

# I. Introduction: roadmap

I.1 what is the Internet?

I.2 network edge

- end systems, access networks, links

I.3 network core

- packet switching, circuit switching, network structure

I.4 delay, loss, throughput in networks

I.5 protocol layers, service models

I.6 networks under attack: security

I.7 history



Self study



## 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  $R=1\text{Mbps}$ . How much capacity must the network be provisioned with to guarantee service to all users?

- A. 100 Mbps it is circuit switched so each of the 100 people need a dedicated 1mbps connection whether they are active or not
- B. 20 Mbps
- C. 200 Mbps
- D. 50 Mbps
- E. 500 Mbps



## 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  $R=1\text{Mbps}$ . **What is the expected aggregate traffic sent by the users?**

A. 100 Mbps

B. 20 Mbps we have an average of 20% of the 100 users (20) active at any given time. each user uses 1 mbps so 20mbps should be expected at any given time

C. 200 Mbps

D. 50 Mbps

E. 500 Mbps



## 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.



the LENGTH of the link does not matter (that is propagation delay), it is the bandwidth of it (i.e. how fast R1 can push packets out onto the link so it can start the next one - transmission delay)

i.e. if the link has a bandwidth of 1 bit/sec then it will take packet length seconds to get it out onto the link. If it has 2 bits/sec then the time halves. If it has packetsize/sec then it does it in 1 sec. If incoming packet rate exceeds this rate then queuing occurs

# Three (networking) design steps

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

# Tasks in Networking

- ❖ What does it takes to send packets across?
- ❖ Prepare data (Application)
- ❖ Ensure that packets get to the dst 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 hands it to administrative aide

Dear John,

» Aide:

Your days are numbered.

» Puts letter in envelope with CEO  
B's full name

» Takes to FedEx

--Pat

- ❖ FedEx Office

- Puts letter in larger envelope
- Puts name and street address on FedEx envelope
- Puts package on FedEx delivery truck

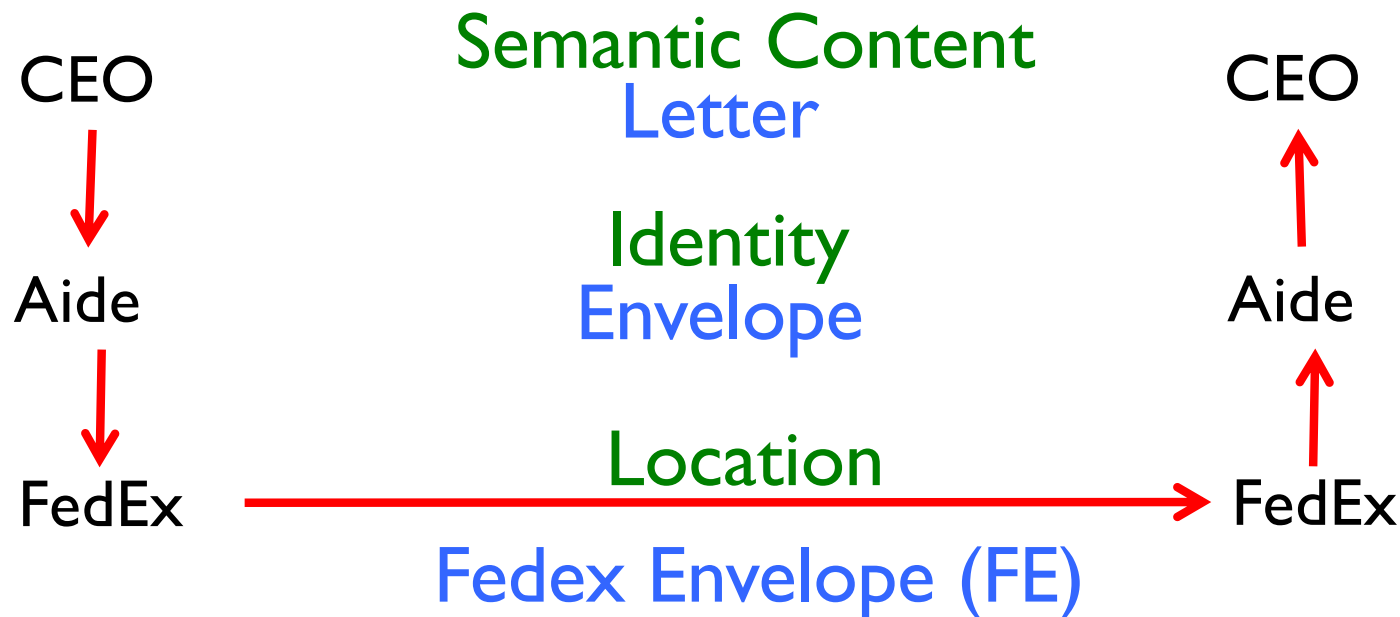
- ❖ FedEx delivers to other company

# The Path of the Letter

“Peers” on each side understand the same things

No one else needs to (abstraction)

