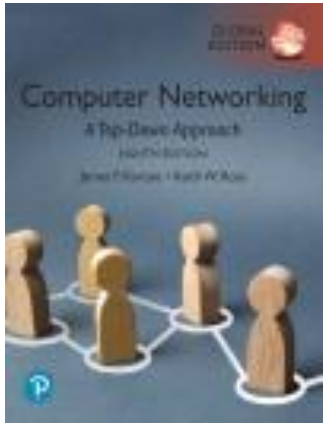
Topic 6: Introduction to Layering

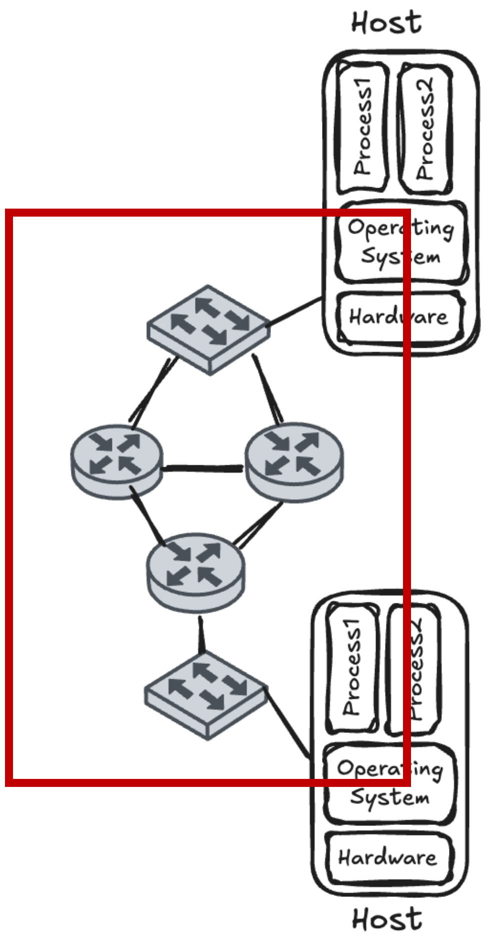
SCC.231 – Computer Networks and Systems



[Chapter 1: Computer Networks and the Internet]

Lecture Plan

- Moving from OS to Networking
  - Protocols
  - Internet Structure
  - Forwarding and routing
  - Abstraction and Layering
- Rest of the networking topic will focus on each layer and present the protocols
- Introduction to Mininet



What’s a Protocol?

Definition

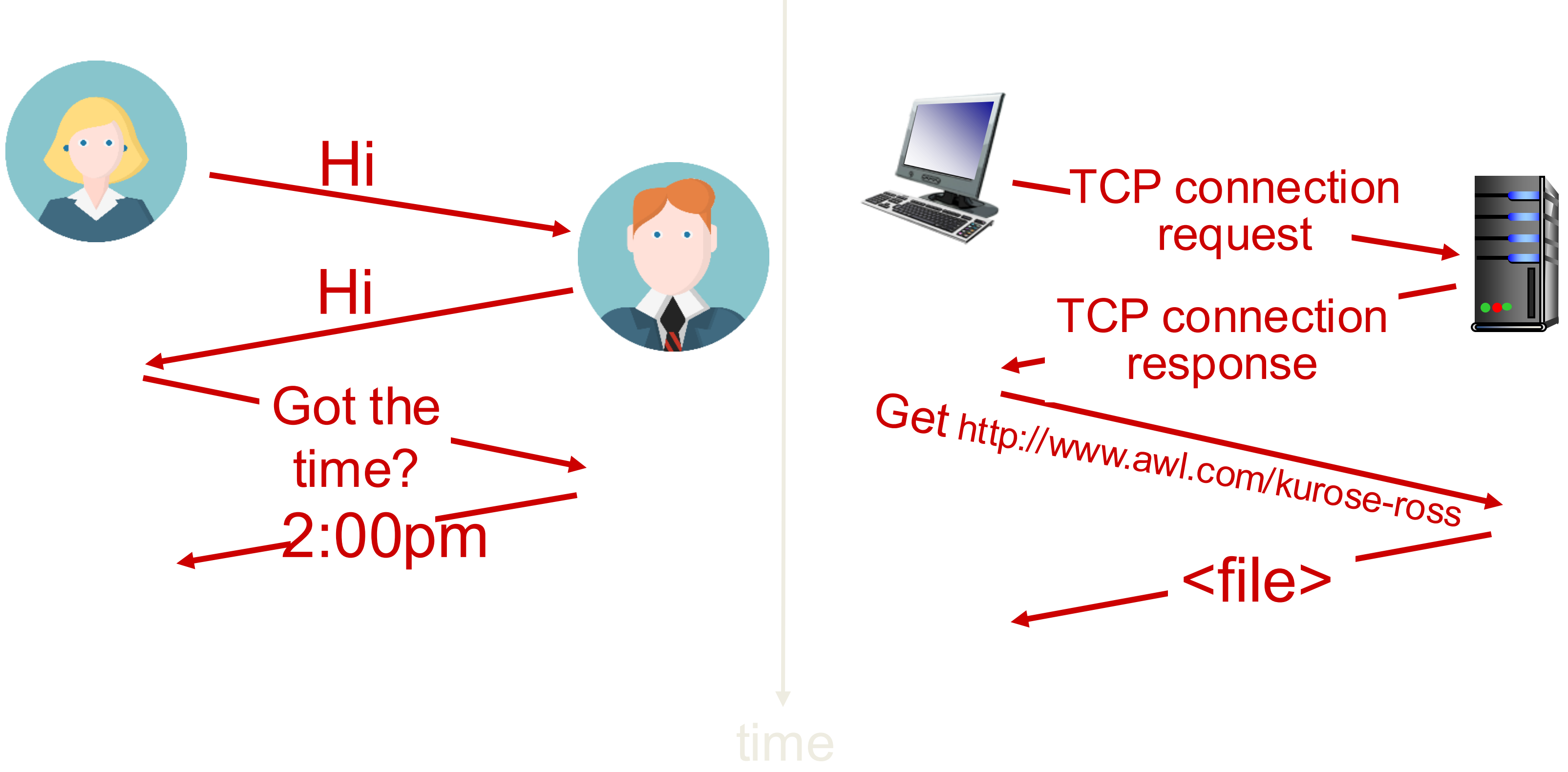
- The Internet is one of the biggest distributed multi-administrative systems
  - Interoperability is key to ensure operation

A Protocol defines the format, order of messages exchanged among network entities (e.g., hosts, routers), and actions to be taken on message transmission/receipt/non-receipt

What's a Protocol?

Comparison

A human protocol and a computer network protocol:



What's a Protocol?

Human vs. Network protocols

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:

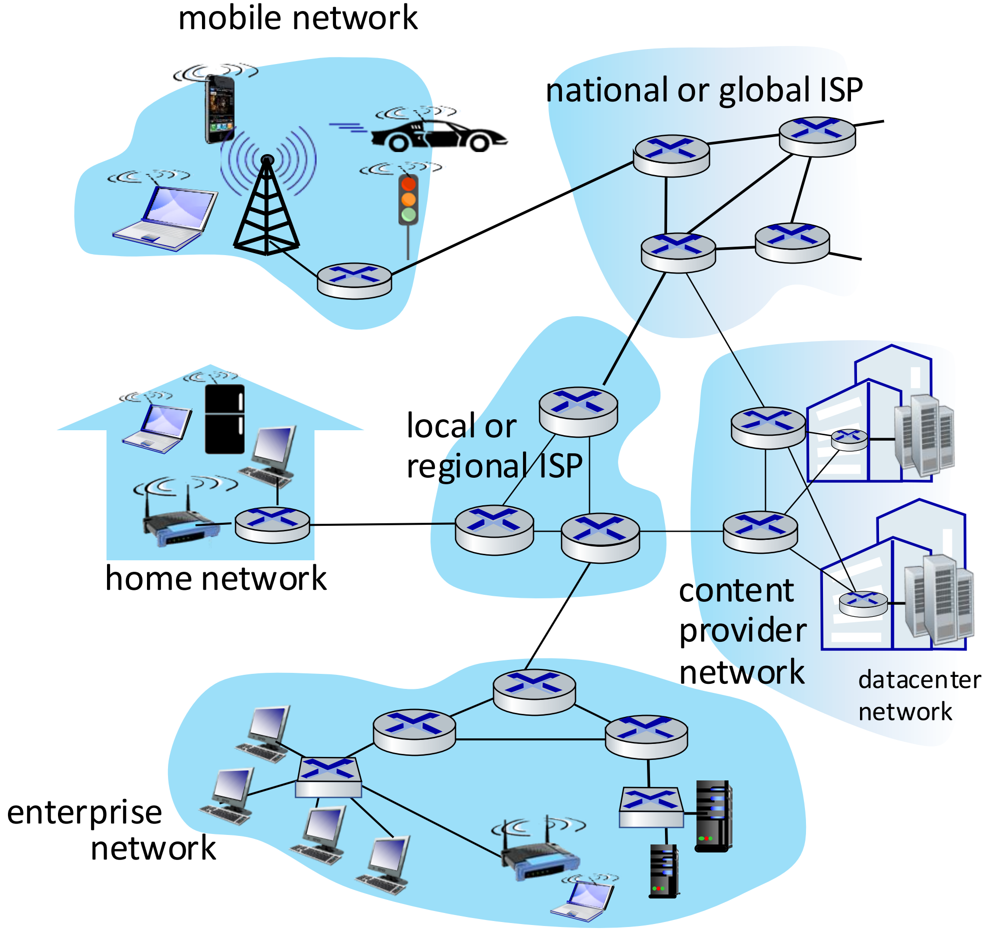
- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

