# 1. Introduction: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

Self study
Not on exam

# Quiz: Circuit Switching

Consider a circuit-switched network with N=100 users where each user is independently active with probability p=0.2 and when active, sends data at a rate of R=1Mbps. How much capacity must the network be provisioned with to guarantee service to all users?

A. 100 Mbps

B. 20 Mbps

C. 200 Mbps

D. 50 Mbps

E. 500 Mbps

**Answer: A**
**(probability of activity is irrelevant)**
**100 x 1 Mbps = 100 Mbps**

Open a browser and type: **www.pollev.com/salil**

# Quiz: Statistical Multiplexing

Consider a packet-switched network with N=100 users where each user is independently active with probability p=0.2 and when active, sends data at a rate of R=1Mbps. What is the expected aggregate traffic sent by all the users?

A. 100 Mbps

B. 20 Mbps

C. 200 Mbps

D. 50 Mbps

E. 500 Mbps

**Answer: B**
**100 x 0.2 x 1 Mbps = 20 Mbps**

Open a browser and type: **www.pollev.com/salil**

# Quiz: Delays

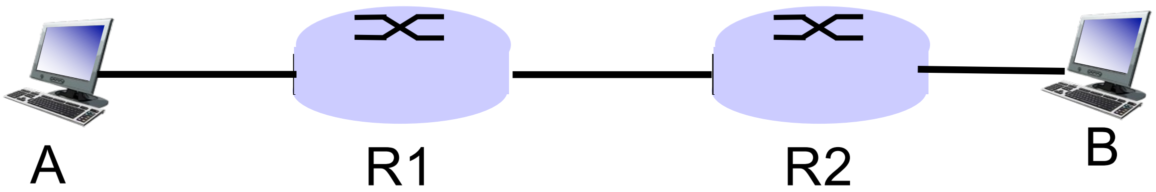
Consider a network connecting hosts A and B through two routers R1 and R2 like this: A-------R1----------R2---------B. Does whether a packet sent by A destined to B experiences queuing at R1 depend on the length of the link R1-R2?

A. Yes, it does

B. No, it doesn't

**Answer: B**

**Queuing happens because R1 cannot transmit packets on the R1-R2 link fast enough, i.e., the transmission rate of the R1-R2 link is lower than that of the link A-R1. The length of the R1-R2 link does not typically impact this.**




Open a browser and type: **www.pollev.com/salil**

# Three (networking) design steps

- Break down the problem into tasks
- Organize these tasks
- Decide who does what

# Tasks in Networking

- What does it take to send packets across?

- Prepare data (Application)
- Ensure that packets get to the destination process (Transport)
- Deliver packets across global network (Network)
- Delivery packets within local network to next hop (Datalink)
- Bits / Packets on wire (Physical)

This is decomposition…
Now, how do we organize these tasks?
Let us have an example

# Inspiration…

- CEO A writes letter to CEO B
  - Folds letter and gives it to Executive Assistant (EA)
    - » EA:
      - » Puts letter in envelope with CEO B's full name
      - » Takes to FedEx

```
Dear John,

Your days are numbered.

            --Grace
```