Lowest level has most packaging

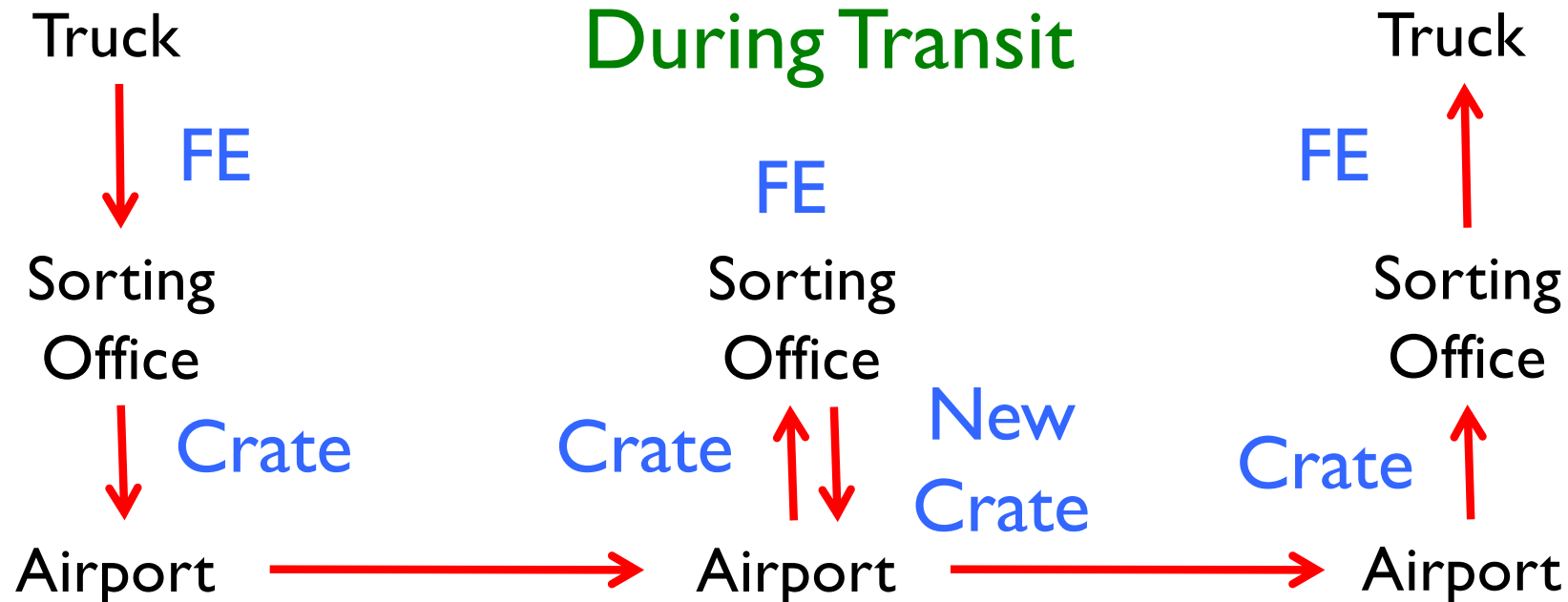




# The Path Through FedEx

Higher “Stack”  
at Ends

Partial “Stack”  
During Transit



Deepest Packaging (Envelope+FE+Crate)  
at the Lowest Level of Transport

# In the context of the Internet

**Applications**

...built on...