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

Internet Structure

Internet structure: a "network of networks"

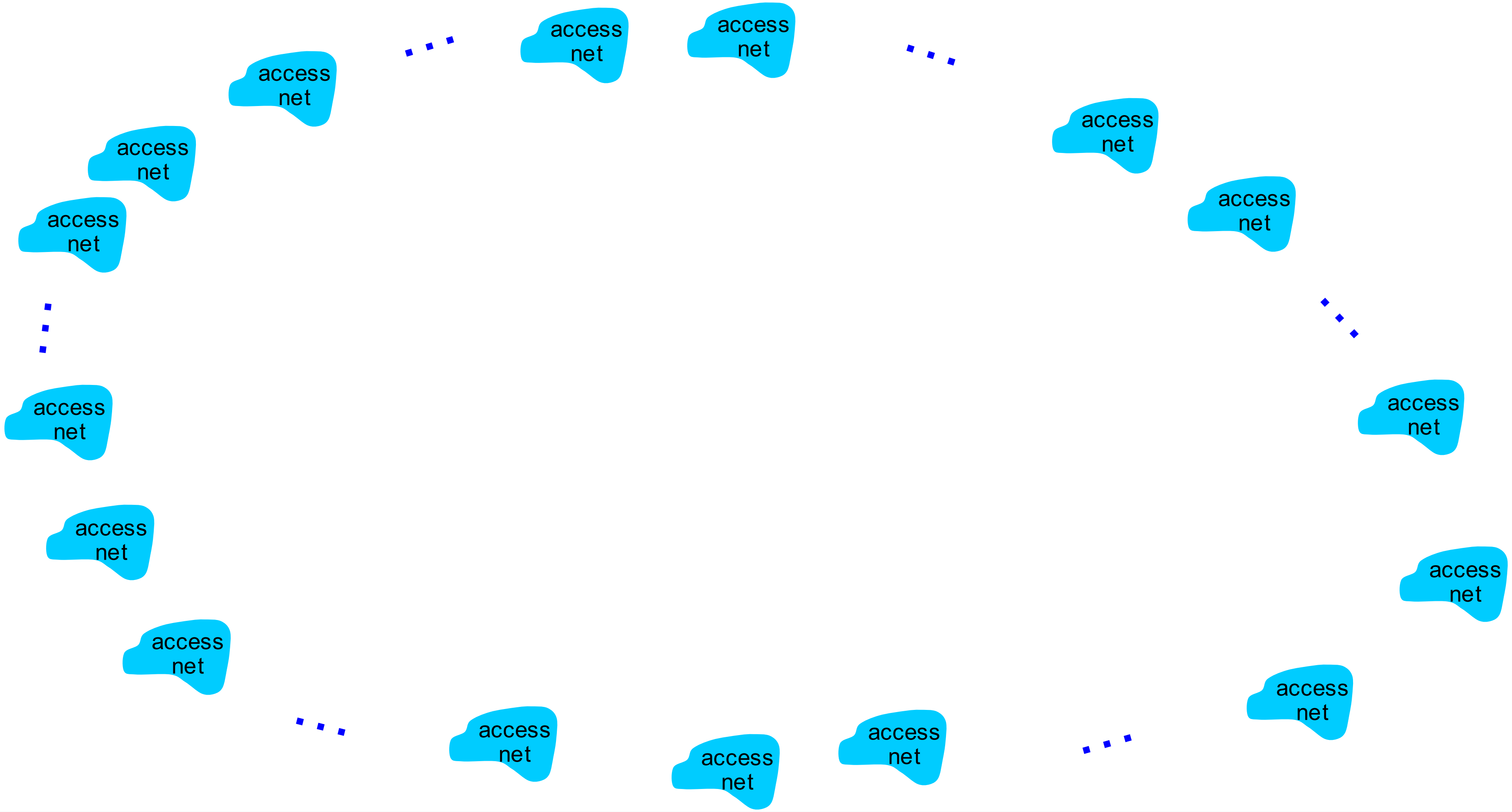
- Hosts connect to Internet via access Internet Service Providers (ISPs)
- Access ISPs in turn must be interconnected
  - So that *any* two hosts *(anywhere!)* can send packets to each other
- Resulting network of networks is very complex
  - Evolution driven by economics, national policies



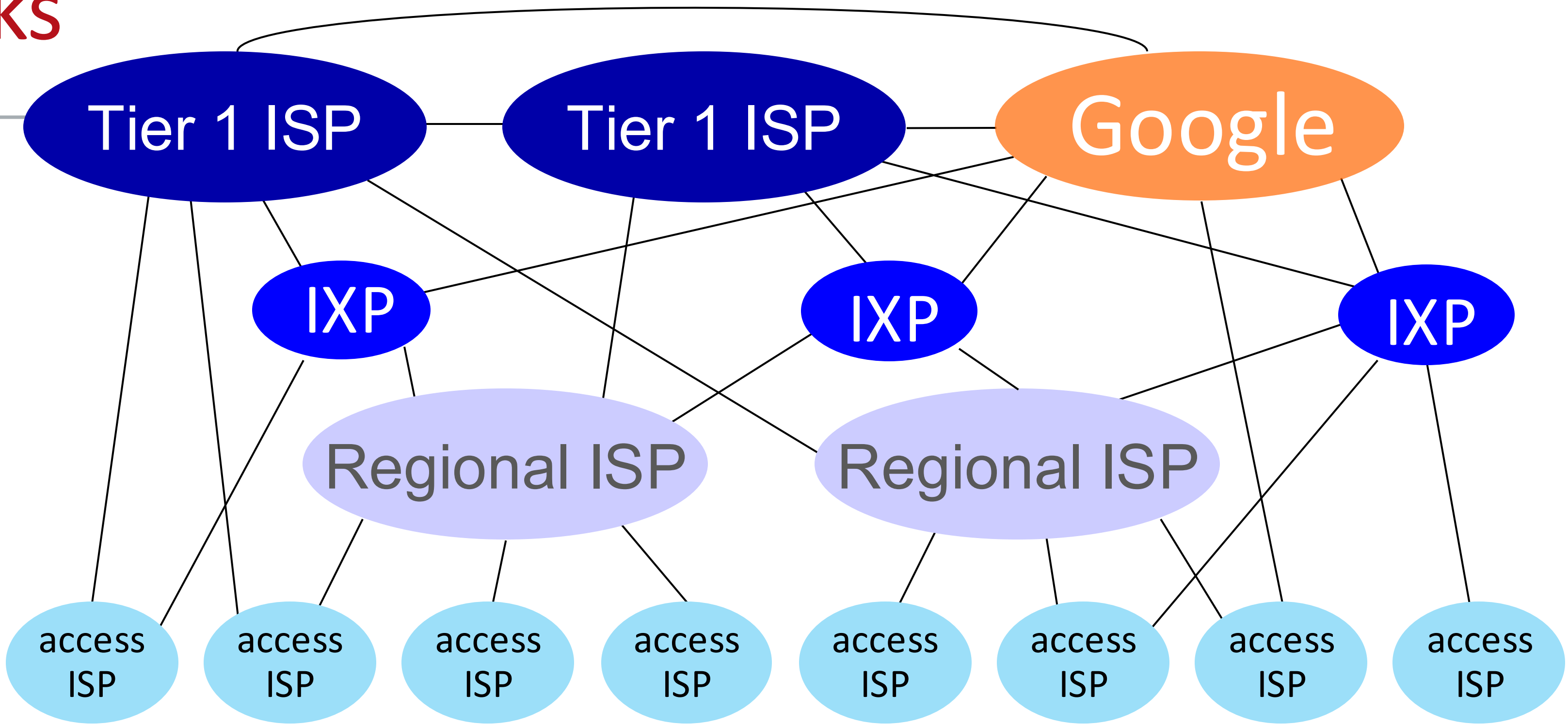
Let's take a stepwise approach to describe current Internet structure

Internet structure: a "network of networks"

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



Internet structure: a “network of networks”



