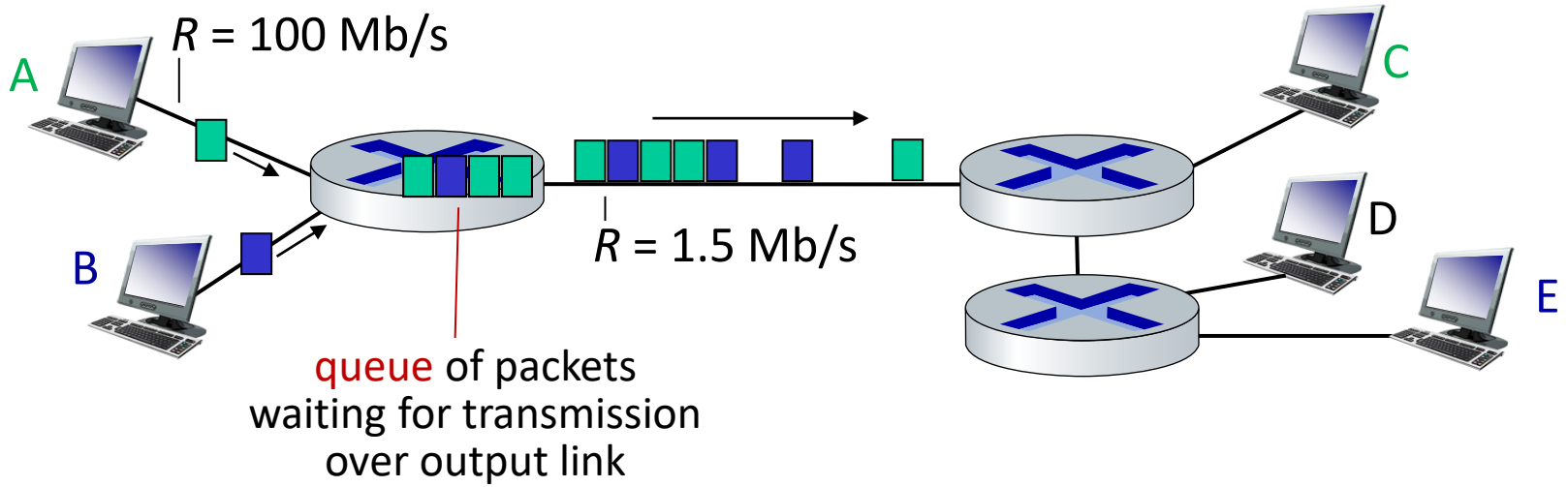


- packet transmission delay: takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

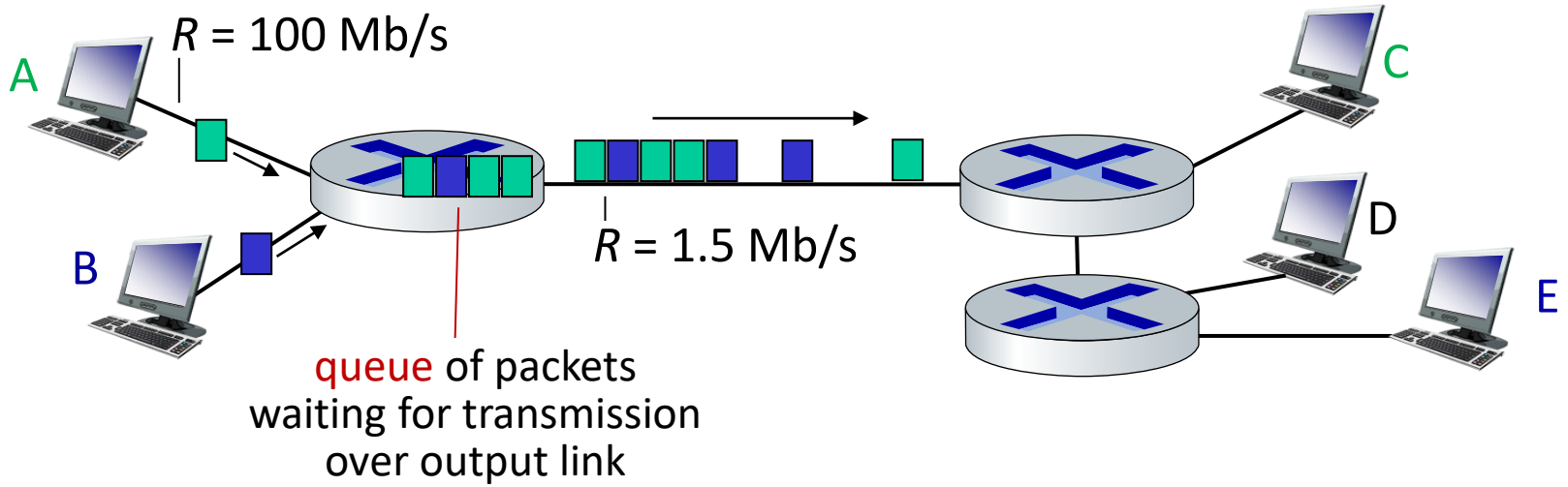
- L = 10 Kbits
- R = 100 Mbps
- one-hop transmission delay = 0.1 msec

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

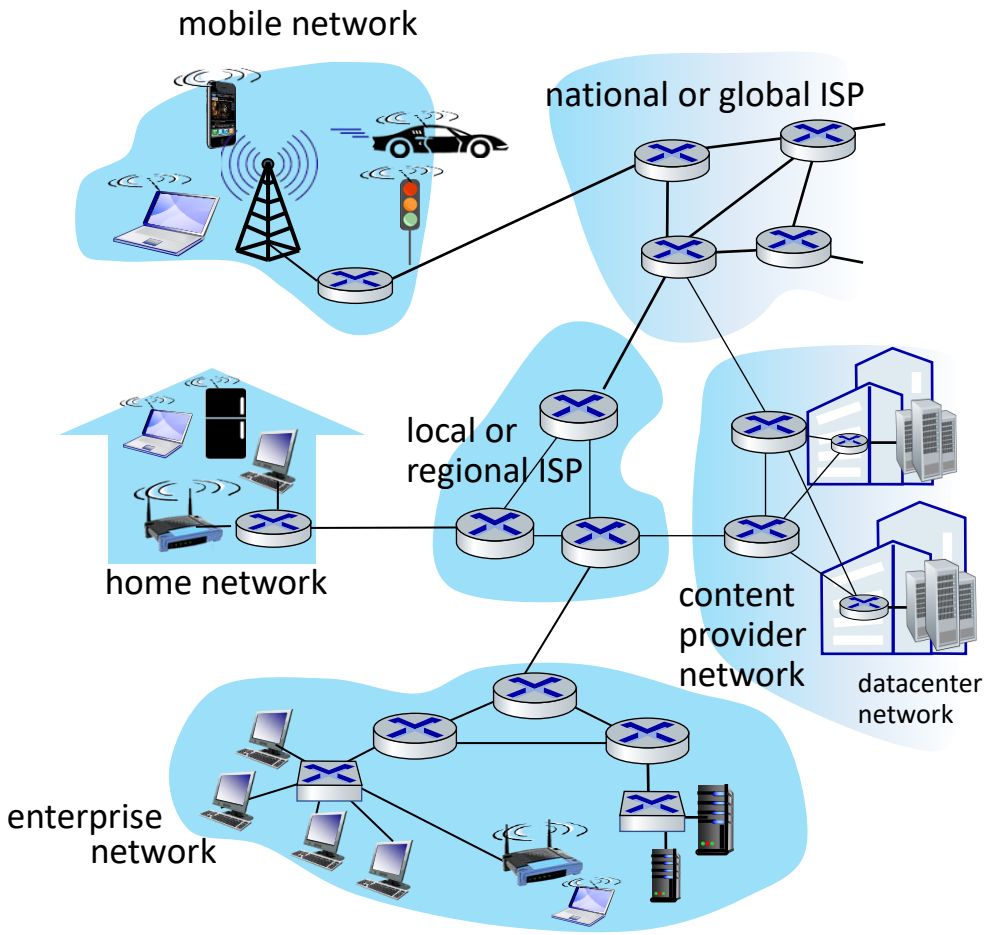
Packet switching versus circuit switching

Is packet switching a “slam dunk winner”?

- great for “bursty” data – sometimes has data to send, but at other times not
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior with packet-switching?
  - “It’s complicated.” We’ll study various techniques that try to make packet switching as “circuit-like” as possible.

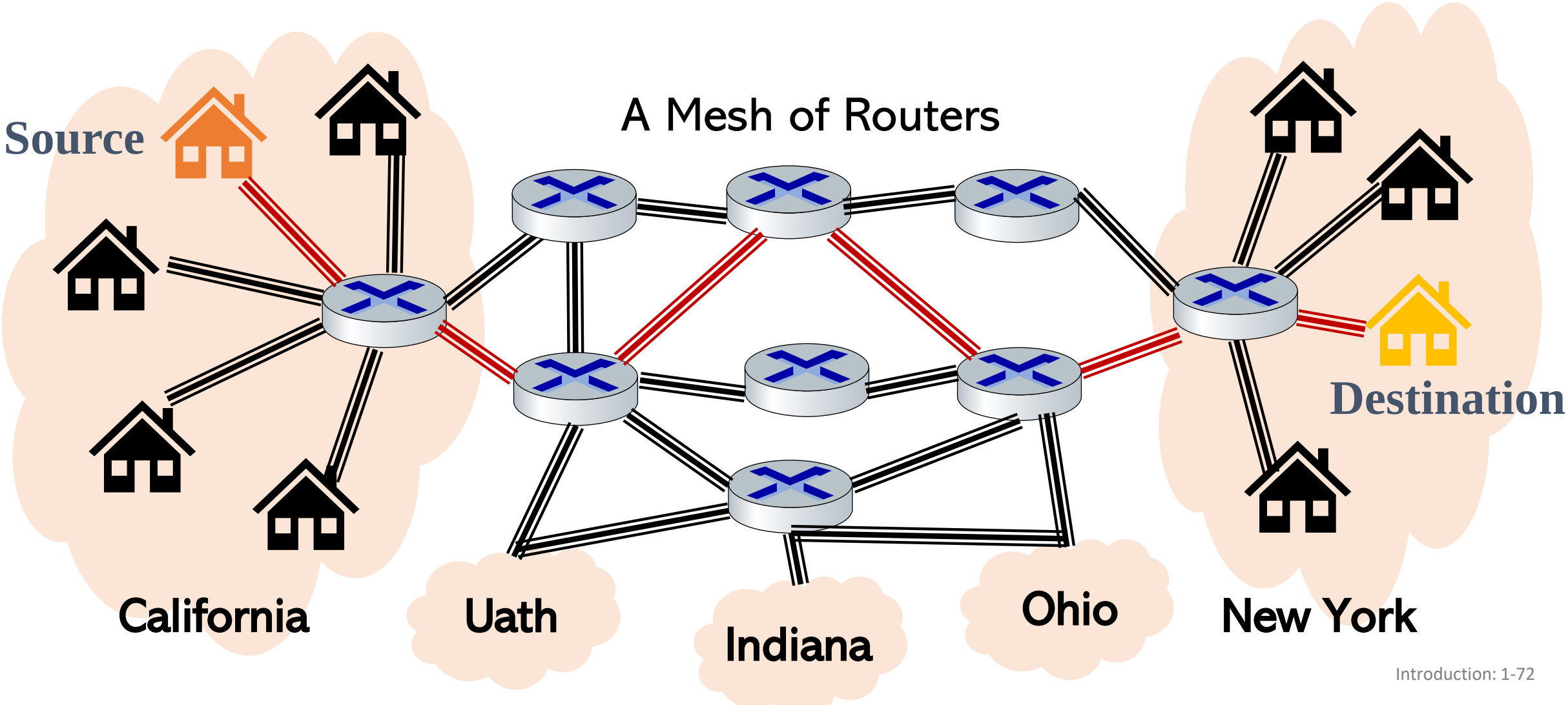
Internet structure: a "network of networks"