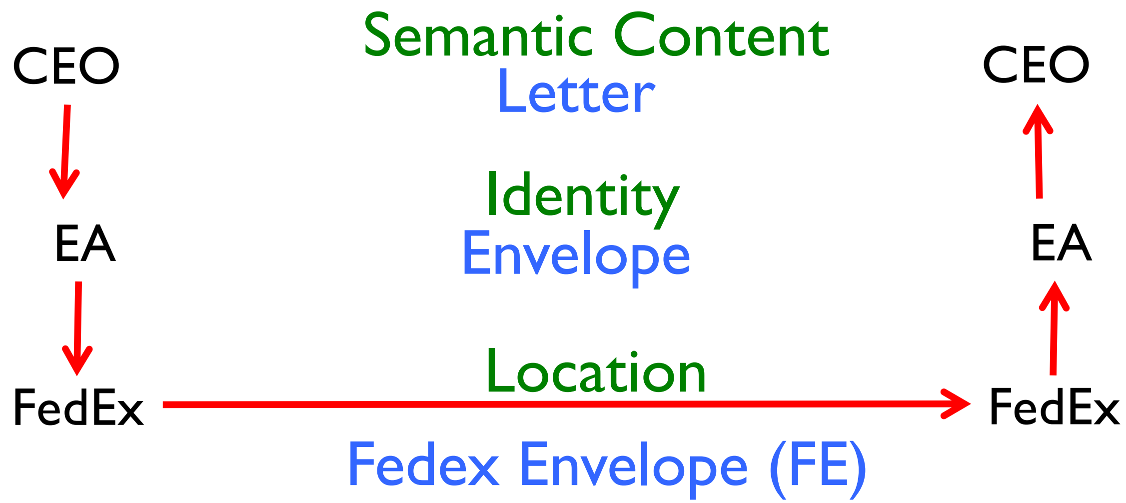
- FedEx Office
  - Puts letter in larger envelope
  - Puts name and street address on FedEx envelope
  - Puts package on FedEx delivery truck
- FedEx delivers to other company

# The Path of the Letter

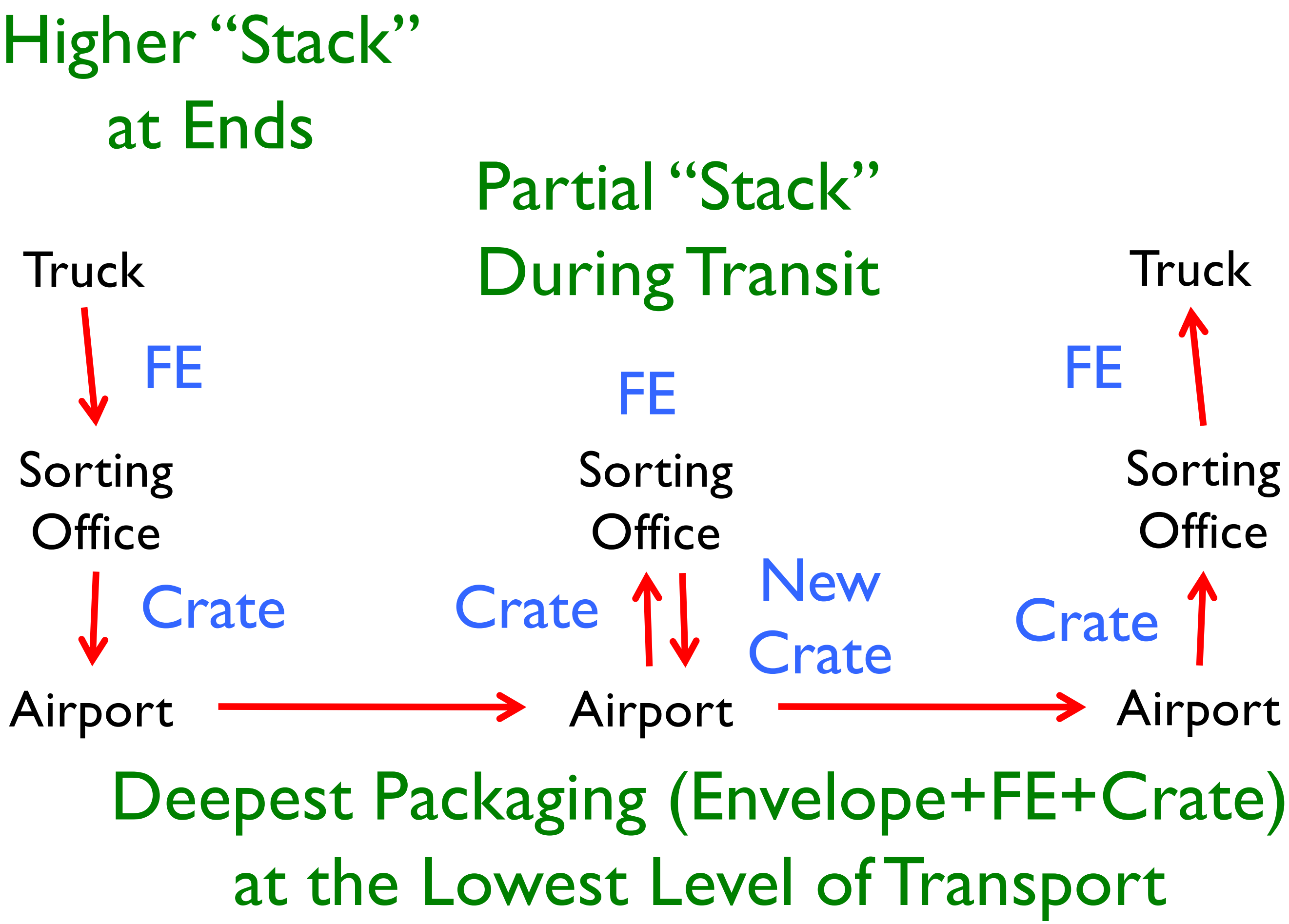
"Peers" on each side understand the same things