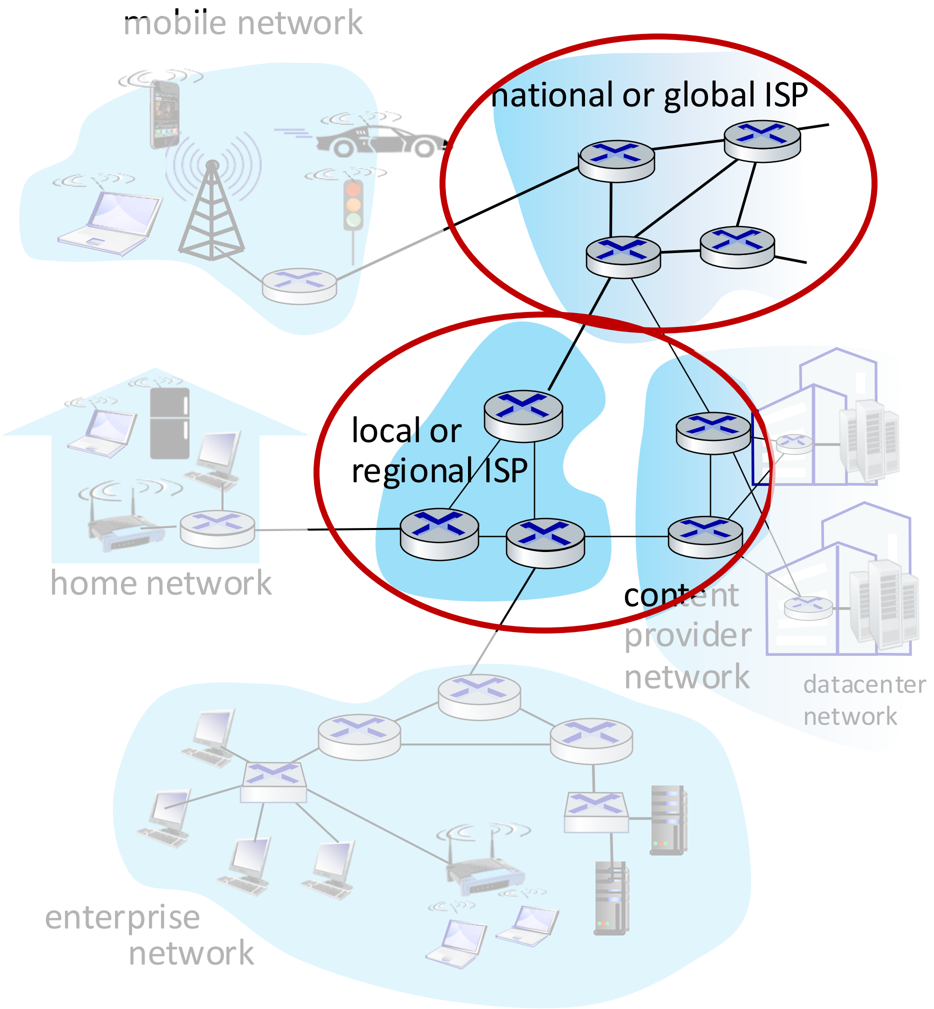
At “center”: small # of well-connected large networks

- “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 Core

The network core

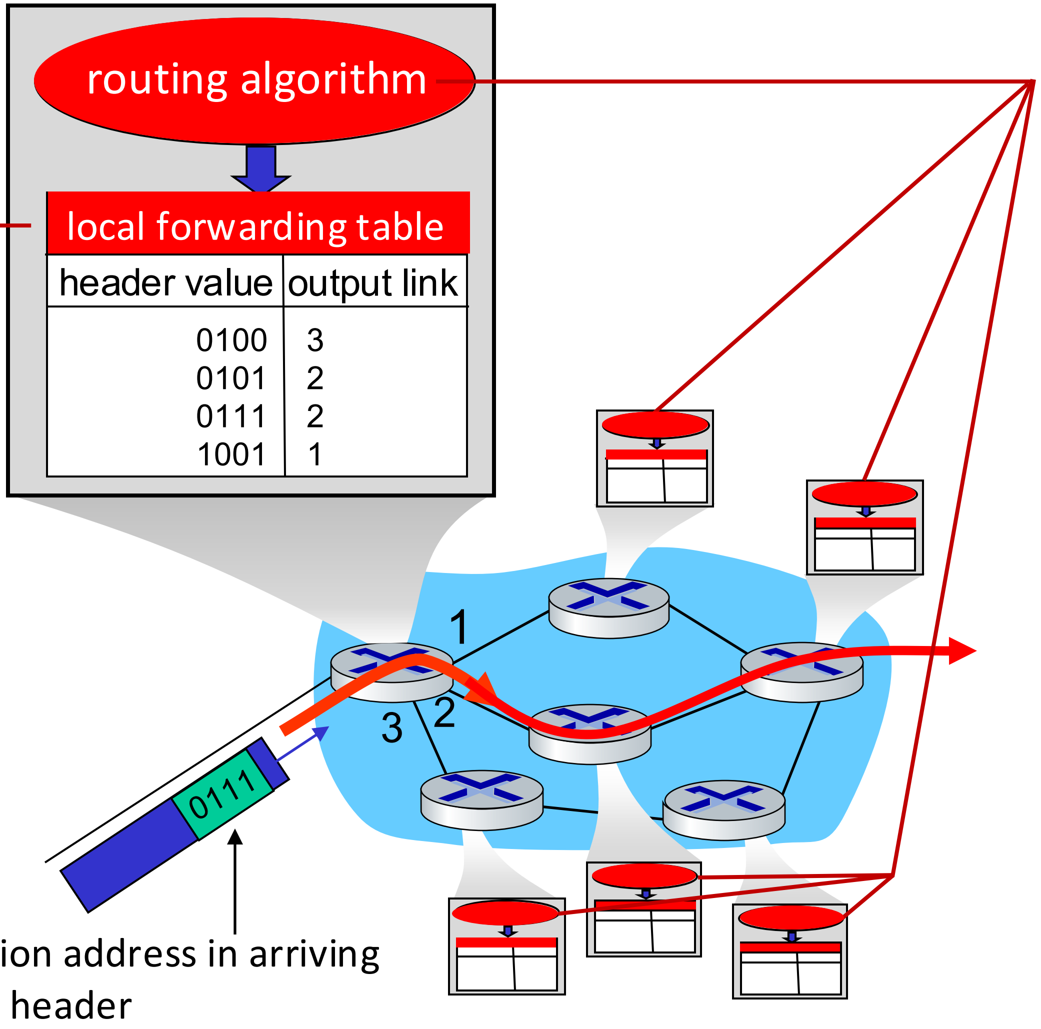
- Mesh of interconnected routers
- Packet-switching: hosts break application-layer messages into *packets*
  - Network forwards packets from one router to the next, across links on path from source to destination



Two key network-core functions

Forwarding:

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



Routing:

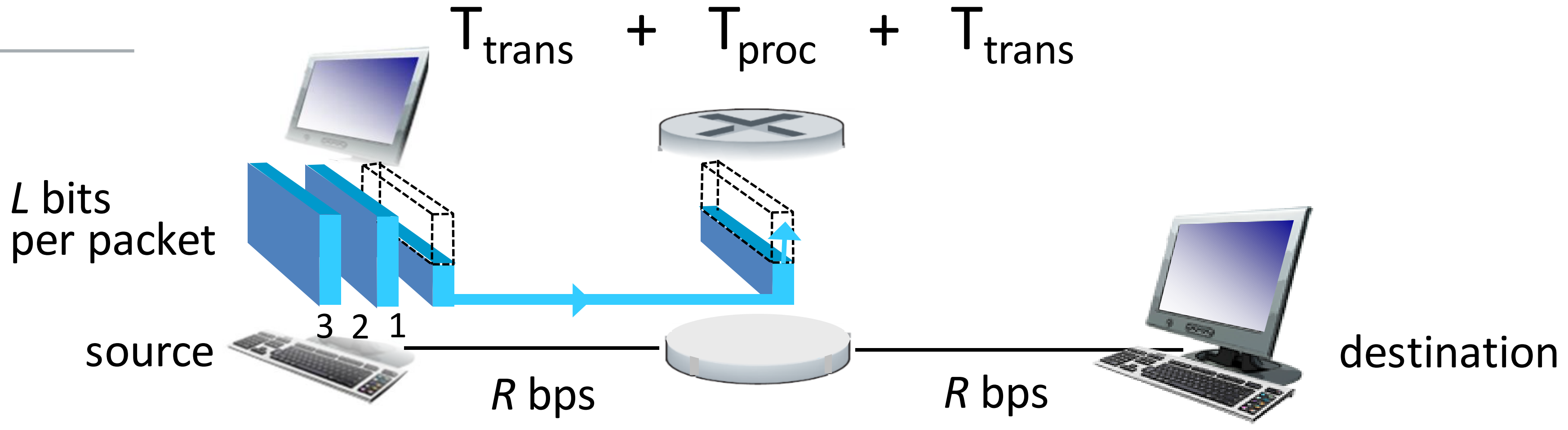
- *global* action: determine source-destination paths taken by packets
- routing algorithms