- 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

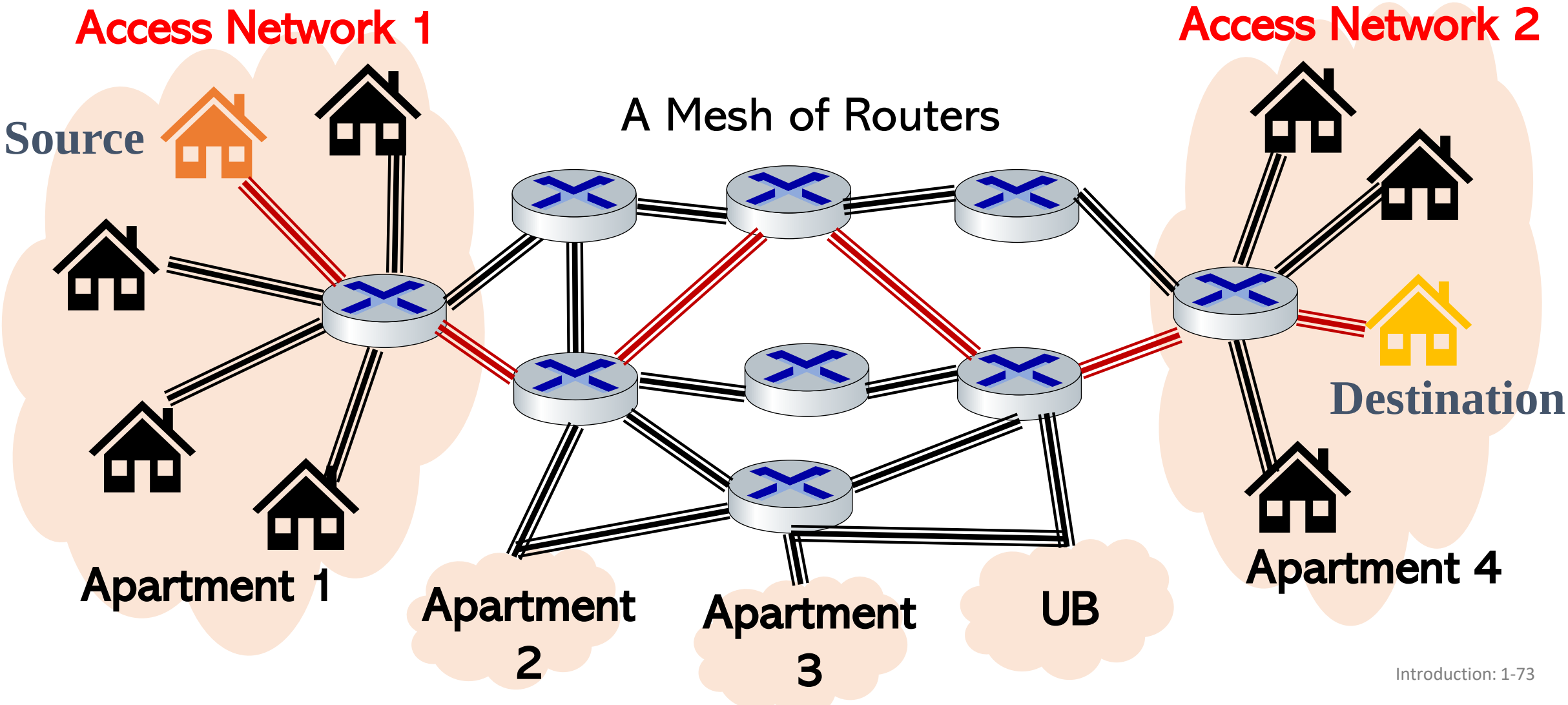


Let's take a stepwise approach to describe current Internet structure

Internet structure: a "network of networks"

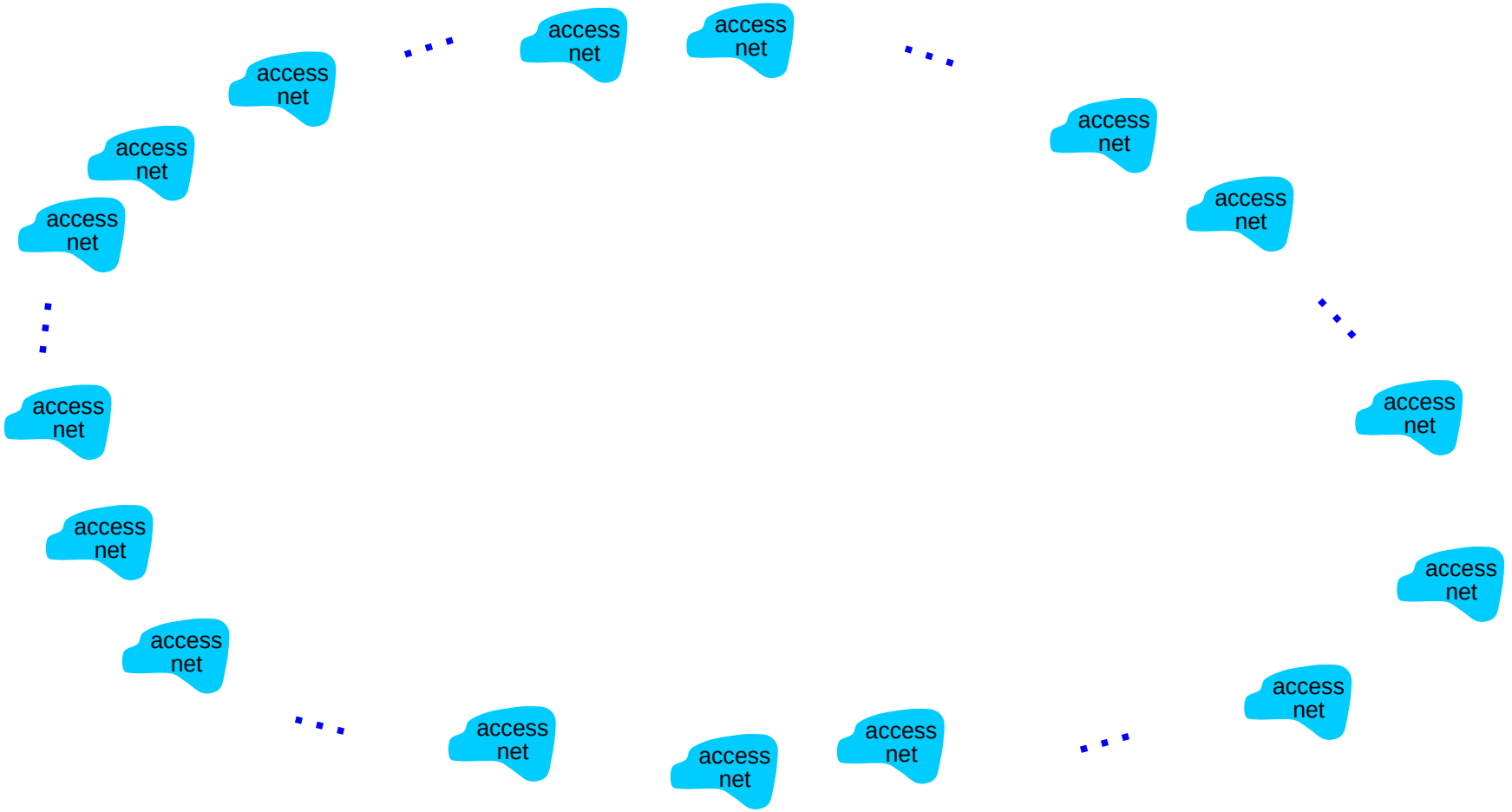


Internet structure: a "network of networks"



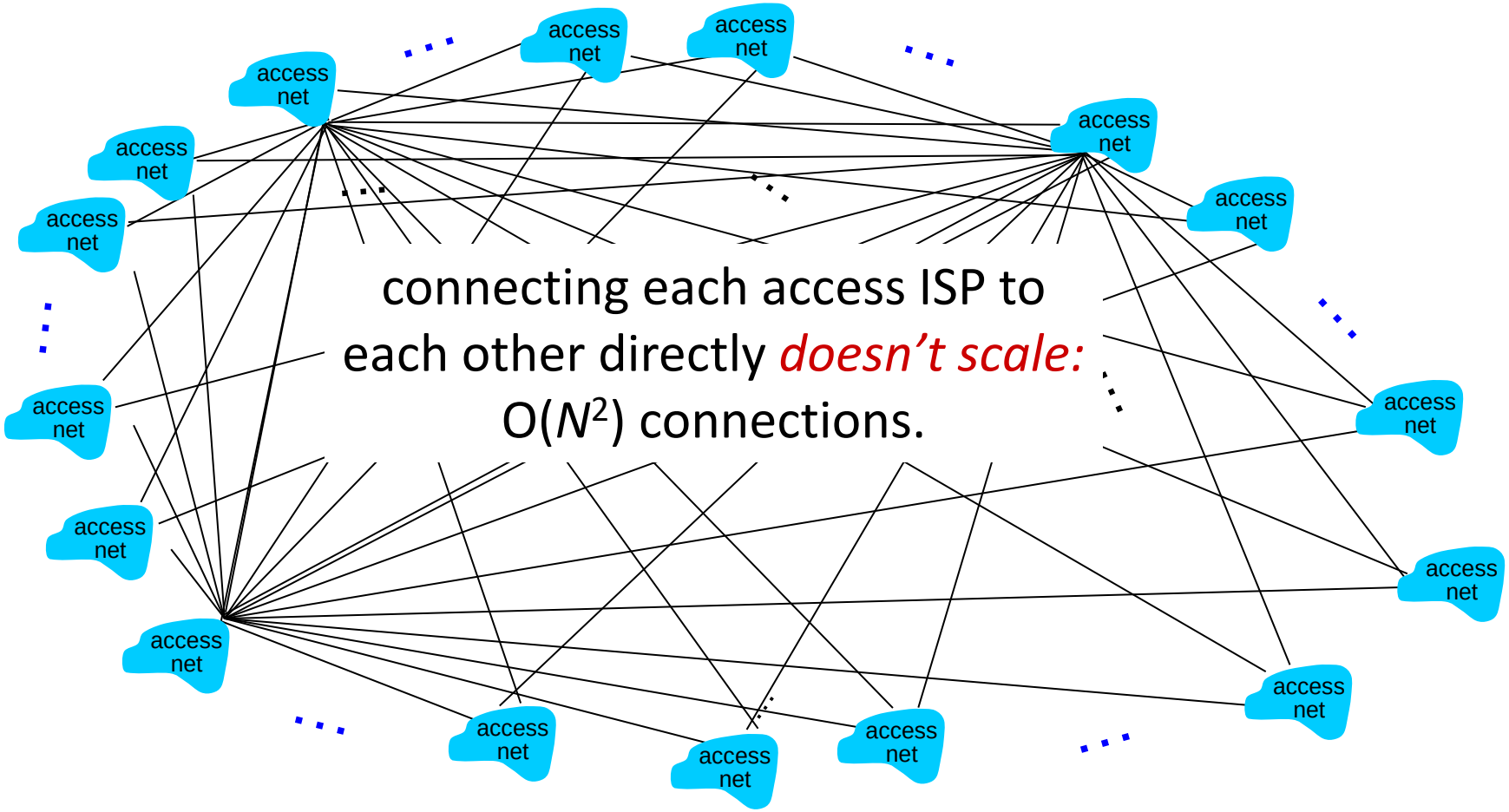
Internet structure: a "network of networks"

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



Internet structure: a "network of networks"

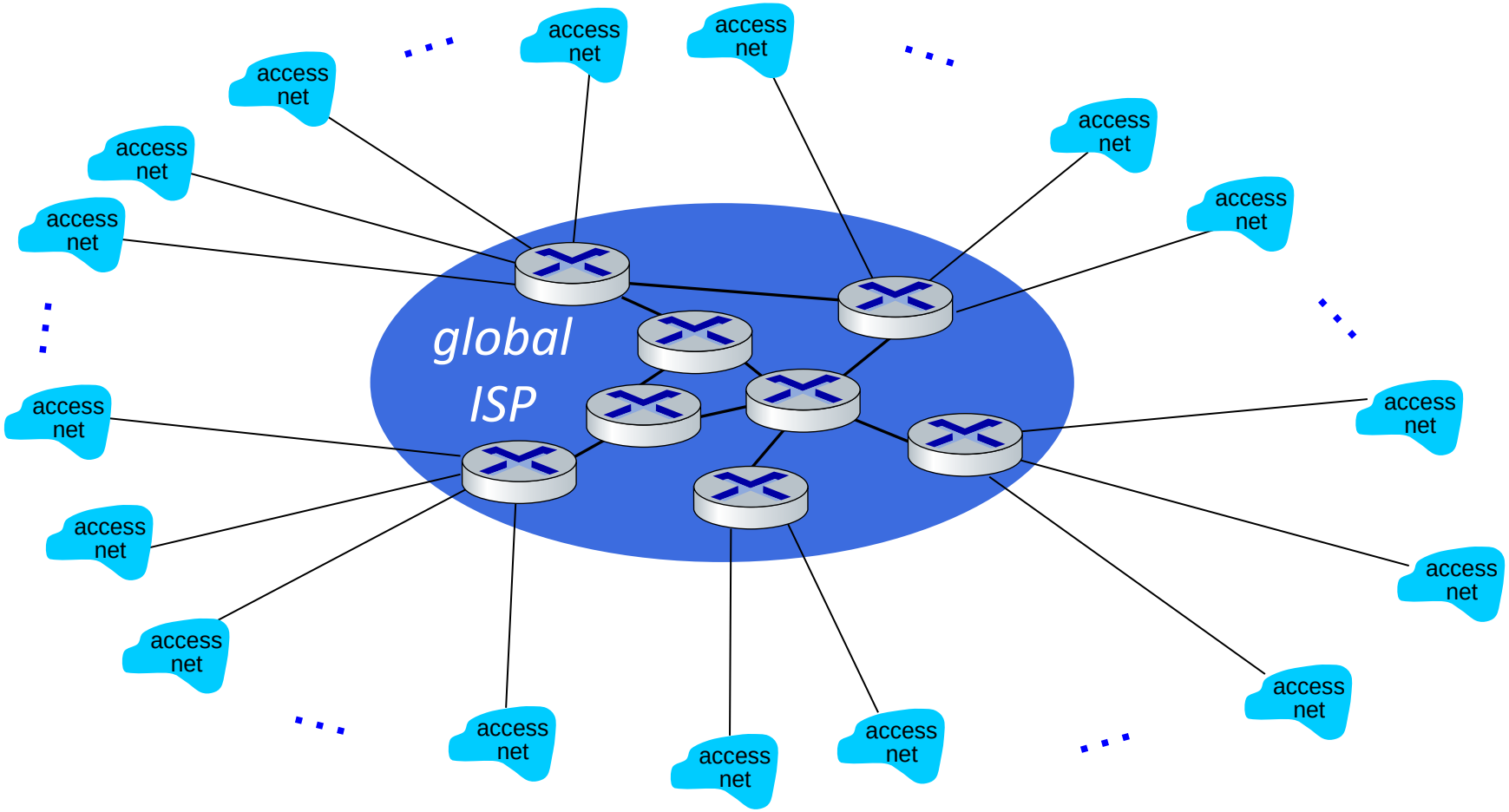
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