No one else needs to (abstraction)

Lowest level has most packaging



# The Path Through FedEx



# In the context of the Internet

Applications

…built on…

Reliable (or unreliable) transport

…built on…

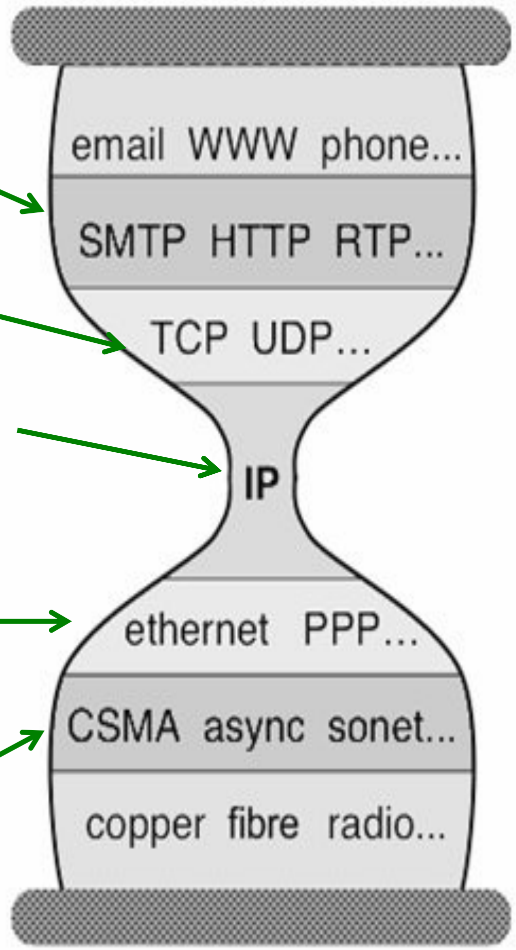
Best-effort global packet delivery

…built on…

Best-effort local packet delivery

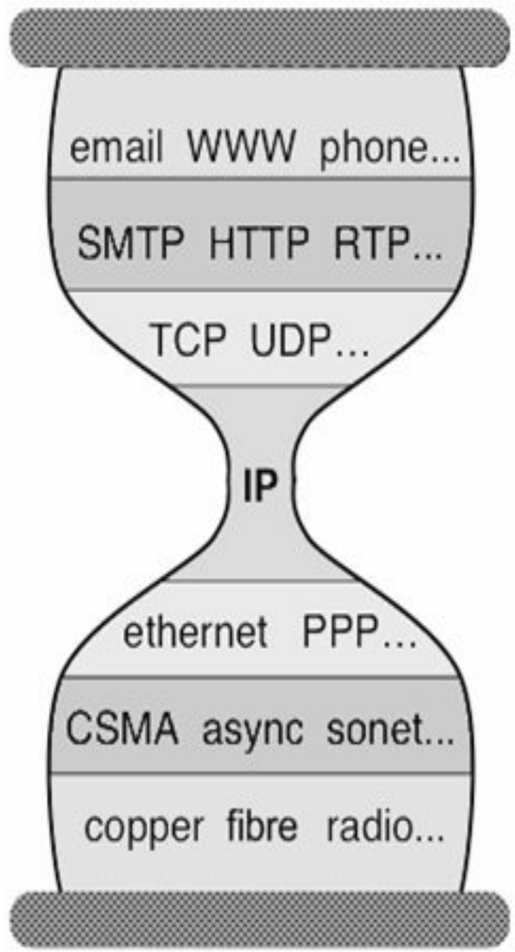
…built on…

Physical transfer of bits



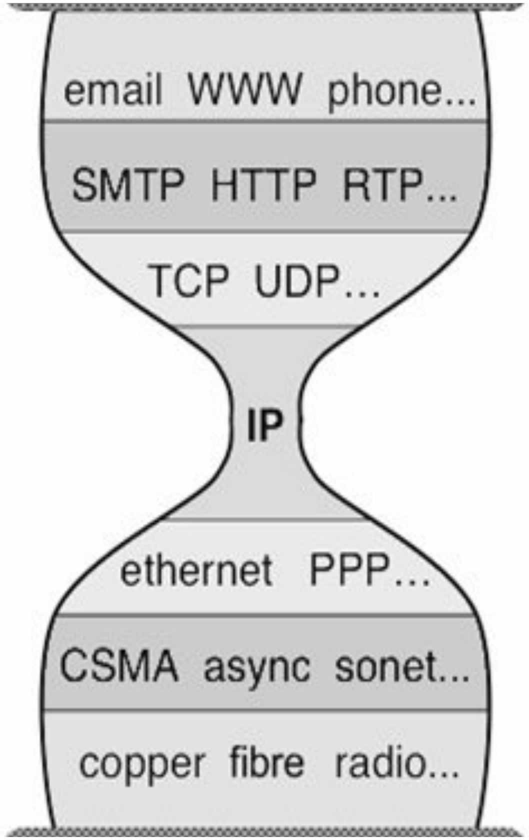
# Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP, Skype, ..
- *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"

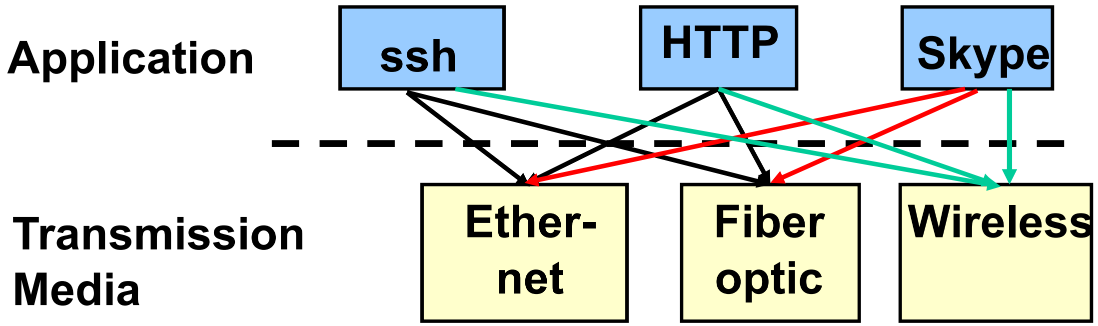


# Three Observations

- Each layer:
  - Depends on layer below
  - Supports layer above
  - Independent of others
- Multiple versions in layer
  - Interfaces differ somewhat
  - Components pick which lower-level protocol to use
- But only one IP layer
  - Unifying protocol