routing
45 h
3,053 miles
48 h
3,309 miles
San Jose
Northampton
United States
Victoria
Seattle
WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
Quebec City
Montreal
Ottawa
MAINE
Portland
OREGON
IDAHO
WYOMING
SOUTH DAKOTA
Minneapolis
WISCONSIN
MICHIGAN
Toronto
VERMONT
NEW HAMPSHIRE
Detroit
Chicago
IOWA
NEVADA
UTAH
COLORADO
Kansas City
ILLINOIS
INDIANA
OHIO
PENNSYLVANIA
New York
Philadelphia
NJ
CT
RI
MARYLAND
DE
Indianapolis
WEST VIRGINIA
VIRGINIA
KENTUCKY
Sacramento
San Francisco
CALIFORNIA
Las Vegas
Nashville
TENNESSEE
NORTH CAROLINA
Charlotte
OKLAHOMA
ARKANSAS
Los Angeles
ARIZONA
NEW MEXICO
Atlanta
SOUTH CAROLINA
MISSISSIPPI
ALABAMA
GEORGIA
San Diego
Phoenix
Dallas
Tucson
El Paso
TEXAS
BAJA CALIFORNIA
SONORA
CHIHUAHUA
Austin
Houston
LOUISIANA
New Orleans
Jacksonville
San Antonio
COAHUILA
Orlando
Tampa
FLORIDA

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

Switching

Packet-switching: store-and-forward

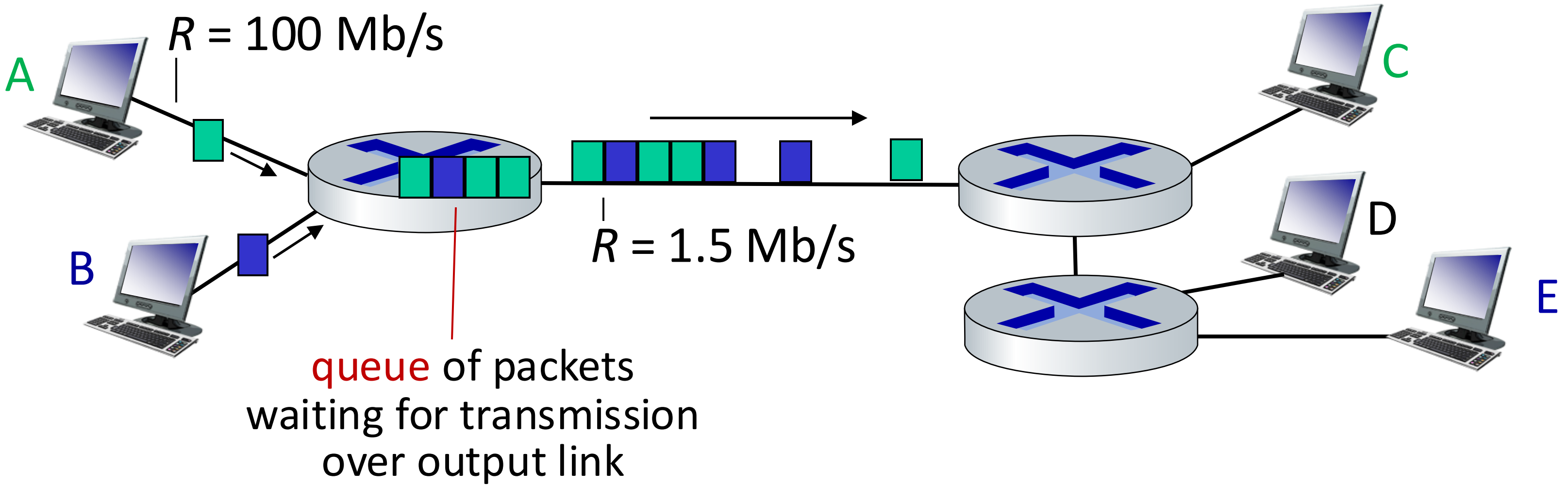


- packet transmission delay: takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link
- *packet processing delay: the time a network takes to examine, process and transmit a packet's header.*

One-hop numerical example:

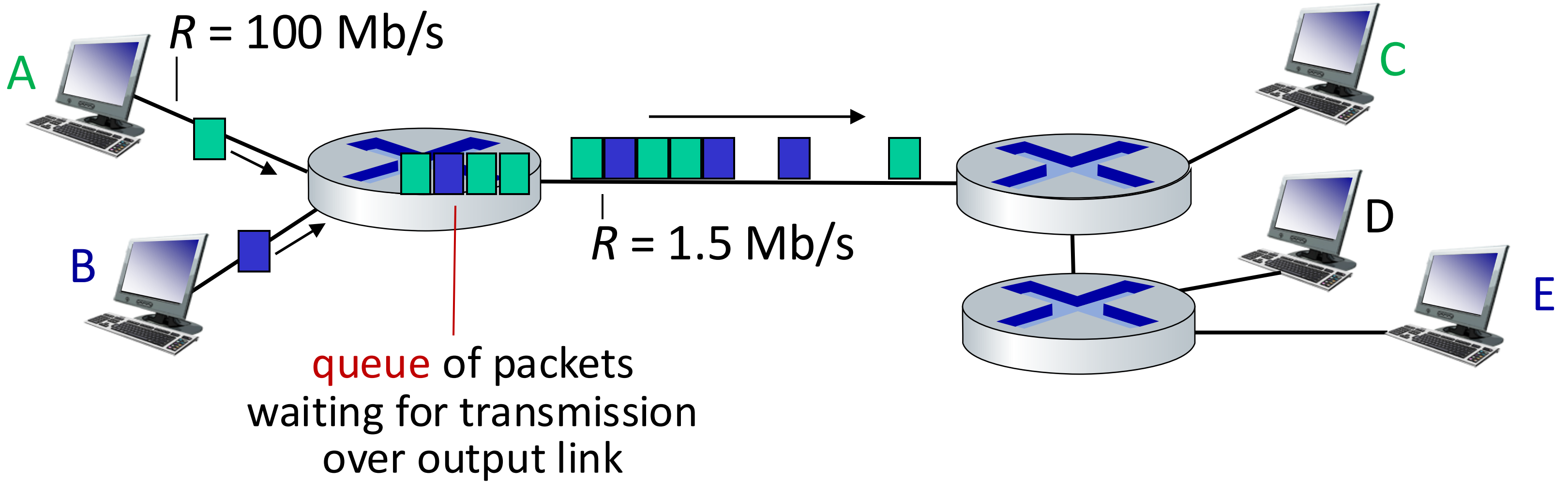
- *L* = 10 Kbits
- *R* = 100 Mbps
- one-hop transmission delay = 0.1 msec

Packet-switching: queueing



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

Packet-switching: queueing

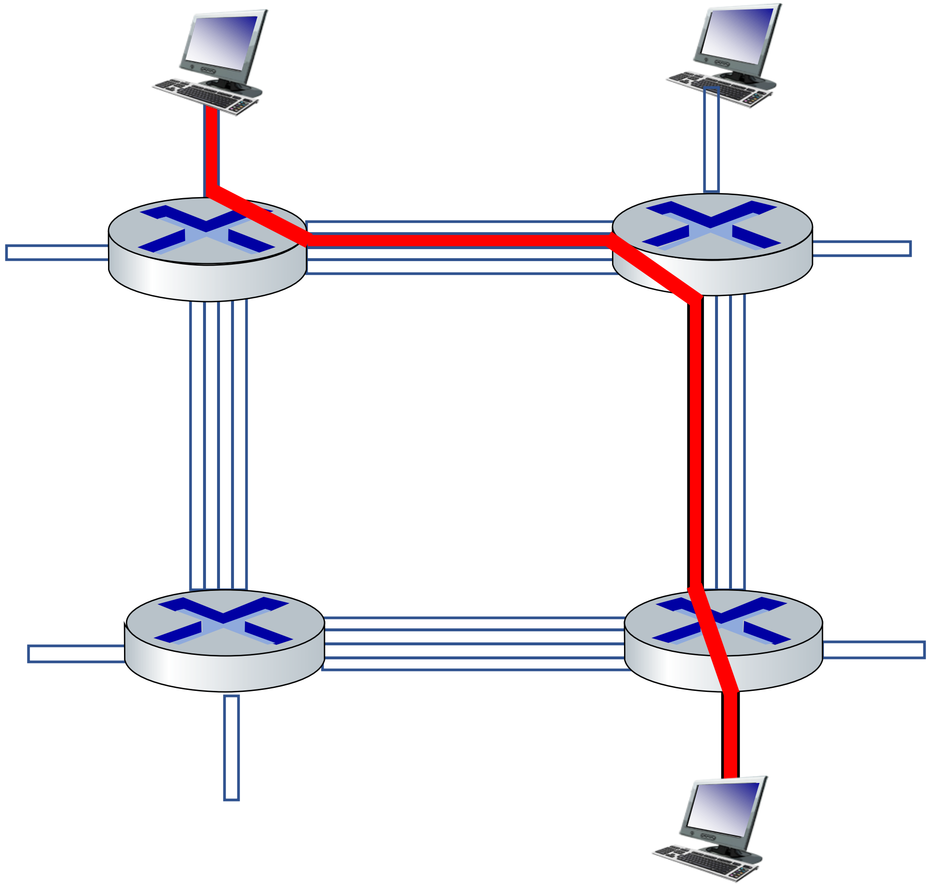


Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

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

Alternative to packet switching: circuit switching

end-end resources allocated to, reserved for "call" between source and destination



- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
  - commonly used in traditional telephone networks

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

Circuit switching: FDM and TDM

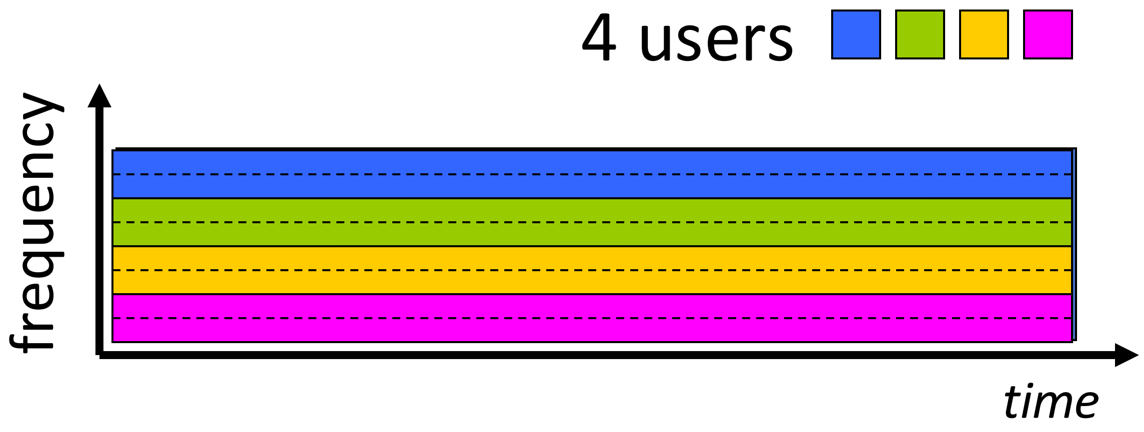
Lancaster University

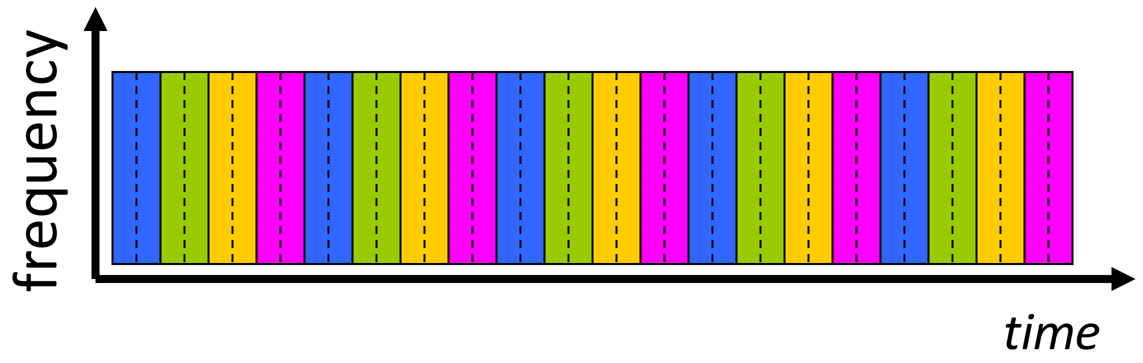
Frequency Division Multiplexing (FDM)

- Optical, electromagnetic frequencies divided into (narrow) frequency bands
- Each call allocated its own band, can transmit at max rate of that narrow band

Time Division Multiplexing (TDM)

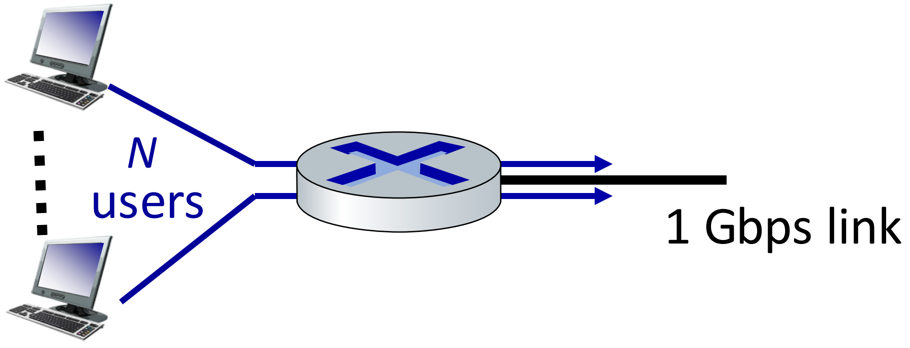
- Time divided into slots
- Each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band (only) during its time slot(s)





Packet switching versus circuit switching

example:
- 1 Gb/s link
- each user:
  - 100 Mb/s when “active”
  - active 10% of time



Q: how many users can use this network under circuit-switching and packet switching?

- *circuit-switching:* 10 users
- *packet switching:* with 35 users, probability > 10 active at same time is less than .0004 *

Q: how did we get value 0.0004?

A: homework problem

Check the binomial distribution

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

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.
- *Q:* human analogies of reserved resources (circuit switching) versus on-demand allocation (packet switching)?

Introducing Abstractions

Wireshark · Packet 16 · data

```
> Frame 16: 870 bytes on wire (6960 bits), 870 bytes captured (6960 bits)
> Ethernet II, Src: 22:b7:bc:26:09:e3, Dst: bc:f8:7e:40:b4:be
> Internet Protocol Version 6, Src: 2a00:23c7:5b4d:5301:9daf:b2dc:3586:724
> Transmission Control Protocol, Src Port: 49664, Dst Port: 443, Seq: 1912
> [2 Reassembled TCP Segments (952 bytes): #15(168), #16(784)]
v Secure Sockets Layer
  > TLSv1.2 Record Layer: Application Data Protocol: http2
```

```
0000  bc f8 7e 40 b4 be 22 b7  bc 26 09 e3 86 dd 60 0d   ..~@..". .&....`.
0010  02 00 03 30 06 40 2a 00  23 c7 5b 4d 53 01 9d af   ...0.@*. #.[MS...
0020  b2 dc 35 86 72 47 26 20  01 00 60 20 00 13 00 00   ..5.rG&  ..` ....
0030  00 00 a2 7d 40 0d c2 00  01 bb bc d9 19 94 1b 1b   ...}@... ........
0040  73 ce 80 18 08 0b 1d df  00 00 01 01 08 0a 9c ae   s....... ........
0050  e8 54 a0 91 cd a9 0b 17  a3 30 c3 50 8c 3d bc 41   .T...... .0.P.=.A
0060  c4 23 a3 c9 80 09 9e a3  67 79 23 9a a9 44 b4 16   .#...... gy#..D..
0070  a9 90 d2 59 79 8e a0 0e  bc 44 a8 f2 86 c8 9c 69   ...Yy... .D.....i
0080  b7 c4 71 1d 4b de 29 77  55 48 f0 ba 3b a0 4a 3e   ..q.K.)w UH..;.J>
0090  08 f2 77 3a 3c d0 be 47  e1 c7 3b 59 e0 f8 0f ee   ..w:<..G ..;Y....
00a0  c8 94 88 de 83 4f b7 5c  53 45 4e e8 ba 59 b2 b2   .....O.\ SEN..Y..
00b0  ef 92 bf 77 43 f1 8d 58  37 4d 99 e4 4b 32 eb 4a   ...wC..X 7M..K2.J
00c0  c2 d4 48 29 fd 6f 25 c8  bb 66 ca d6 bc 45 95 7e   ..H).o%. .f...E.~
00d0  a8 ce 71 d4 40 fc f0 9d  b2 93 16 31 55 05 44 ae   ..q.@... ...1U.D.
00e0  c1 ce df 1f 2e 3f 5f e8  ba 6f 5e 2e 79 dd fb 98   .....?_. .o^.y...
00f0  8c a2 8b c8 6a 20 cb ac  40 2b 0d 24 55 b3 9f 81   ....j .. @+.$U...
```

Frame (870 bytes) | Reassembled TCP (952 bytes)

No.: 16 · Time: 0.453666 · Source: 2a00:23c7:5b4d:5301:9d... Protocol: TLSv1.2 · Length: 870 · Info: Application Data

Help | Close

Protocol "layers" and reference models

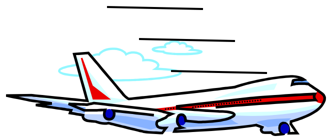
Networks are complex, with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question: is there any hope of *organizing* structure of network?