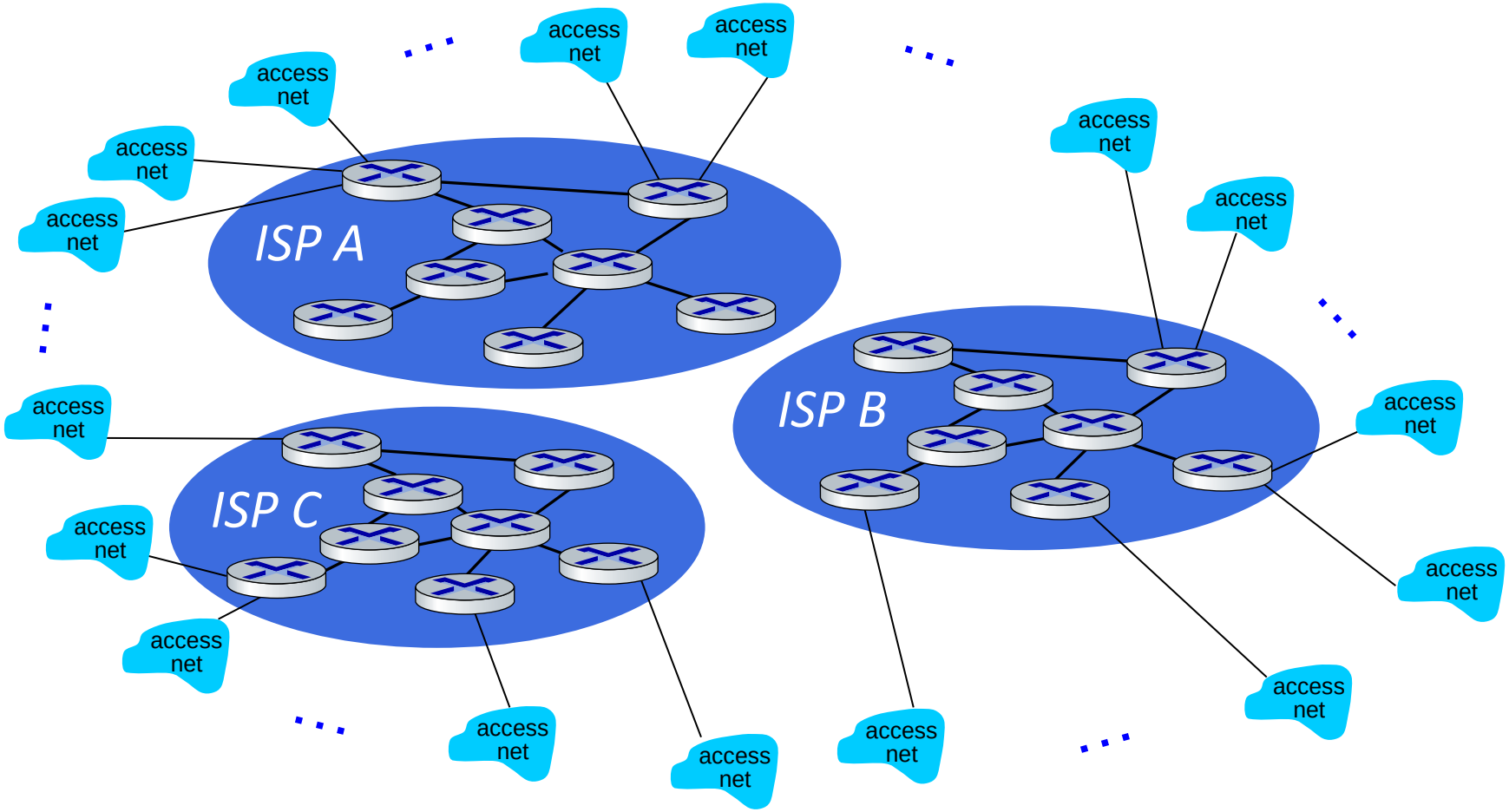
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



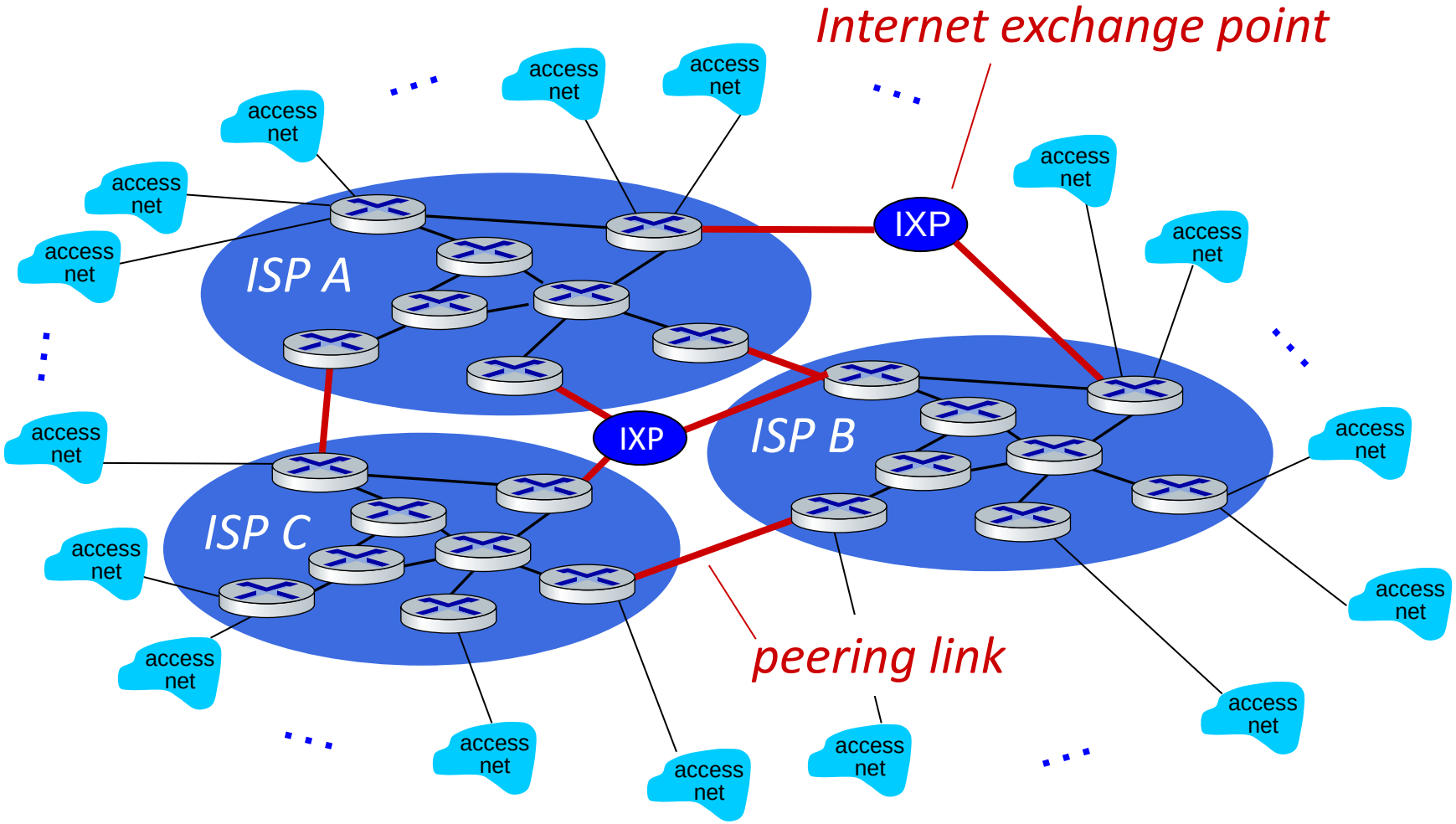
Internet structure: a "network of networks"

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



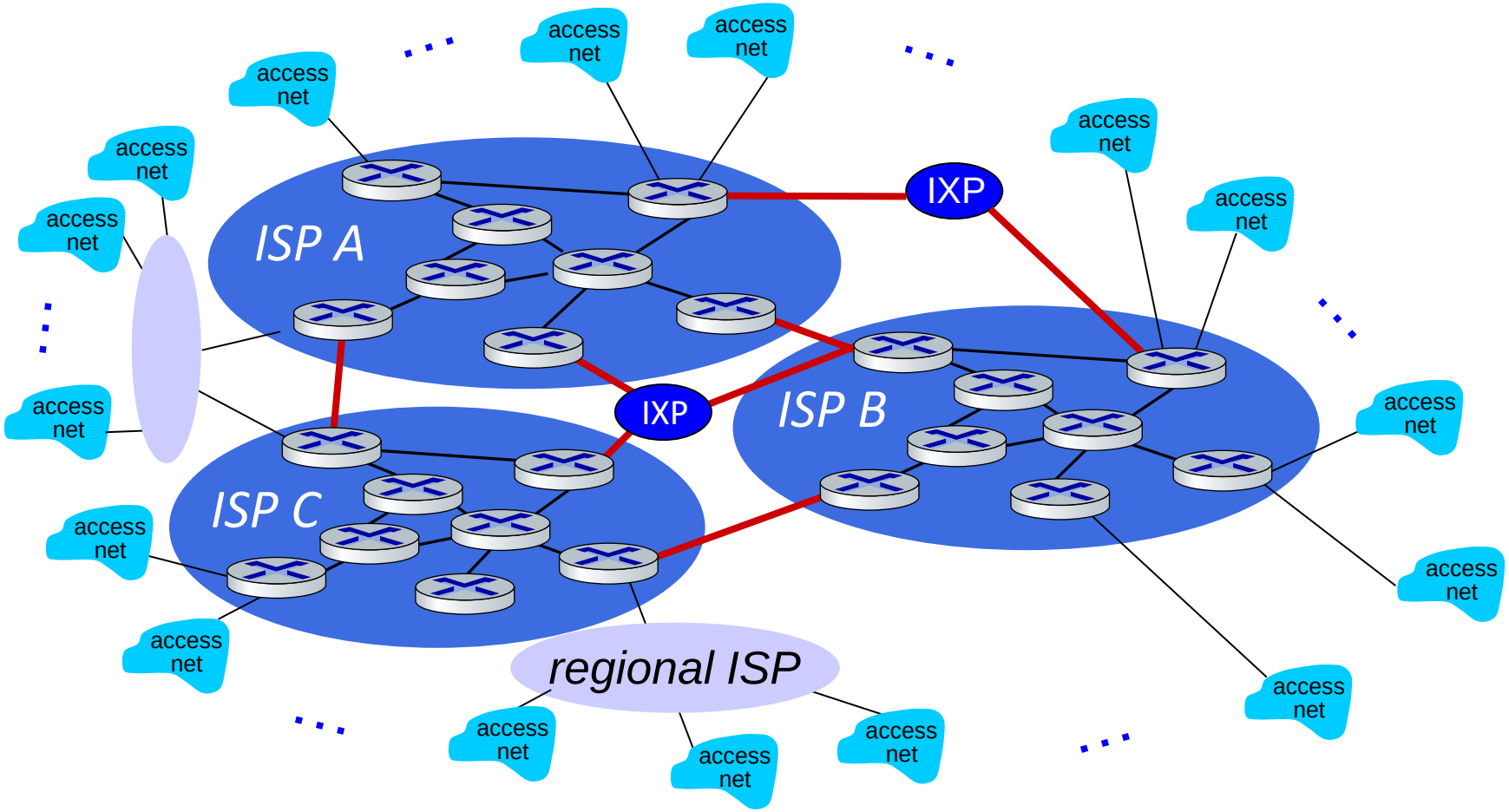
Internet structure: a "network of networks"