TCP or UDP transport protocols

**Reliable (or unreliable) transport**

...built on...

best-effort packet switching of the network layer



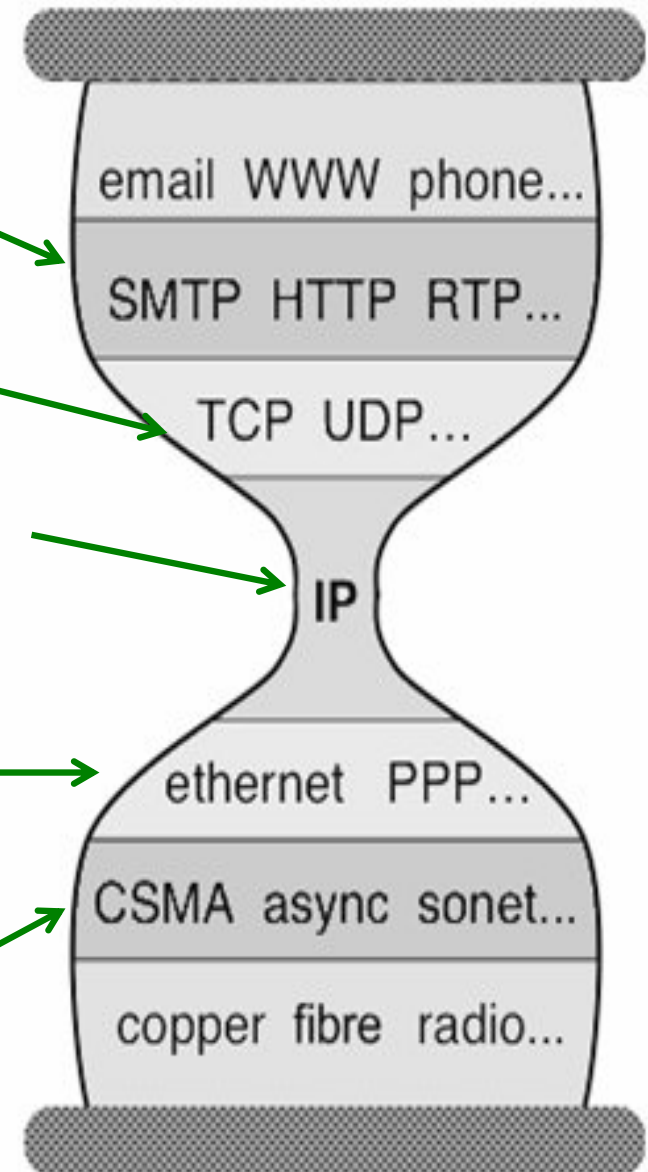
**Best-effort global packet delivery**

...built on...

**Best-effort local packet delivery**

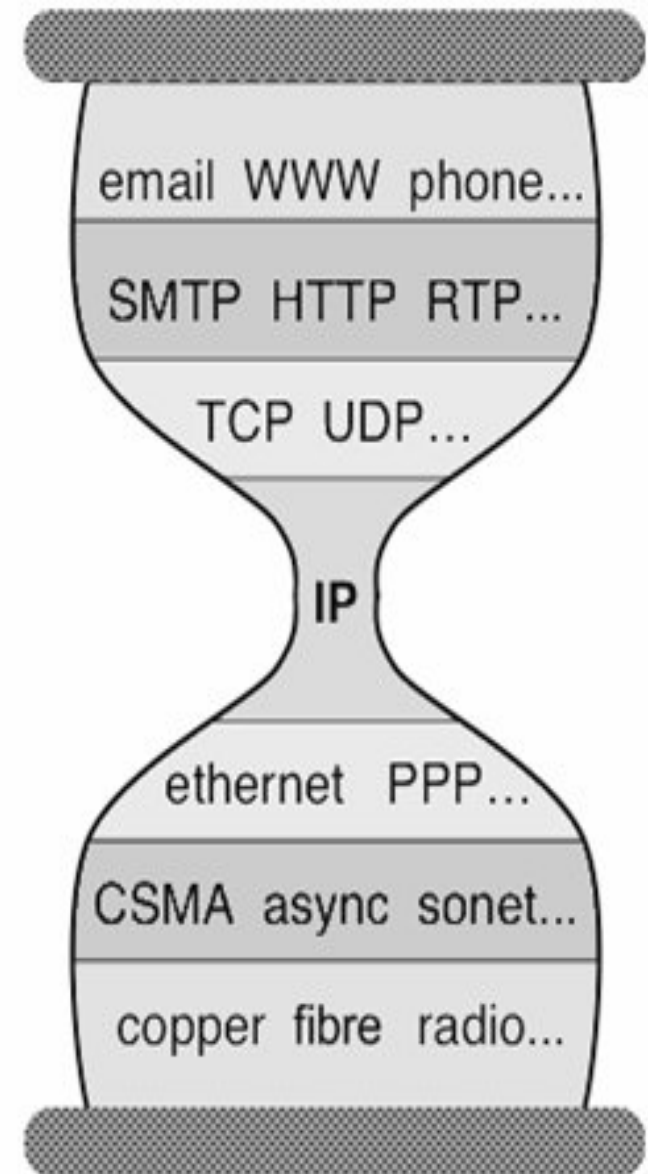
...built on...