- and/or our *discussion* of networks?

Example: organization of air travel



end-to-end transfer of person plus baggage

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

How would you *define/discuss* the *system* of airline travel?

- a series of steps, involving many services

Example: organization of air travel

ticket (purchase)	*ticketing service*	ticket (complain)
baggage (check)	*baggage service*	baggage (claim)
gates (load)	*gate service*	gates (unload)
runway takeoff	*runway service*	runway landing
airplane routing	*routing service*	airplane routing

layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

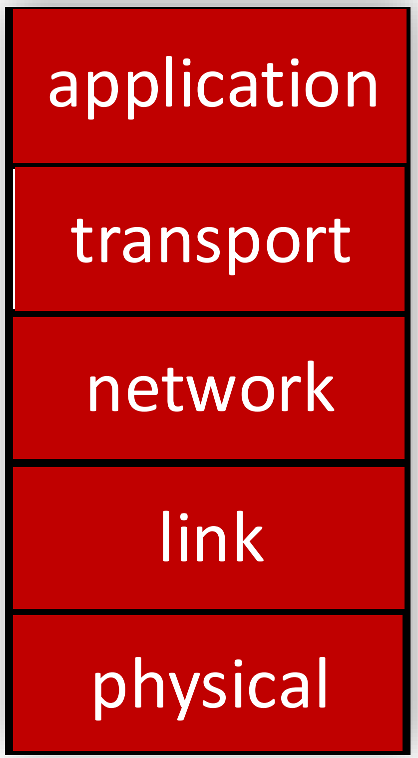
Lancaster University

Approach to designing/discussing complex systems:

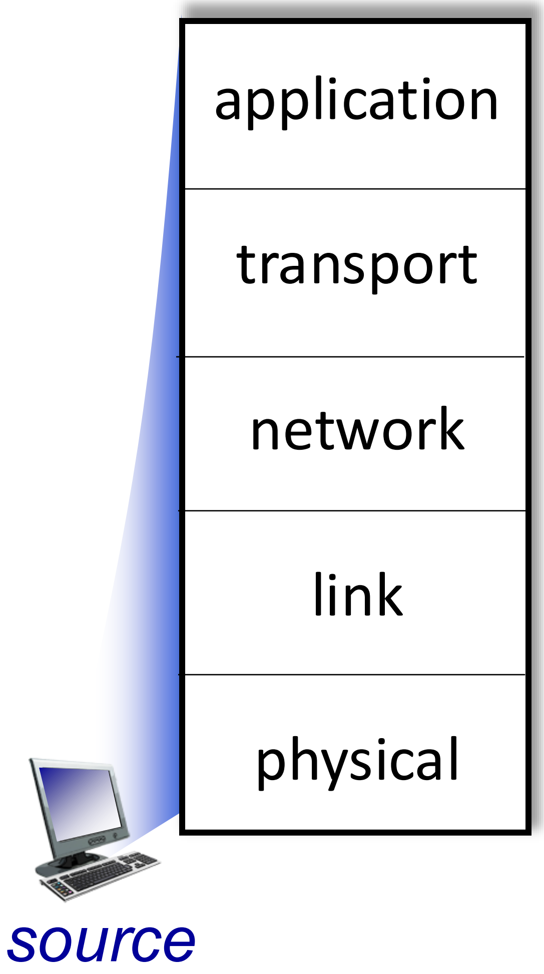
- explicit structure allows identification, relationship of system’s pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change in layer's service *implementation*: transparent to rest of system
  - e.g., change in gate procedure doesn’t affect rest of system

Layered Internet protocol stack

- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits “on the wire”



Services, Layering and Encapsulation



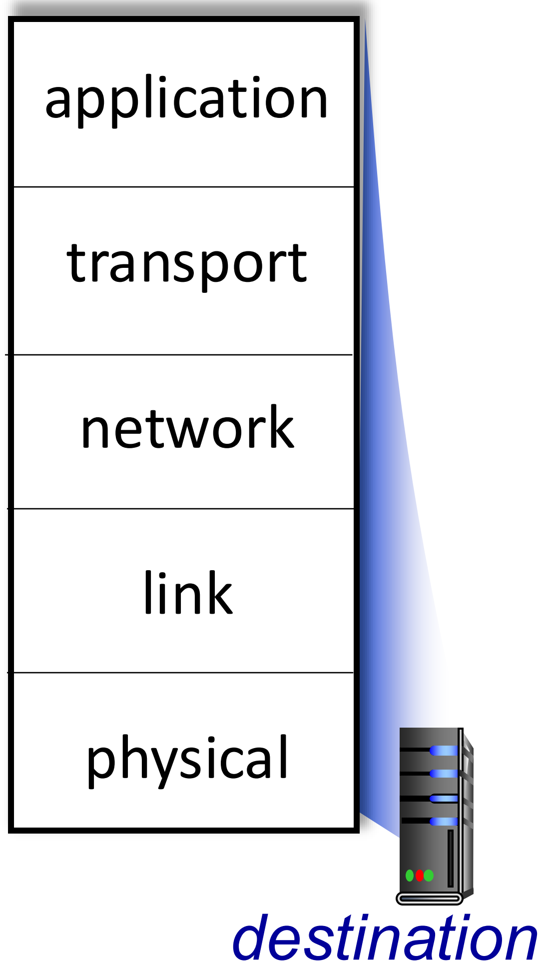
M

Application exchanges messages to implement some application service using *services* of transport layer