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



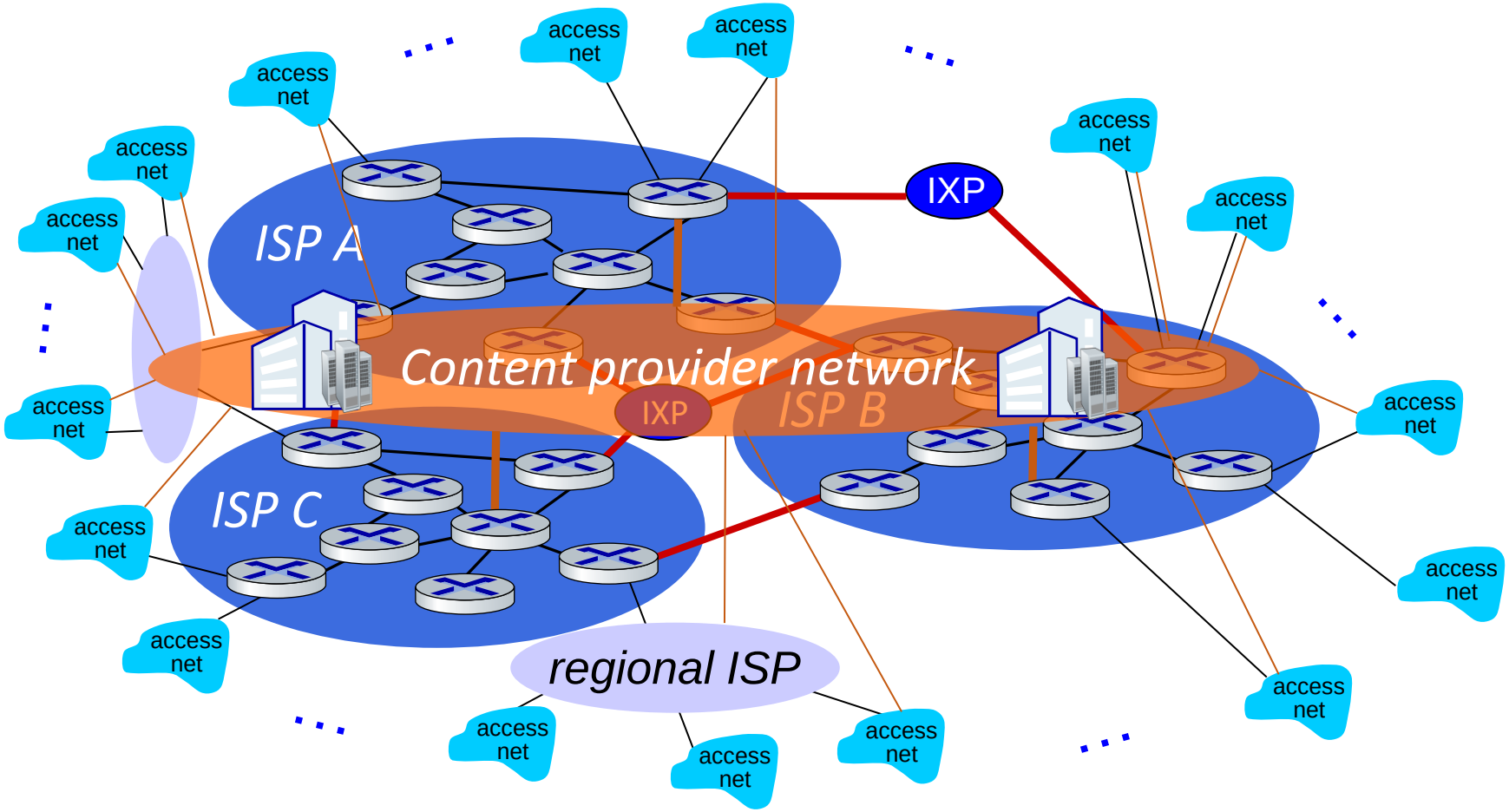
Internet structure: a "network of networks"

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

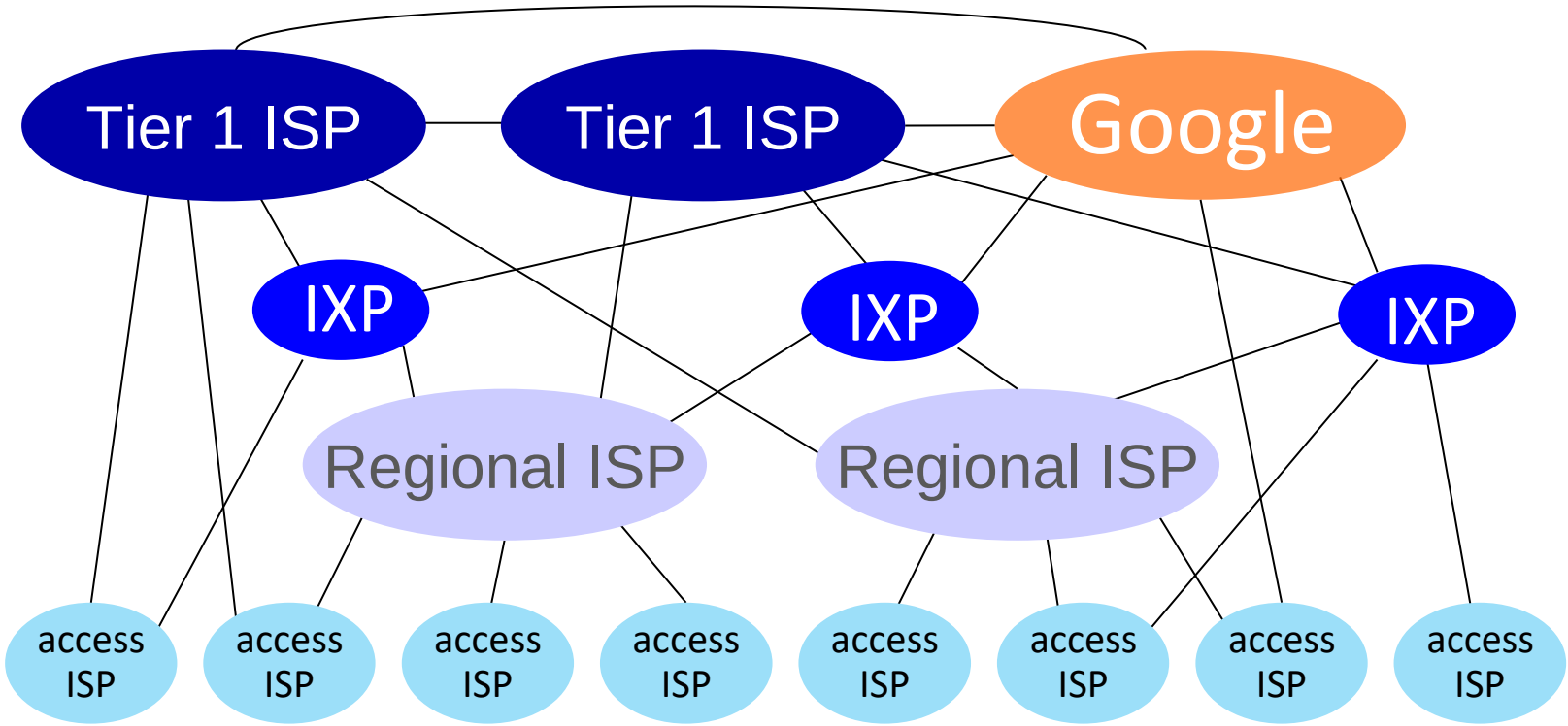


Internet structure: a "network of networks"

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



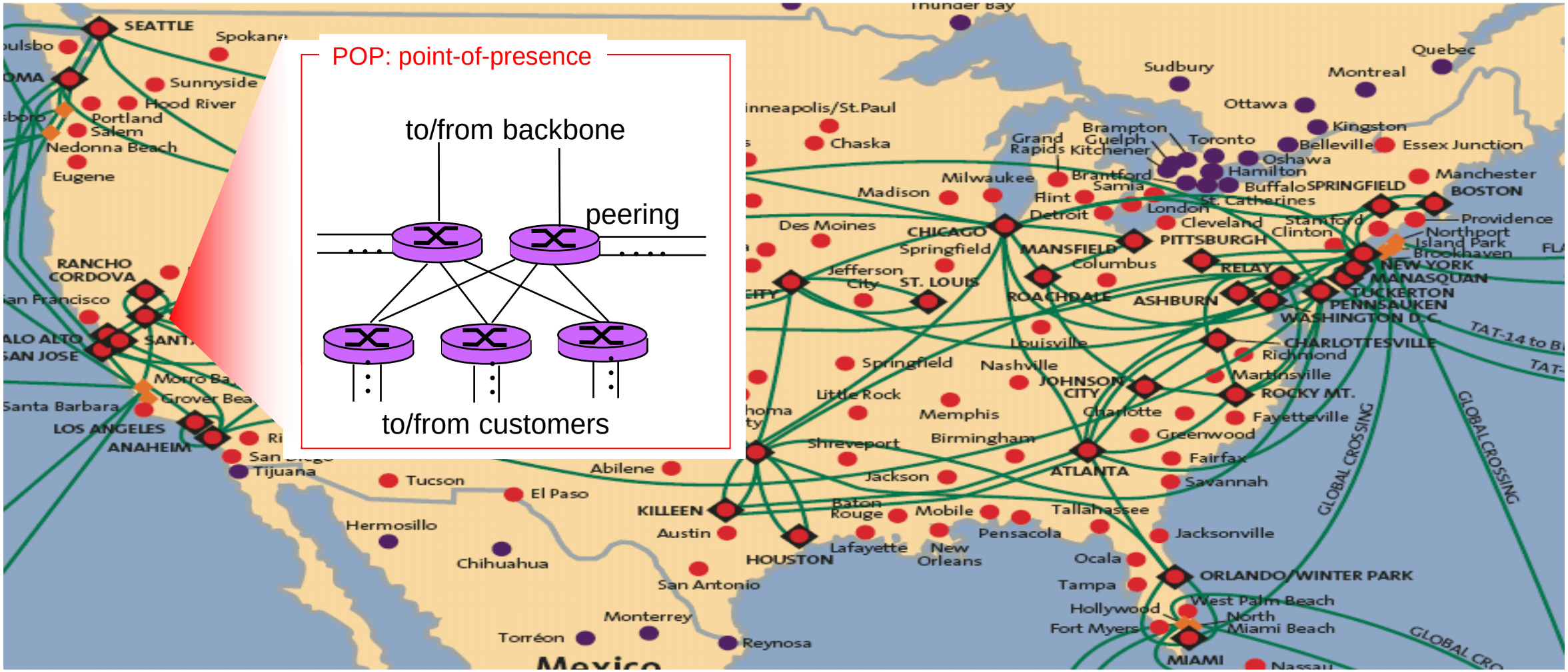
Internet structure: a "network of networks"



At "center": small # of large but well-connected networks

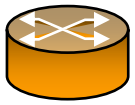
- "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- content provider networks (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Tier-1 ISP: e.g., Sprint/T-Mobile



POPs from different Tier-1 ISP connect to each other at IXPs – residing at a building like this in London

Internet Core Routers (including those at POPs/IXP)

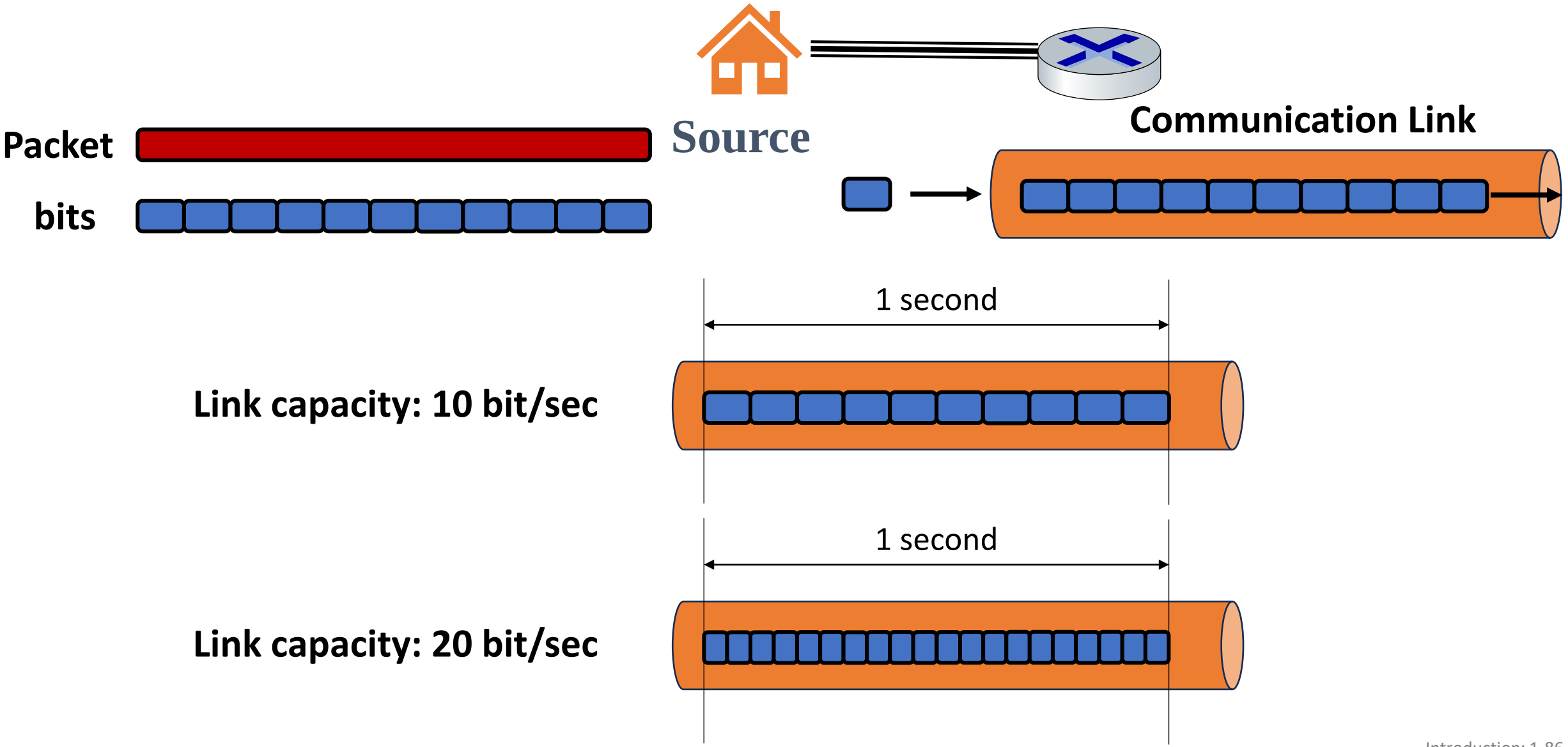


Router on
“paper”