**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, 802.111 (WiFi), PPP
- ❖ *physical*: bits “on the wire”



# Three Observations

## ❖ Each layer:

- Depends on layer below
- Supports layer above
- Independent of others

i.e. other than your parent and child, no one else needs to know about you. You support your parent and depend on your child (kind of backwards to real life lol)

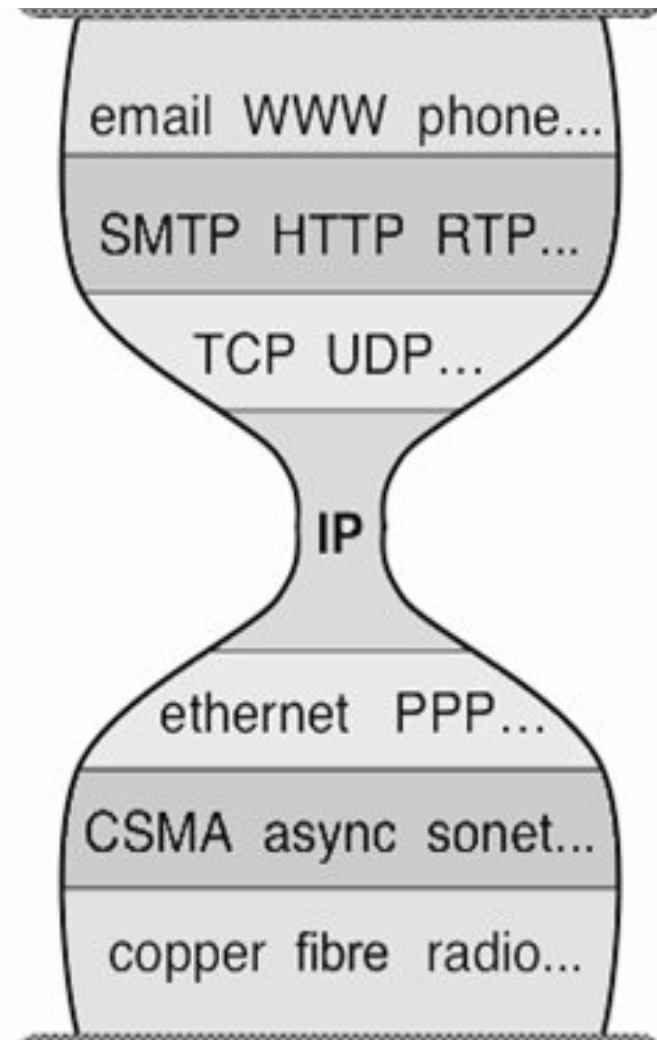
## ❖ Multiple versions in layer

- Interfaces differ somewhat
- Components pick which lower-level protocol to use

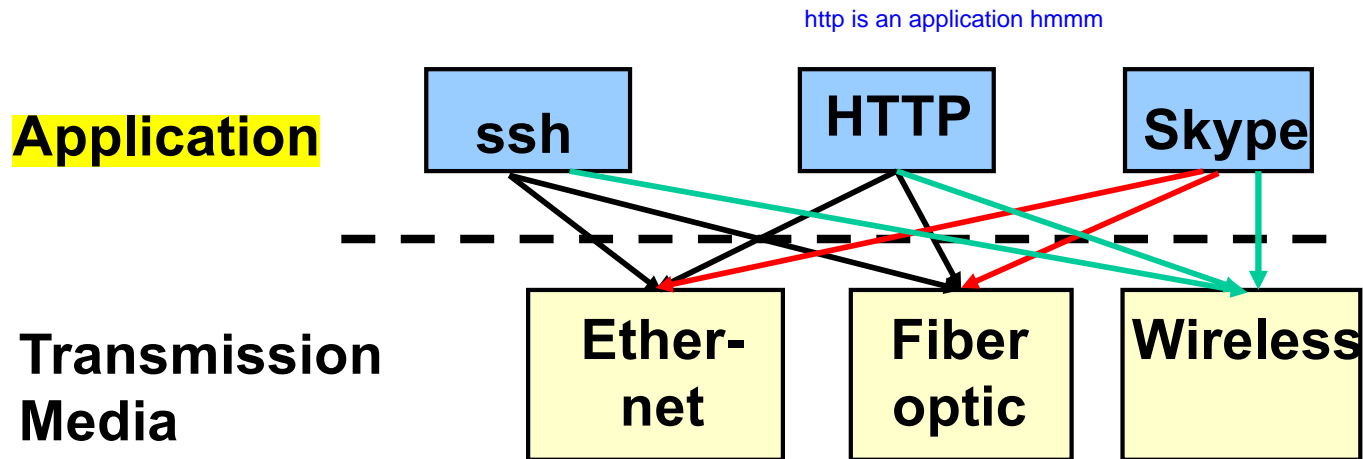
i.e. UDP vs TCP