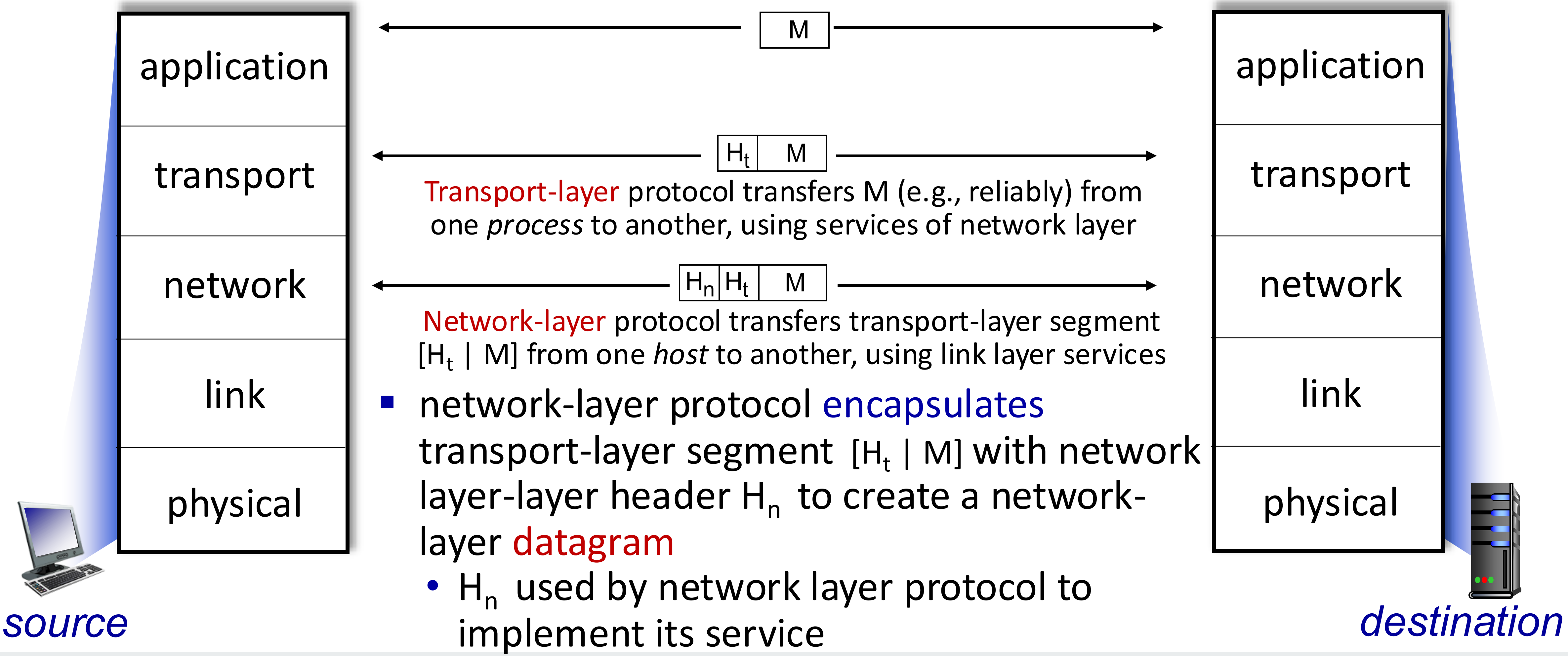
H_t M

Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

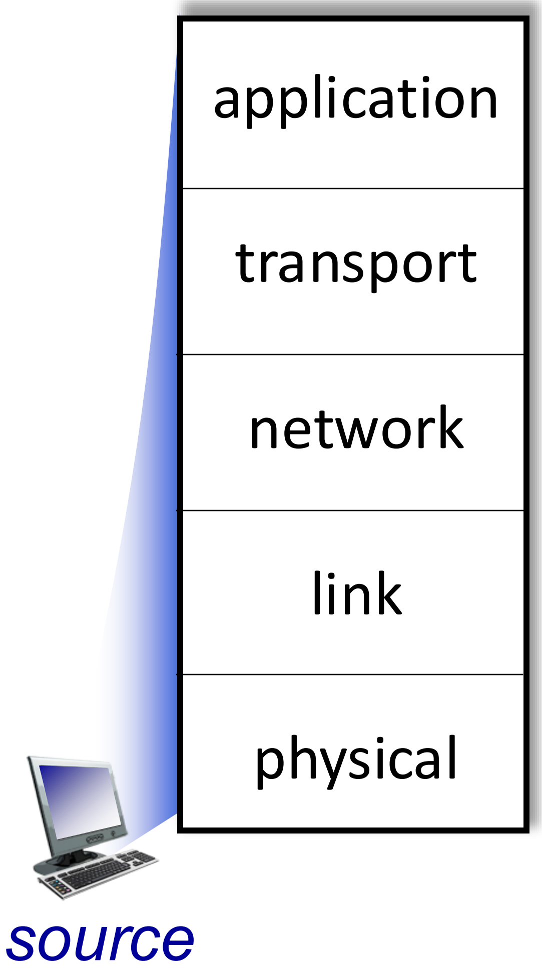


Services, Layering and Encapsulation



- network-layer protocol encapsulates transport-layer segment [H_t | M] with network layer-layer header H_n to create a network-layer datagram
  - H_n used by network layer protocol to implement its service

Services, Layering and Encapsulation



M

H_t | M

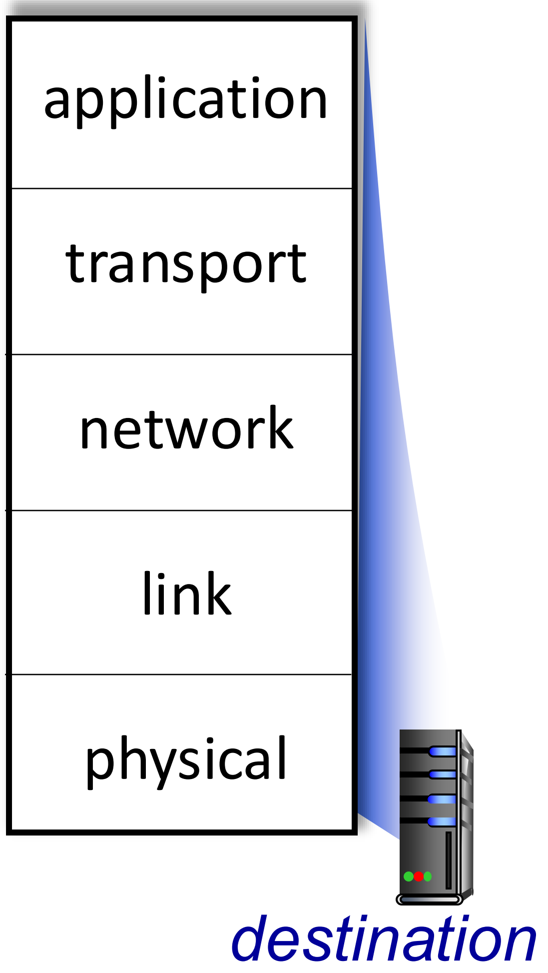
H_n | H_t | M

Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

H_l | H_n | H_t | M

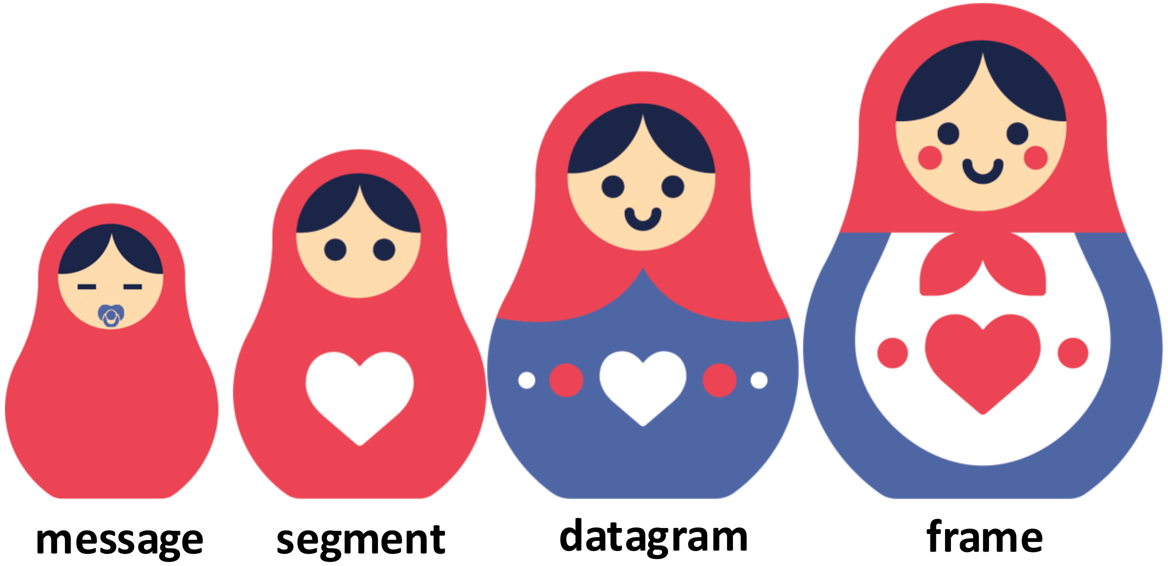
Link-layer protocol transfers datagram [H_n| [H_t |M] from *host* to neighboring host, using network-layer services

- link-layer protocol **encapsulates** network datagram [H_n| [H_t |M], with link-layer header H_l to create a link-layer **frame**



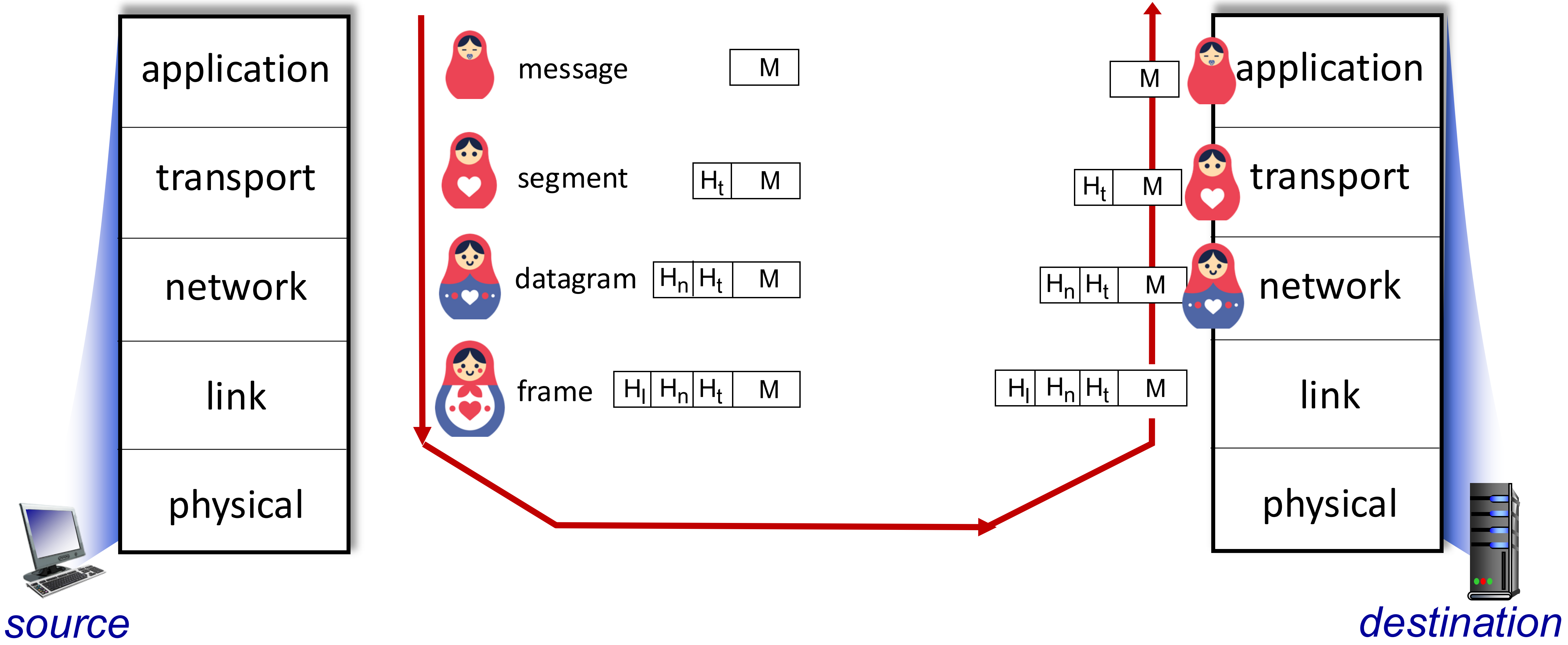
Encapsulation

Matryoshka dolls (stacking dolls)