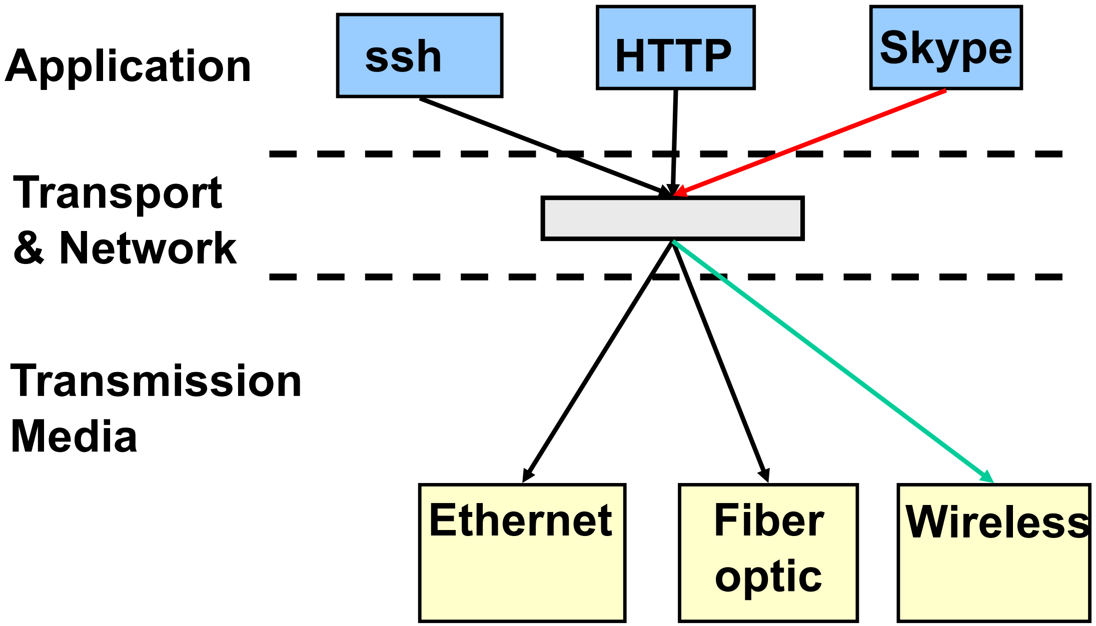
# An Example: No Layering




- No layering: each new application has to be re-implemented for every network technology !

# An Example: Benefit of Layering

- Introducing an intermediate layer provides a common abstraction for various network technologies



# Is Layering Harmful?

- Layer N may duplicate lower-level functionality
  - E.g., error recovery to retransmit lost data
- Information hiding may hurt performance
  - E.g., packet loss due to corruption vs. congestion
- Headers start to get large
  - E.g., typically, TCP + IP + Ethernet headers add up to 54 bytes
- Layer violations when the gains too great to resist
  - E.g., Network Address Translation (NAT – to be covered in Network Layer)
- Layer violations when network doesn’t trust ends
  - E.g., Firewalls (Security)

# Distributing Layers Across Network

- Layers are simple if only on a single machine
  - Just stack of modules interacting with those above/below
- But we need to implement layers across machines
  - Hosts
  - Routers
  - Switches
- What gets implemented where?

# What Gets Implemented on Host?

- Hosts have applications that generate data/messages that are eventually put out on wire