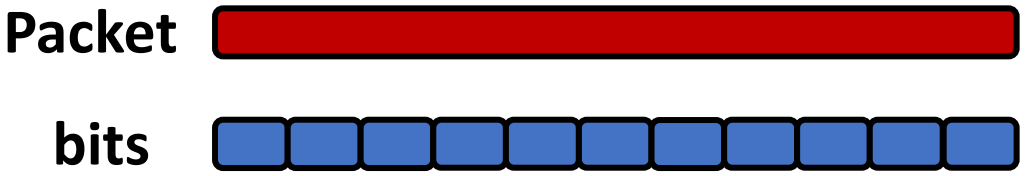
Chapter 1: roadmap

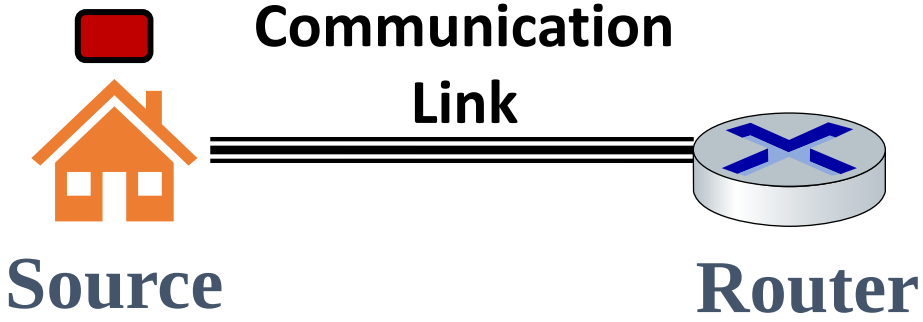
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: internet structure, routing and forwarding
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

How to send a packet via network

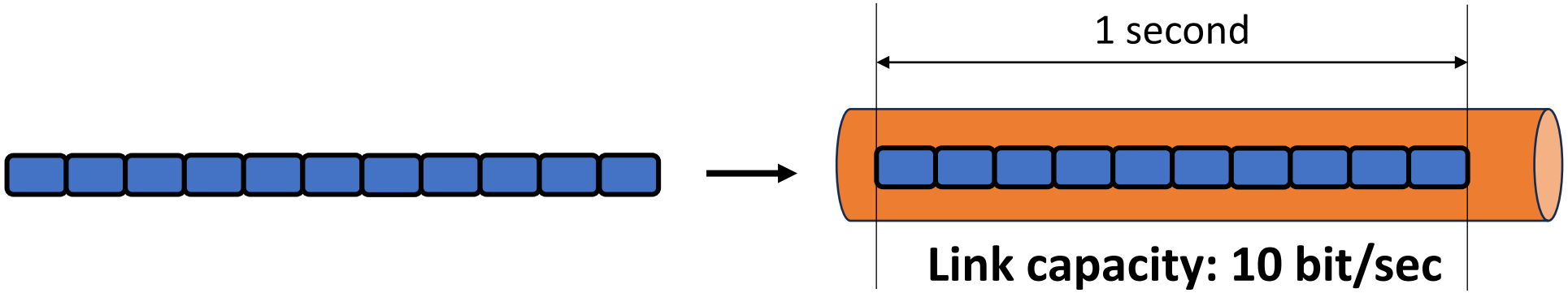


How to send a packet via network

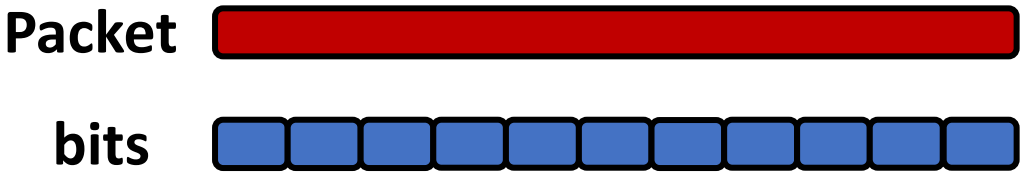


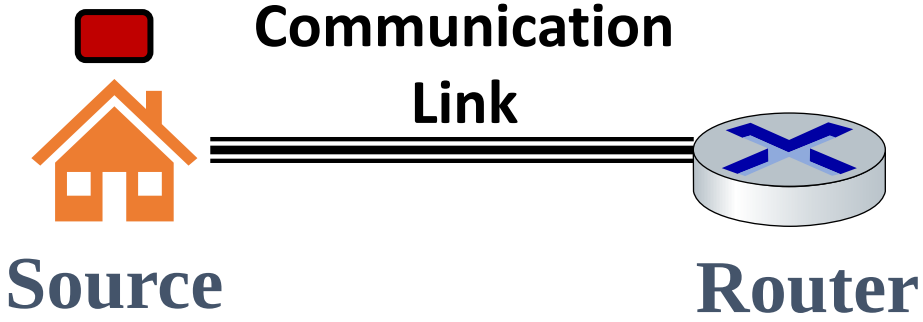


Step 1: Transmit the packets into the link



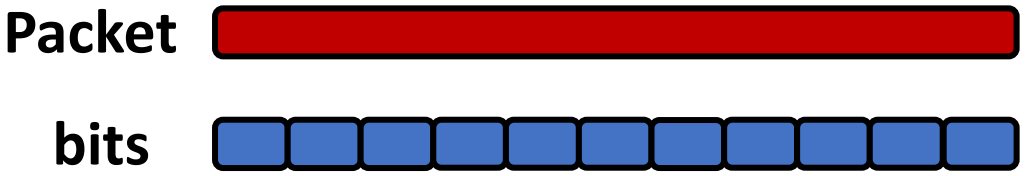
How to send a packet via network

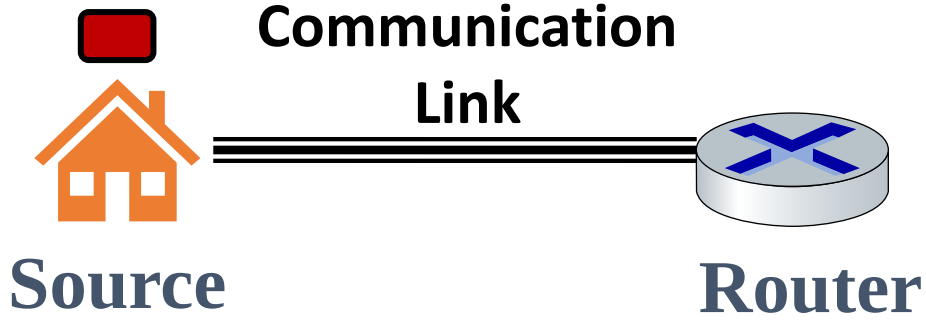




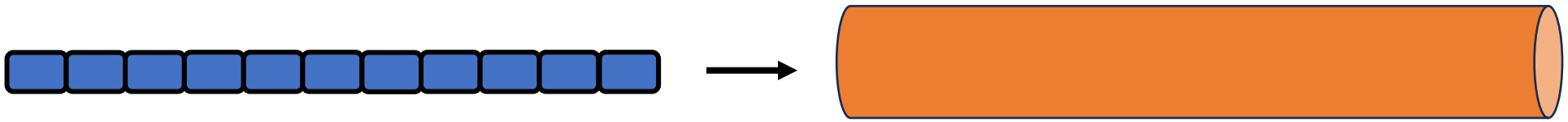
Step 1: Transmit the packets into the link

How to send a packet via network



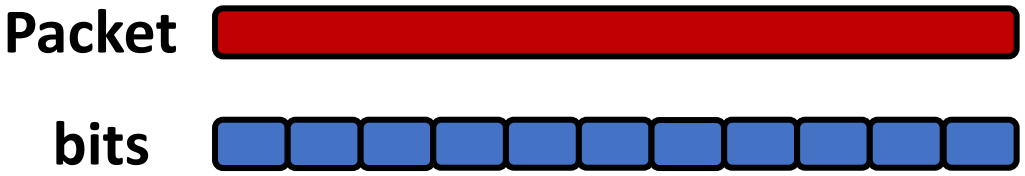


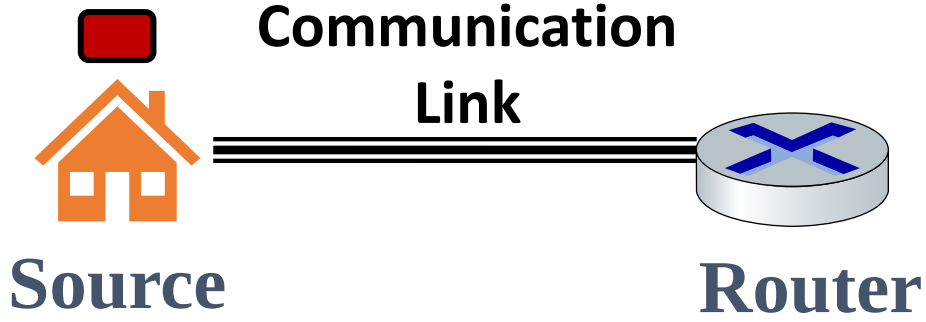
Step 1: Transmit the packets into the link



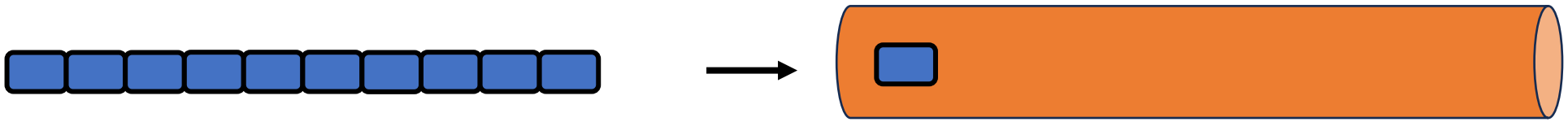
Link capacity: 10 bit/sec

How to send a packet via network



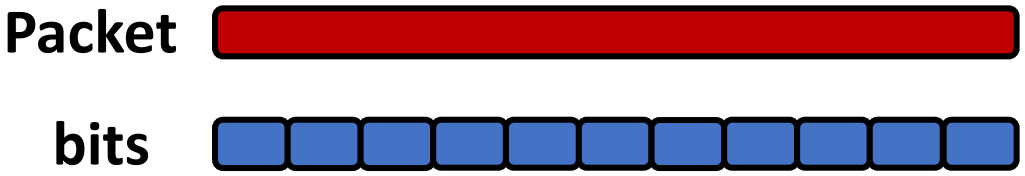


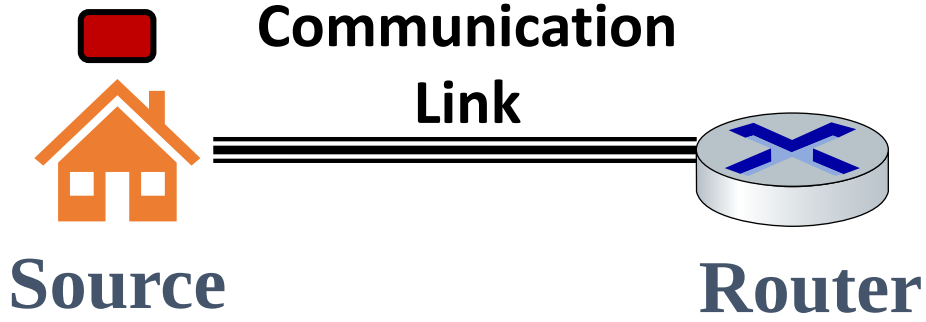
Step 1: Transmit the packets into the link