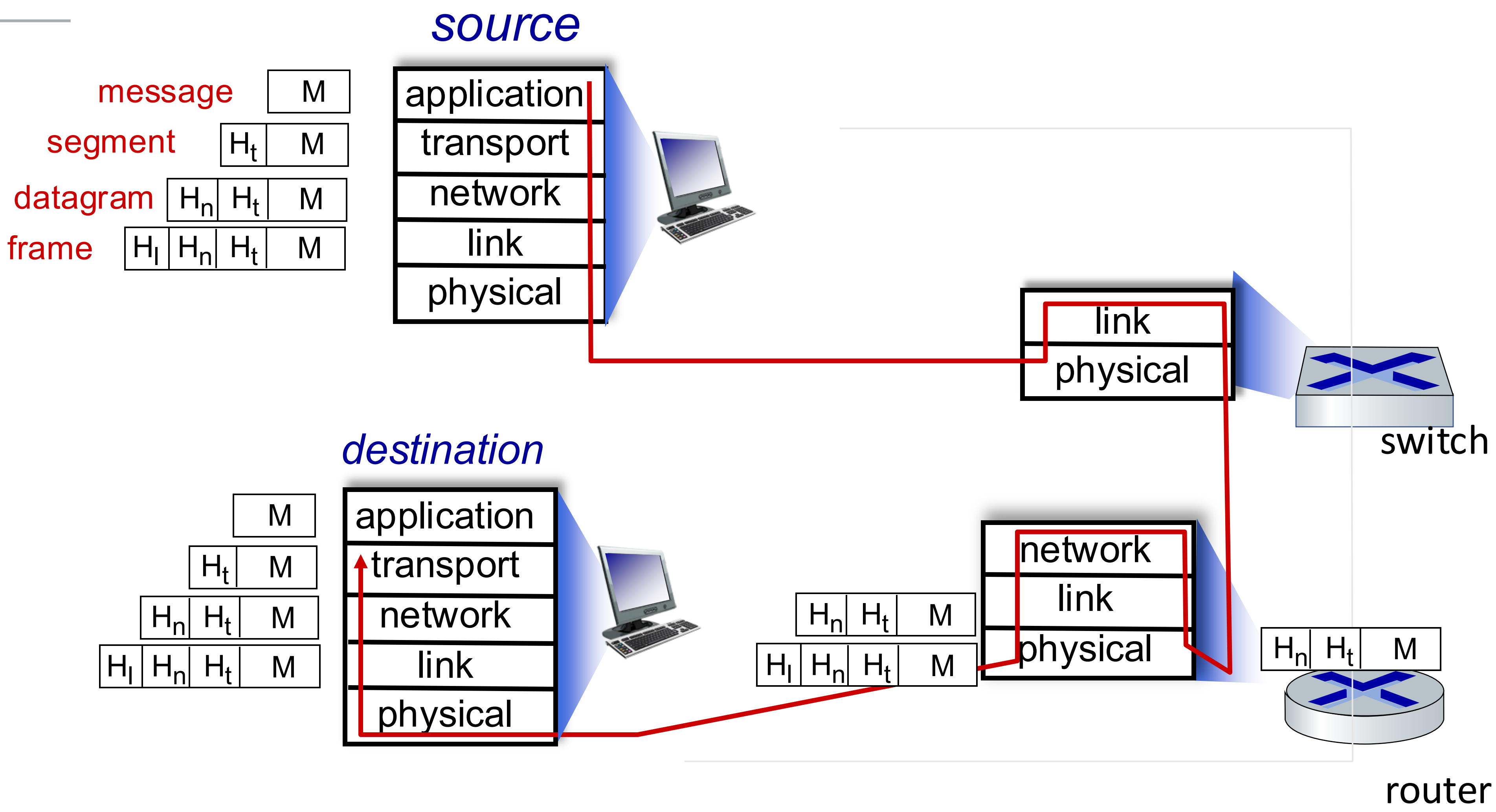


Credit: https://dribbble.com/shots/7182188-Babushka-Boi

Services, Layering and Encapsulation



Encapsulation: an end-end view



Network Experimentation

Introduction to Mininet

- As part of the 231 labs we will implement small versions of the Internet to understand how layers work using Mininet.
- **Mininet** is a **network emulator** that runs a **virtual network** — complete with hosts, switches, links, and controllers — **on a single Linux machine**.
  - Topologies described in Python
  - Interactive CLI to run commands on different hosts
  - All network layers in an IP system are replicated

Example: Time Server

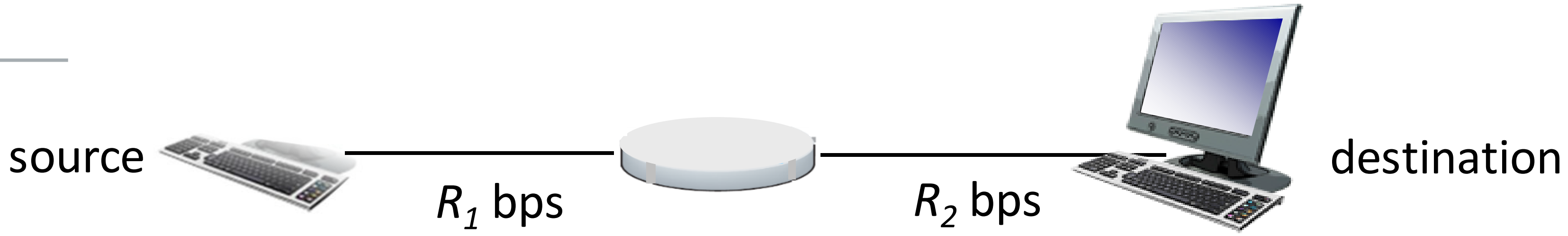
```
[3.650s][ scc_231/mininet][/workspace/Lecture6-IntroTo Layering]$ python3 ./topology.py
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding hosts:
h1 h2
*** Adding switches:
Warning: Linux bridge may not work with net.bridge.bridge-nf-call-arptables = 1
Warning: Linux bridge may not work with net.bridge.bridge-nf-call-iptables = 1
Warning: Linux bridge may not work with net.bridge.bridge-nf-call-ip6tables = 1
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1
*** Starting CLI:
mininet> h2 python3 time_server.py &
mininet> h1 curl http://10.0.0.2:8000/
Current server time: 2025-10-18 14:51:39
mininet>
```

```python
class TutorialTopology(Topo):

def build(self):
  # add two host to the network
  h1 = self.addHost('h1')
  h2 = self.addHost('h2')

  s1 = self.addSwitch('s1') # add a switch to the network


  # add a link between the hosts `h1` and `h2` and the
`s1` switch
  self.addLink(h1, s1)
  self.addLink(h2, s1)

def main():
  net = Mininet(topo=TutorialTopology(), controller=None,
switch=LinuxBridge)
  net.start()

  CLI(net)
  net.stop()
```

Conclusion

- The Internet is a very complex system.
  - Multiple access ISPs connecting billion of devices, capable to route packets across the world.
  - Packet-based mechanisms exploit statistical multiplexing for efficient resource use
  - Switching and Routing can simplify forwarding across multiple domains.
- Abstraction through Layering can simplify the design and operation of the Internet
  - Split network funcitonalities into layers.
- Next Topic: Socket programming

Revision Question (1)



1. Suppose there is exactly one packet switch between a sending host and a receiving host. The transmission rates between the sending host and the switch and between the switch and the receiving host are R_1 and R_2, respectively. Assuming that the switch uses store-and-forward packet switching, what is the total end-to-end delay to send a packet of length L ? (Ignore queuing, propagation delay, and processing delay.)

Kurose, James, and Keith Ross. *Computer Networking: a Top-Down Approach, Global Edition*, Pearson Education, Limited, 2021. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/lancaster/detail.action?docID=6644633.

Revision Question (2)

2. A user requires 100 kbps when active and is active 5% of the time. How many such users can share a 1 Mbps link using circuit switching?
3. Assume users run a telephony service only over this nework. What are the benefits and drawbacks when using circuit switching and packet forwarding to deliver the service? Consider aspects like user satisfaction and cost to compare the two approaches.