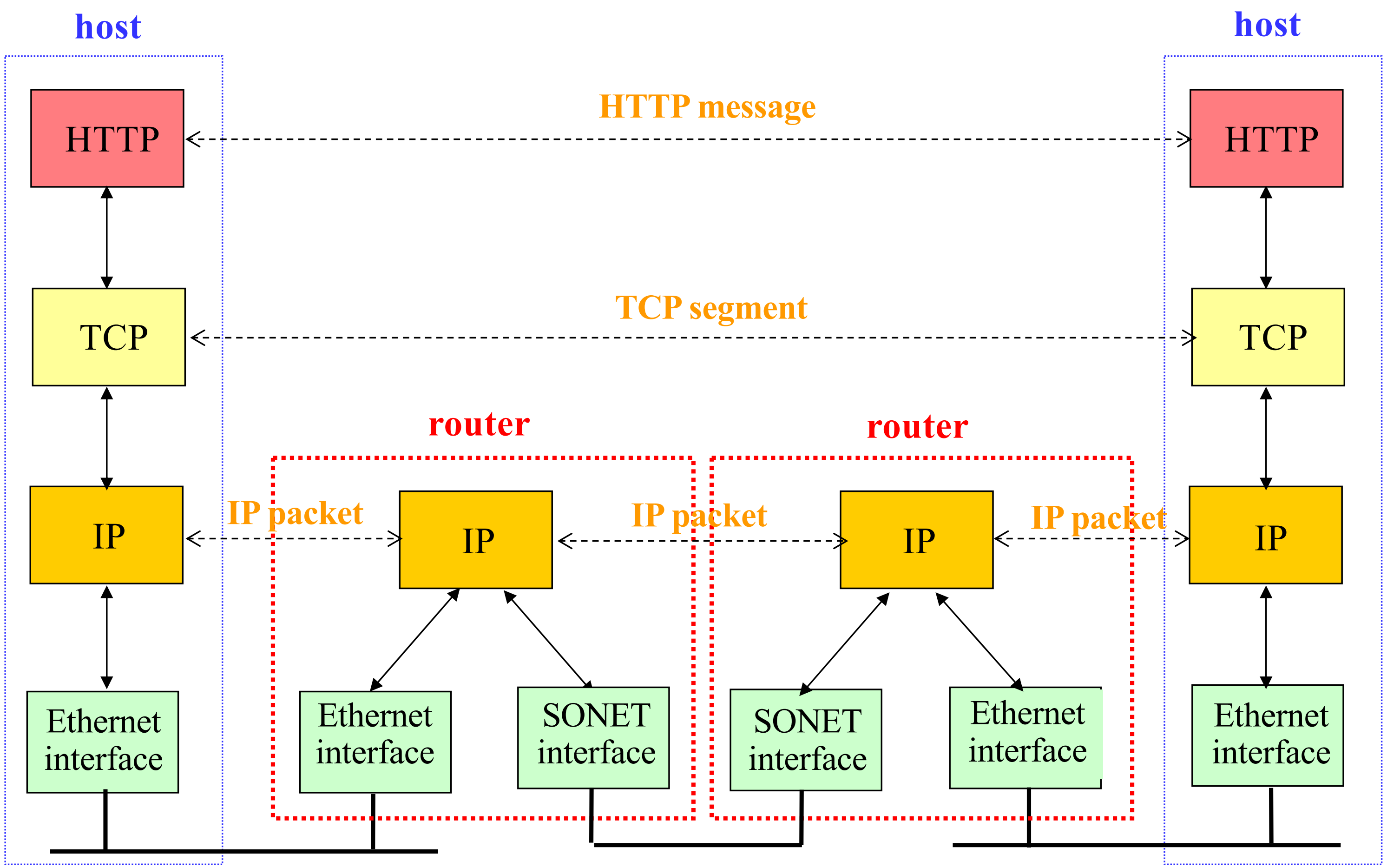
- At receiver host bits arrive on wire, must make it up to application

- Therefore, all layers must exist at host!