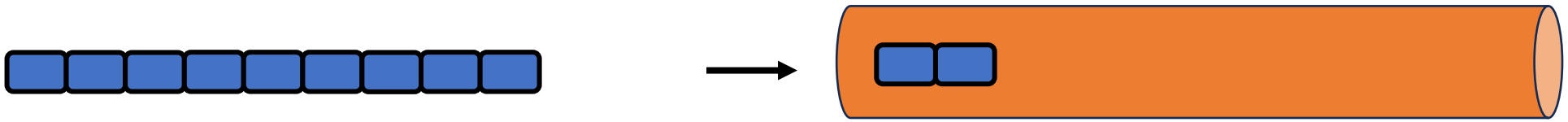
Link capacity: 10 bit/sec

How to send a packet via network



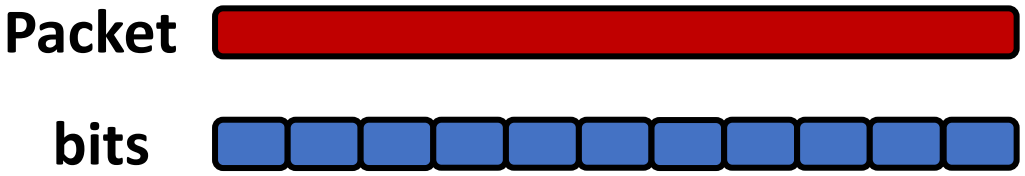


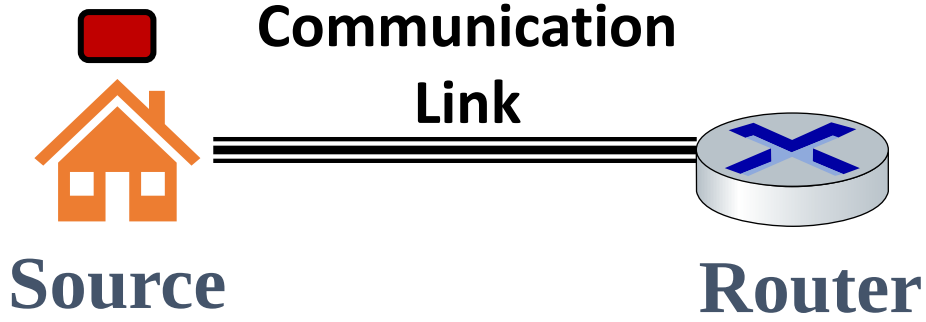
Step 1: Transmit the packets into the link



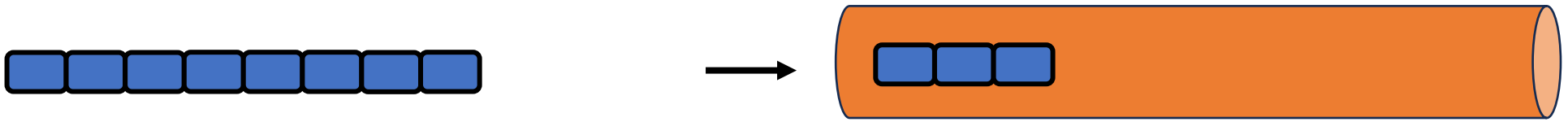
Link capacity: 10 bit/sec

How to send a packet via network



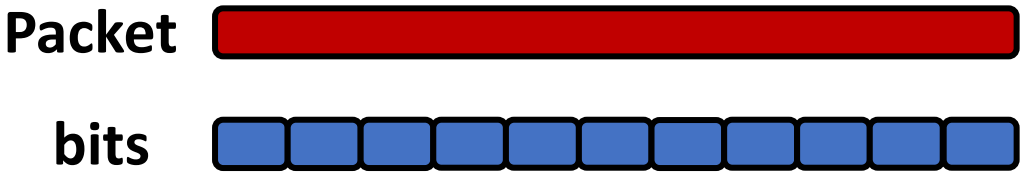


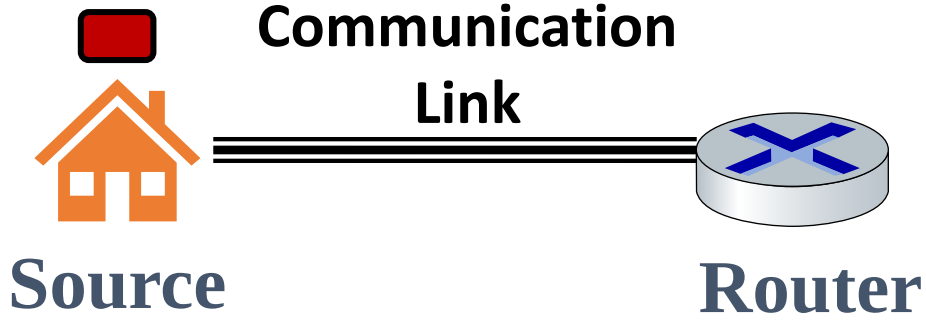
Step 1: Transmit the packets into the link



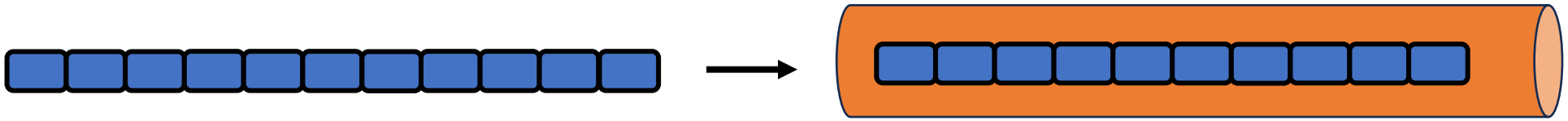
Link capacity: 10 bit/sec

How to send a packet via network



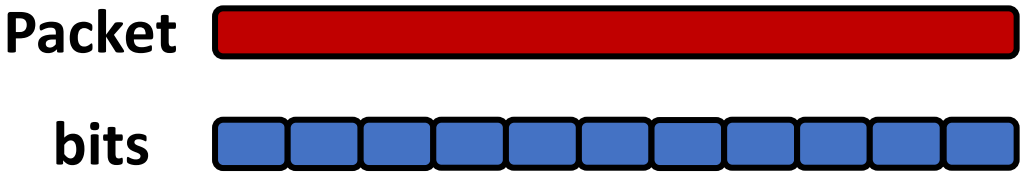


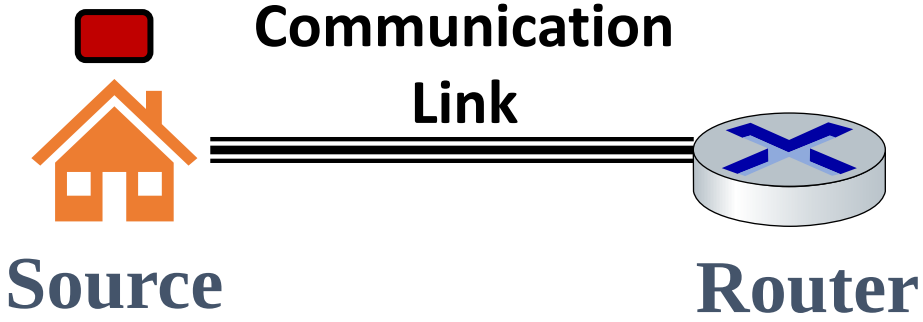
Step 1: Transmit the packets into the link



Link capacity: 10 bit/sec

How to send a packet via network

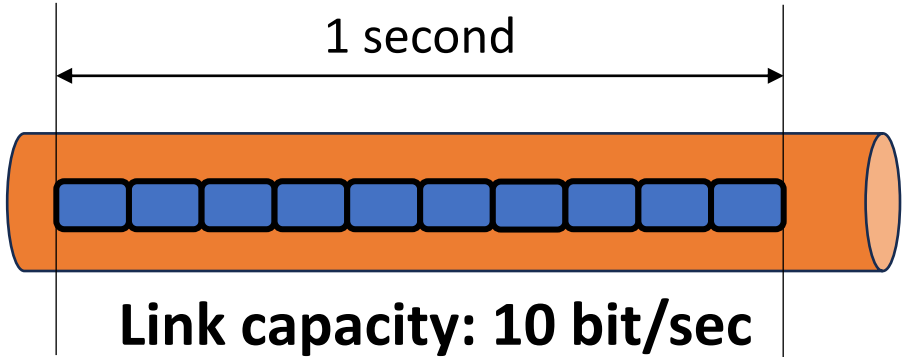




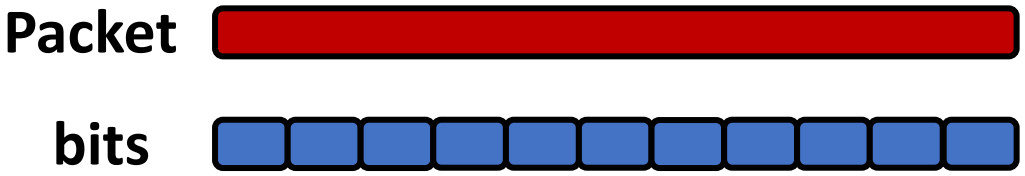
Step 1: Transmit the packets into the link

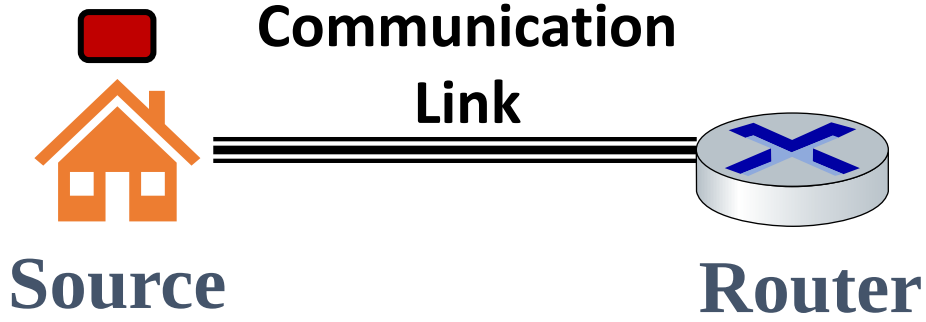
d_{trans}: transmission delay:

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



How to send a packet via network

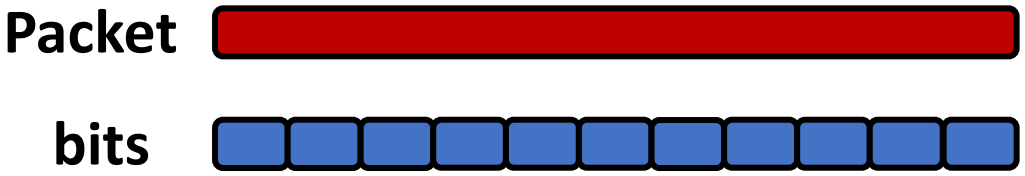


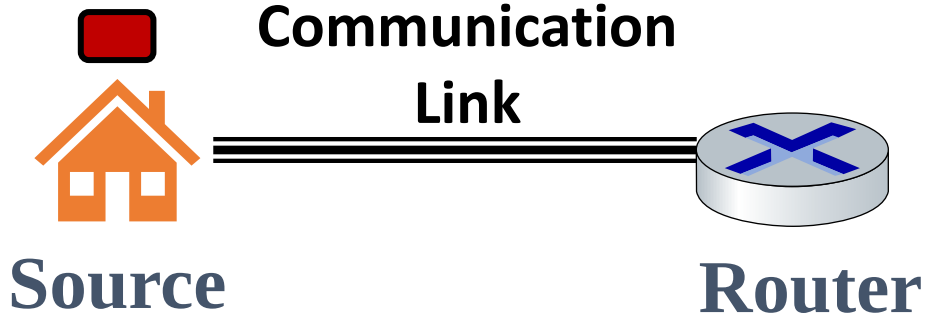


Step 1: Transmit the packets into the link

Step 2: The packet bits propagates to the router

How to send a packet via network



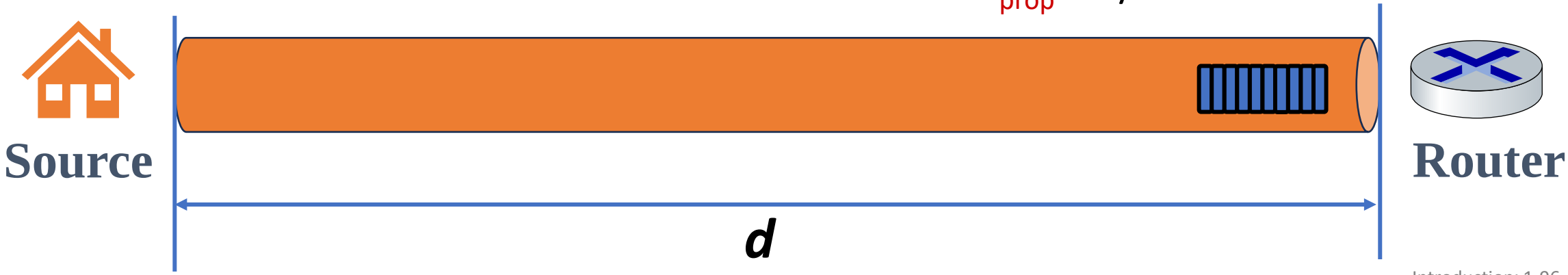


Step 1: Transmit the packets into the link

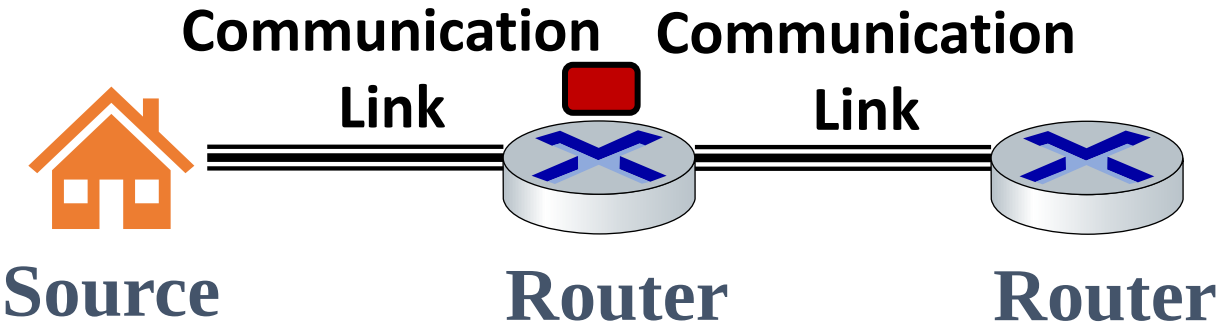
Step 2: The packet bits propagates to the router

d_{prop}: propagation delay:

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



How to send a packet via network

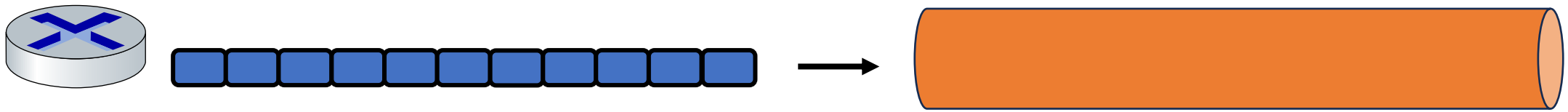


Step 1: Transmit the packets into the link

Step 2: The packet bits propagates to the router

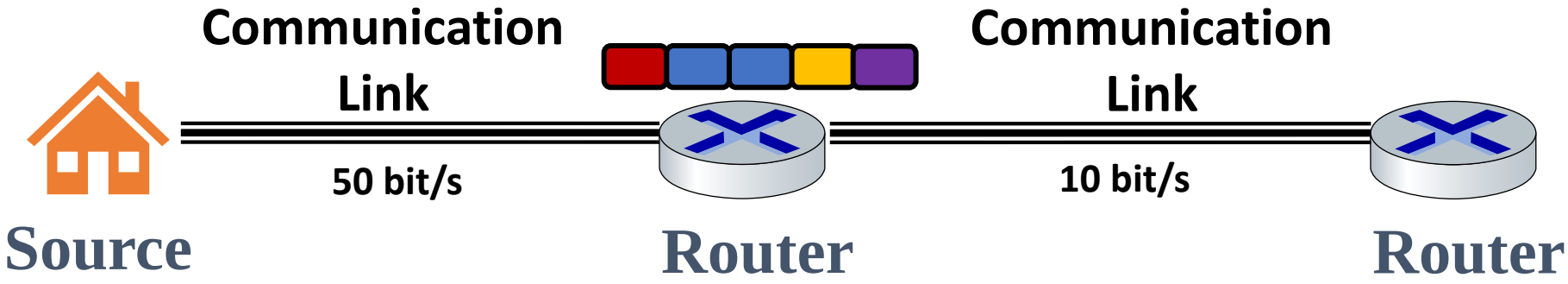
d_{trans}: transmission delay:

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



Link capacity: 10 bit/sec

How to send a packet via network



Key point:

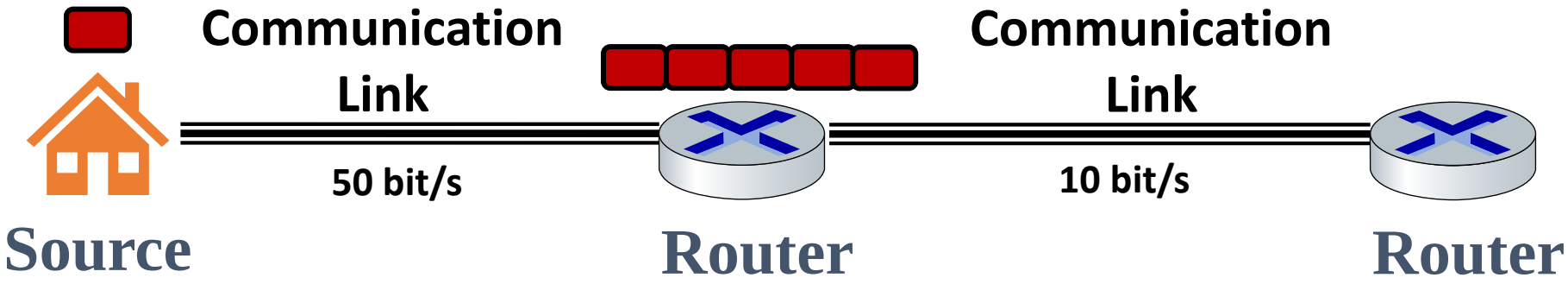
- Router takes transmission delay to transmit a packet to the link
- The packet may arrive faster than the packets get out of the router
- The later arrived packets must wait at the router until all the packets arriving before it are transmitted into the link

d_{queue}: queueing delay

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

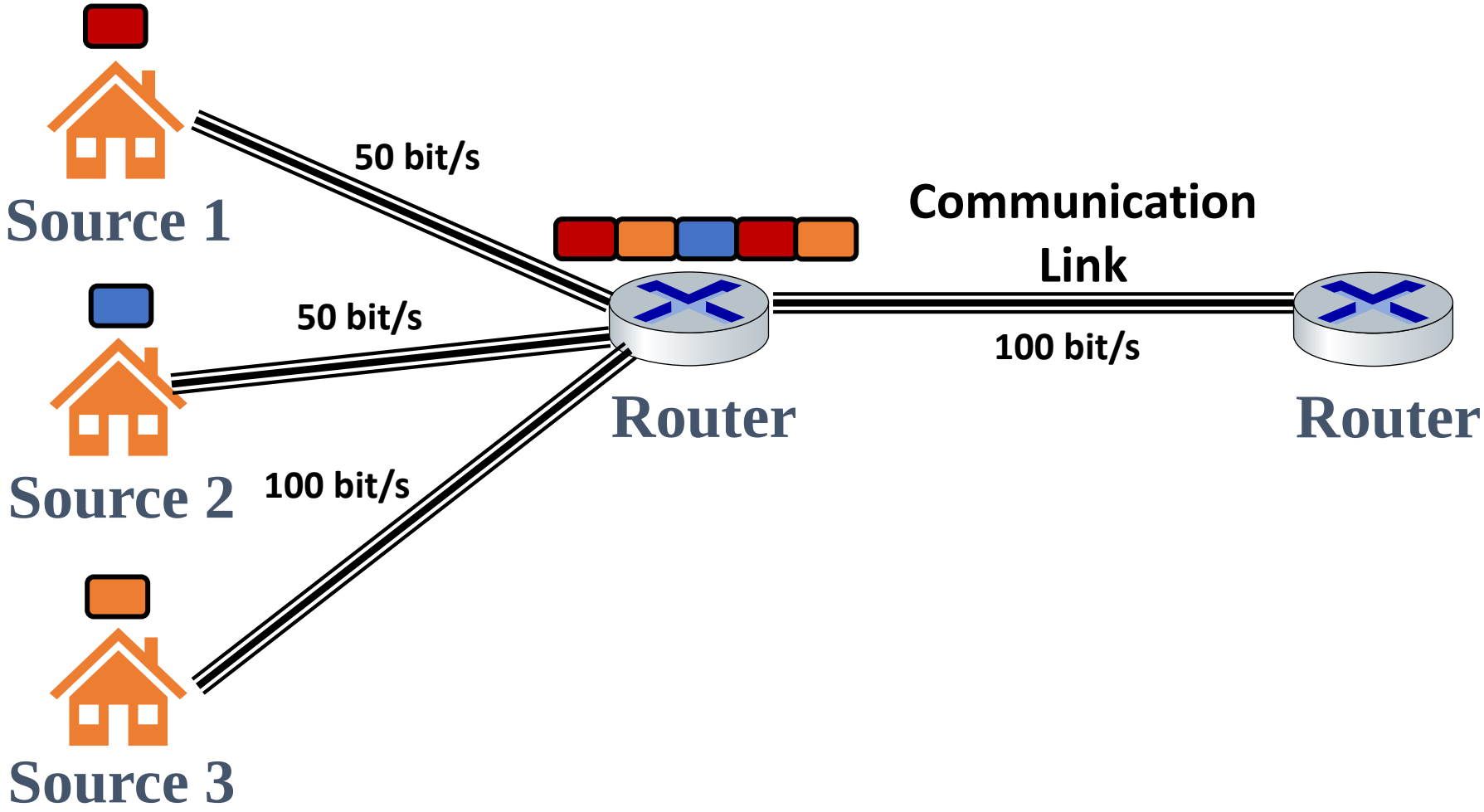
How to send a packet via network

Various reasons of queuing inside the router

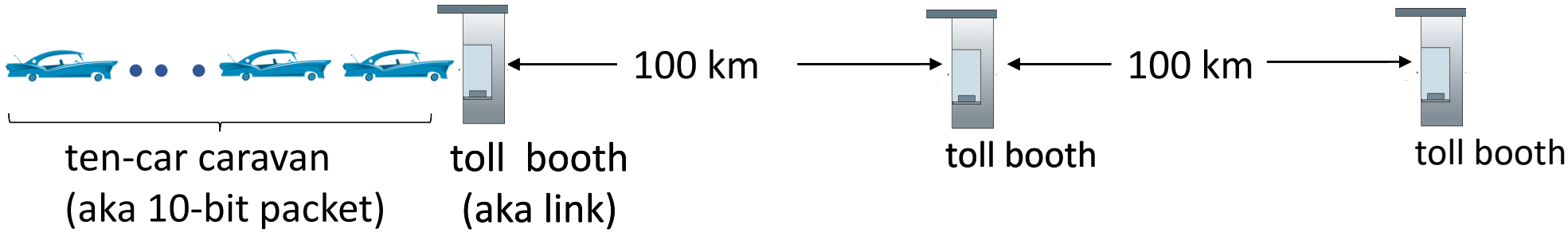


How to send a packet via network

Various reasons of queuing inside the router



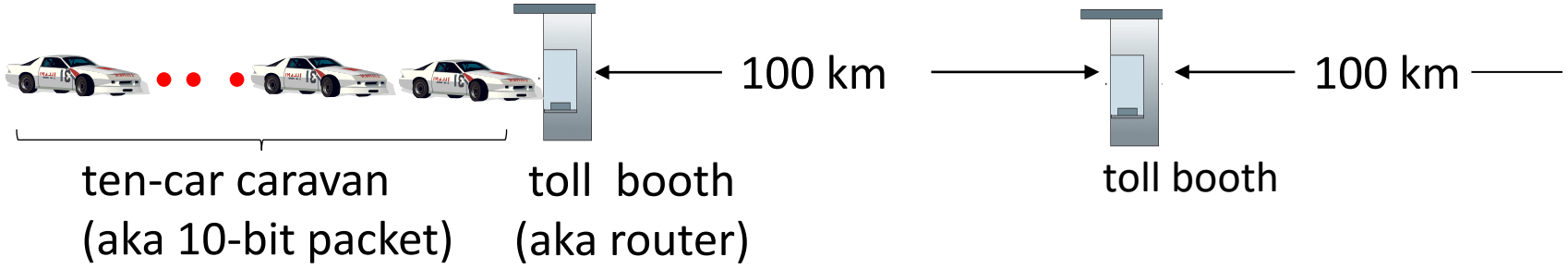
Caravan analogy



- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- "propagate" at 100 km/hr
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr) = 1 hr
- *A:* 62 minutes

Caravan analogy



- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will some cars arrive at 2nd booth before all cars serviced at first booth?

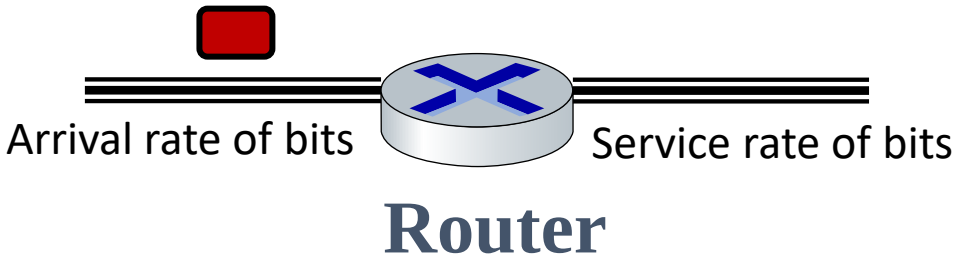
  A: Yes! after 7 min, first car arrives at second booth; three cars still at first booth

Packet queueing delay (revisited)

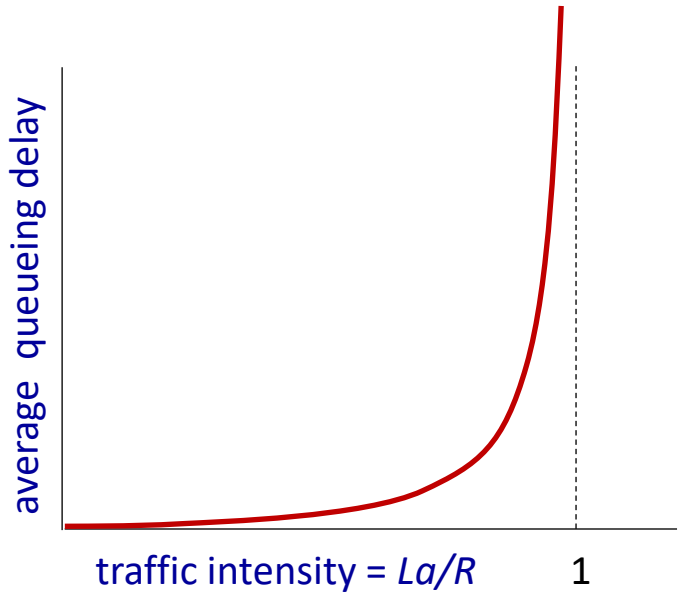
- *a:* average packet arrival rate
- *L:* packet length (bits)
- *R:* link bandwidth (bit transmission rate)

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

"traffic intensity"

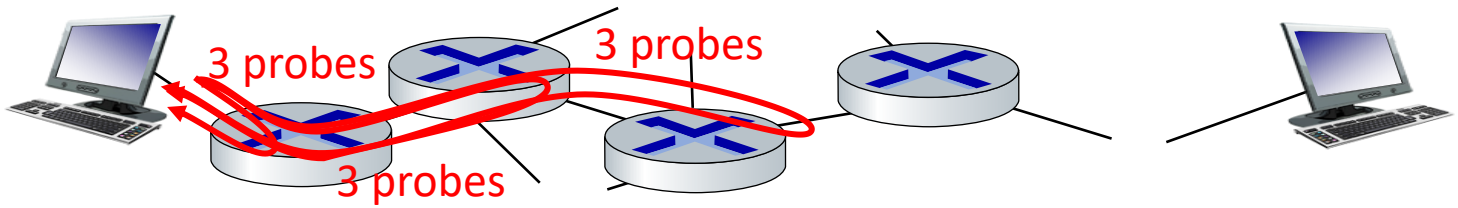


- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!