# What Gets Implemented on Router?

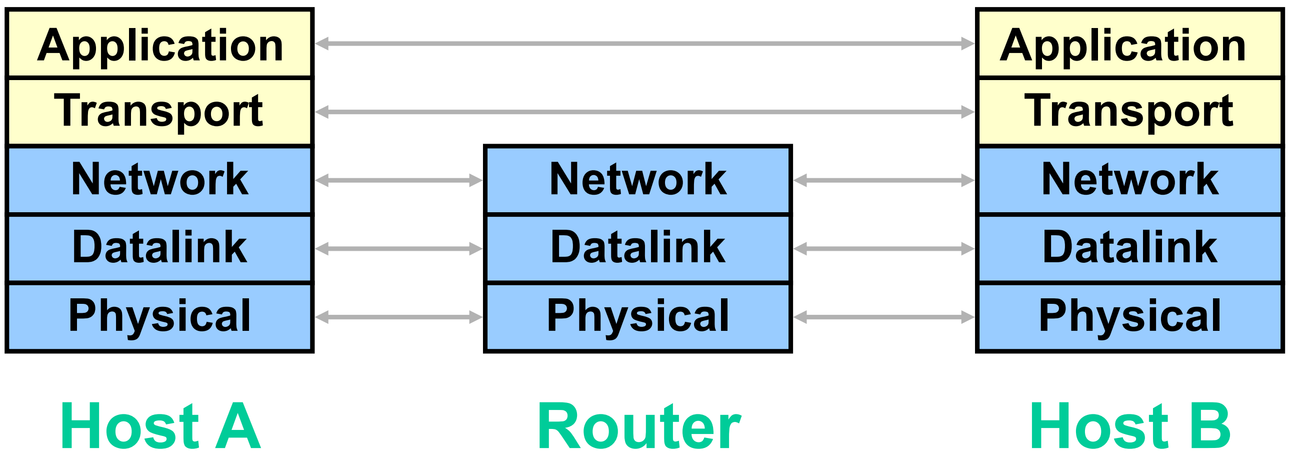
- Bits arrive on wire
  - Physical layer necessary
- Packets must be delivered to next-hop
  - datalink layer necessary
- Routers participate in global delivery
  - Network layer necessary
- Routers don't support reliable delivery
  - Transport layer (and above) ***not*** supported