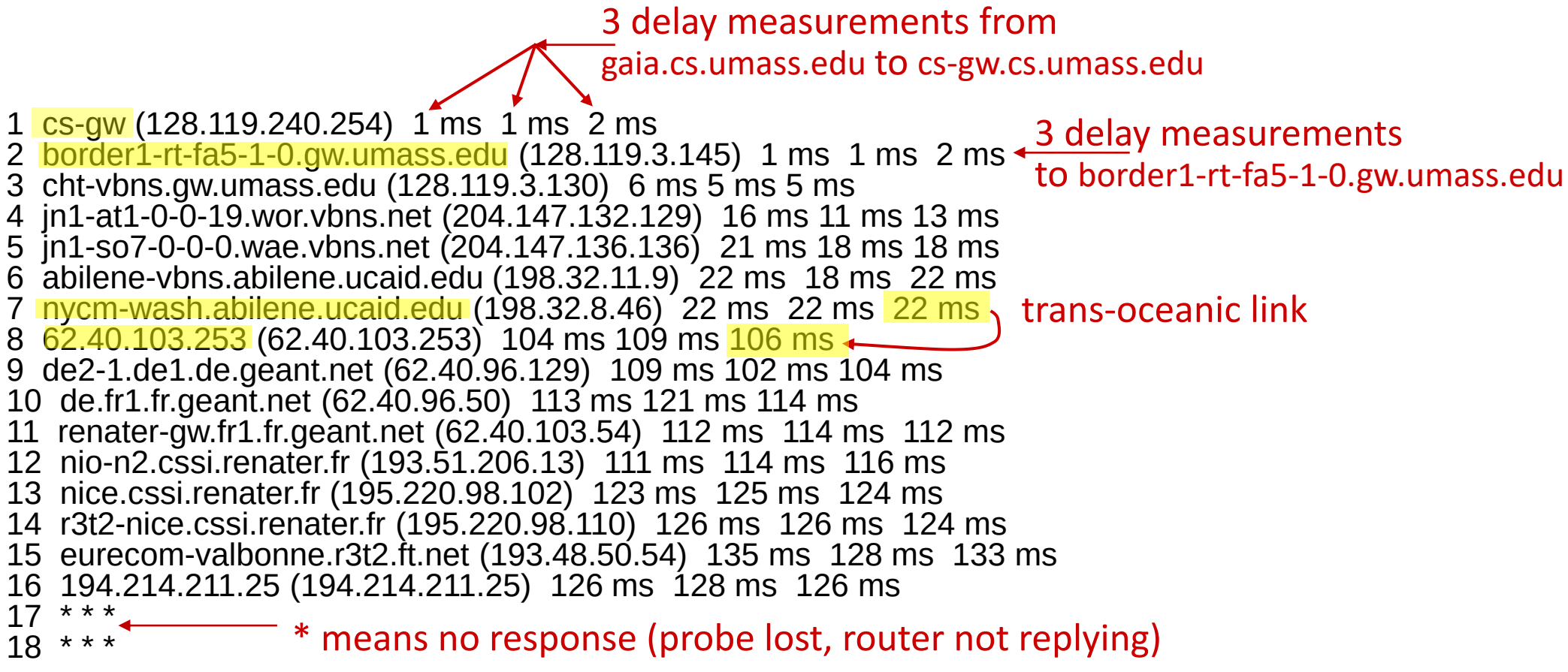
“Real” Internet delays and routes

- 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

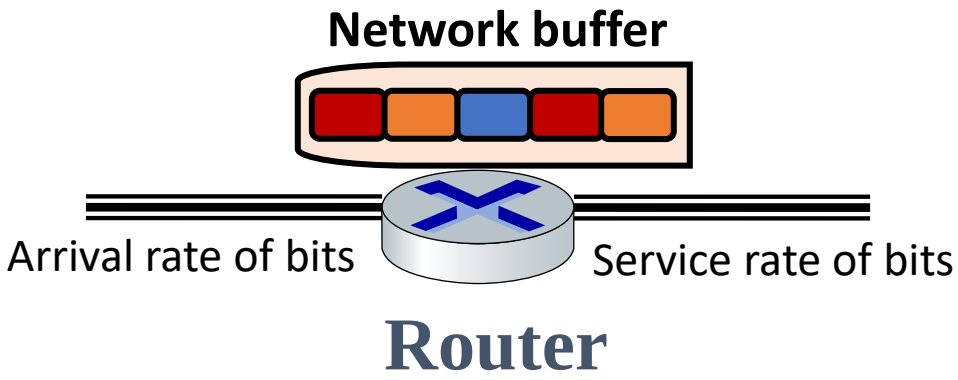
traceroute: gaia.cs.umass.edu to www.eurecom.fr



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

Packet loss

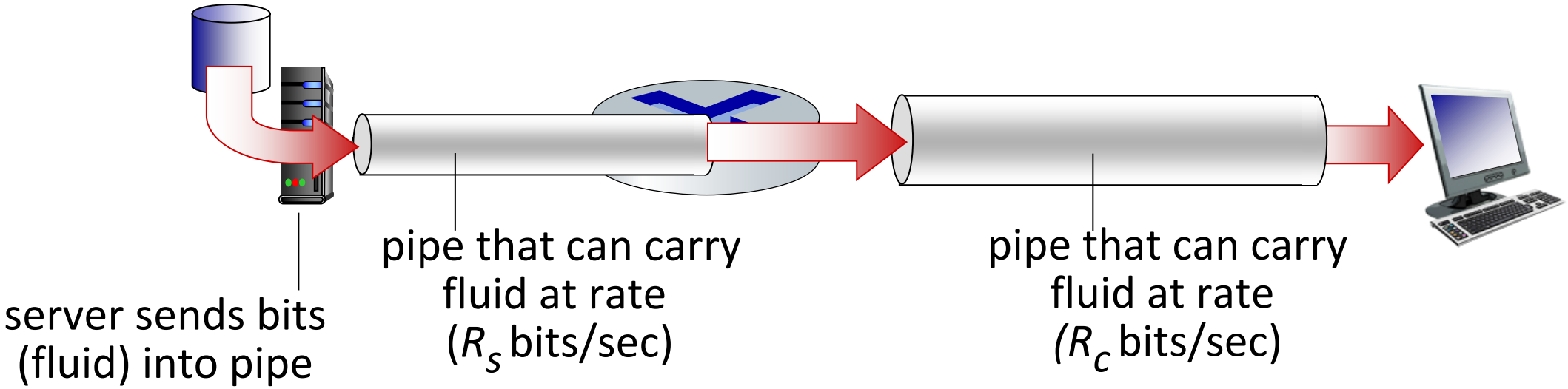
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving at a full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



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

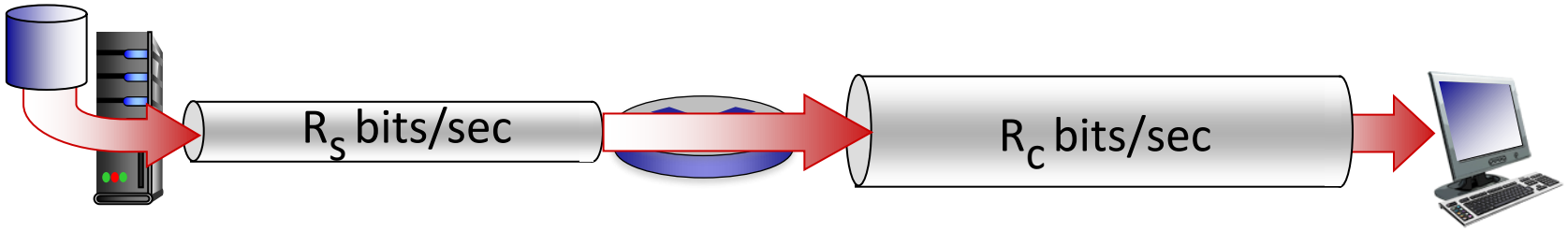
Throughput

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

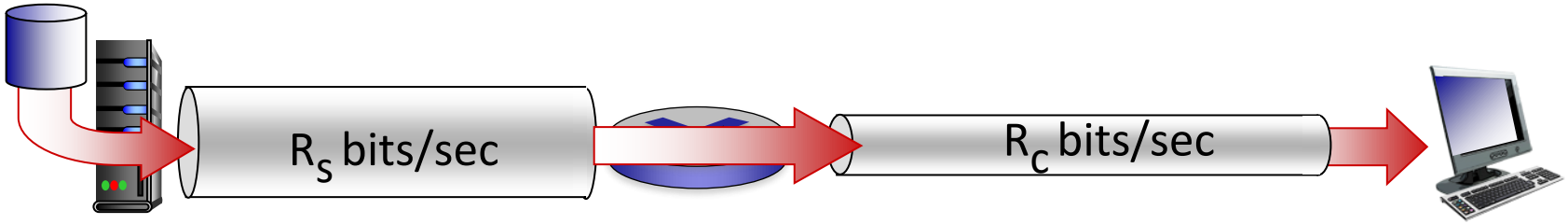


Throughput

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



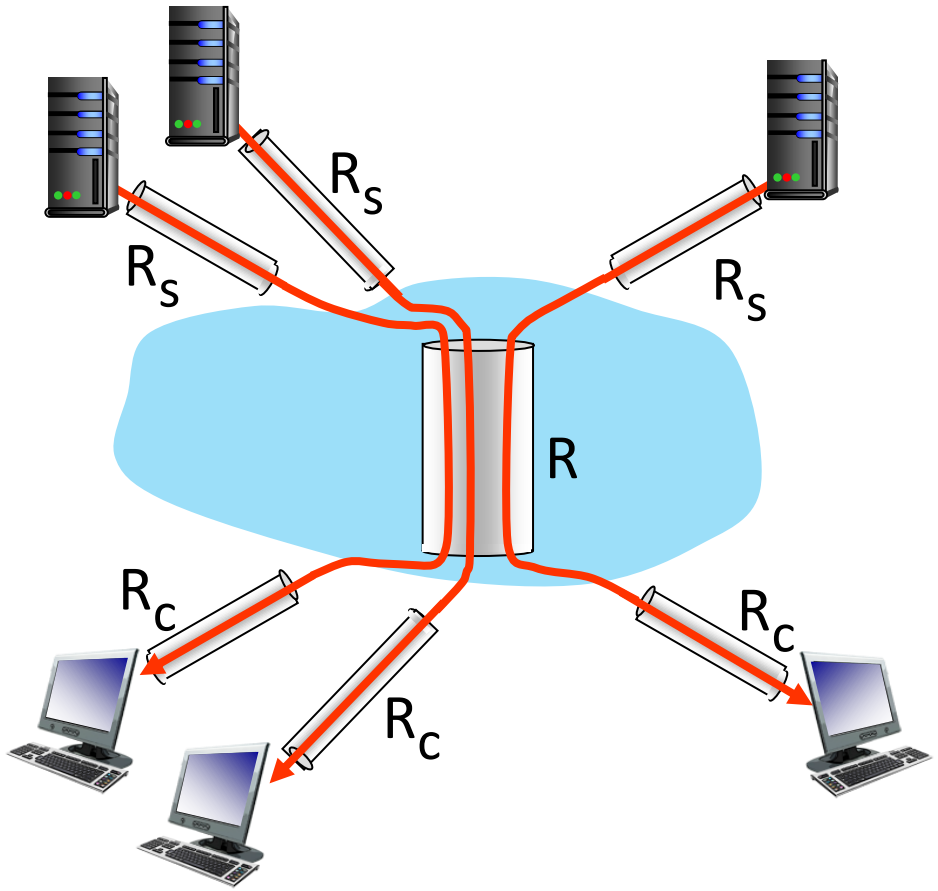
$R_s > R_c$ What is average end-end throughput?



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/