# Internet Layered Architecture



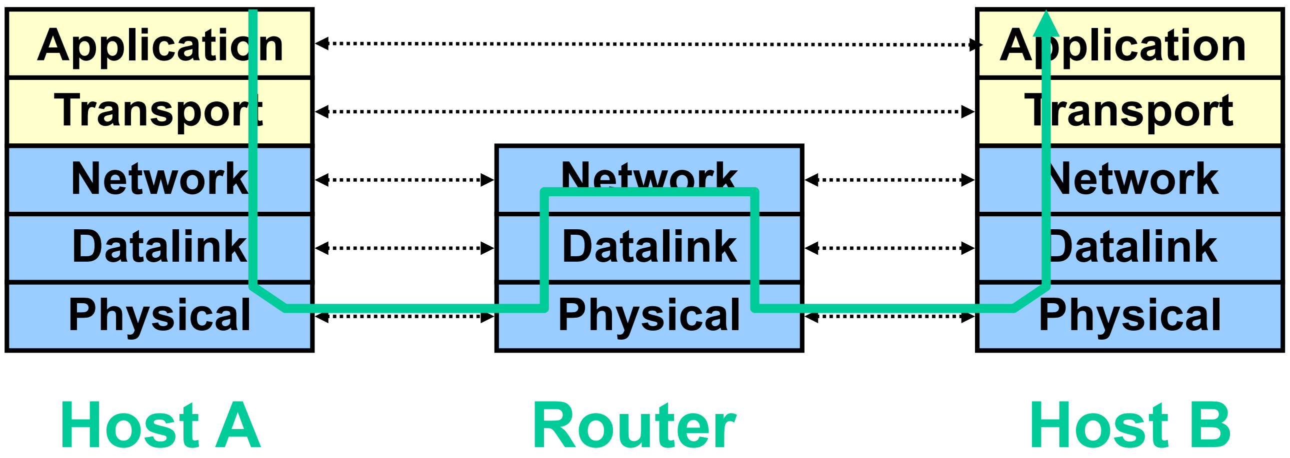
# Logical Communication

- Layers interacts with peer's corresponding layer

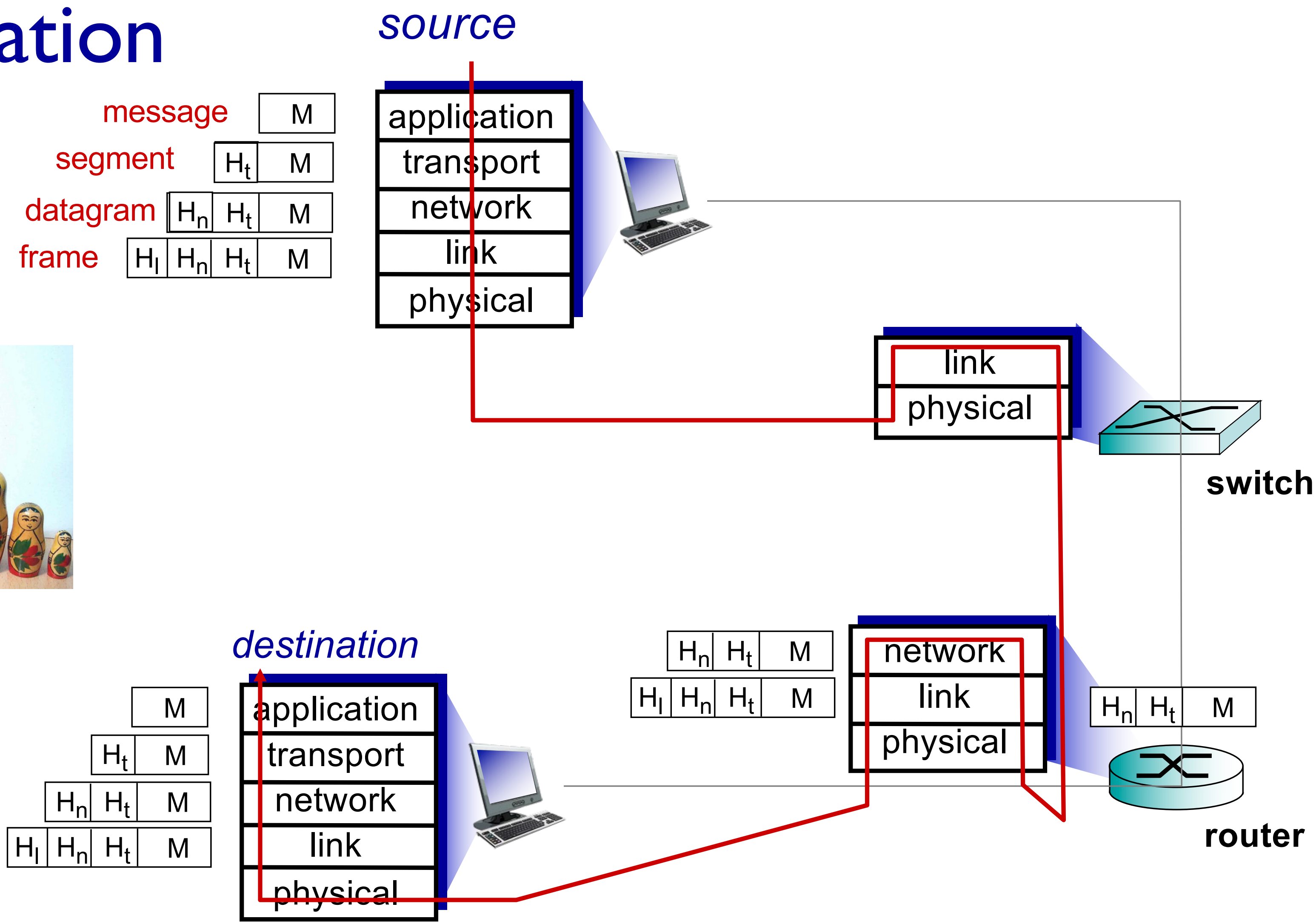


# Physical Communication

- Communication goes down to physical network
- Then from network peer to peer
- Then up to relevant layer



# Encapsulation



# Quiz: Layering

What are two benefits of using a layered network model ? (Choose two)

A. It makes it easy to introduce new protocols

B. It speeds up packet delivery

C. It allows us to have many different packet headers

D. It prevents technology in one layer from affecting other layers

E. It creates many acronyms

F. It reminds me of cake

**Answer: A+D**
**C and E are true statements, but side-effects not benefits,**
**yum I also love cake**

Open a browser and type: **www.pollev.com/salil**

# Introduction: summary

*We've covered a "ton" of material!*

- Internet overview
- what's a protocol?
- network edge, access network, core
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

*You now have:*

- context, overview, vocabulary, "feel" of networking
- more depth, detail, *and fun* to follow*!*