## ❖ But only one IP layer

- Unifying protocol



# An Example: No Layering

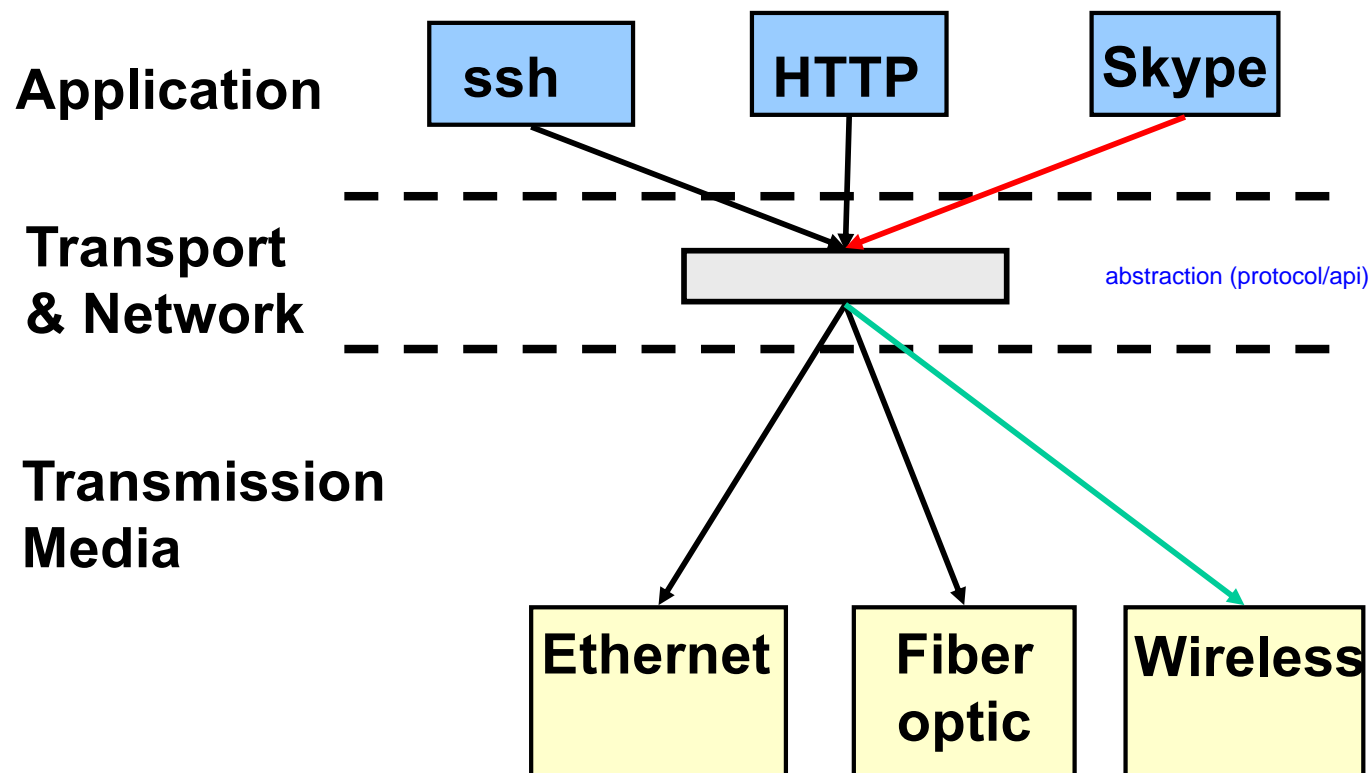


- ❖ No layering: each new application has to be re-implemented for every network technology !

write the ethernet version of the app, the fibre version, the wireless version  
OR...

# 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 really big
  - E.g., typically TCP + IP + Ethernet headers add up to 54 bytes
- ❖ Layer violations when the gains too great to resist
  - E.g., NAT
- ❖ Layer violations when network doesn't trust ends
  - E.g., Firewalls

# 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?

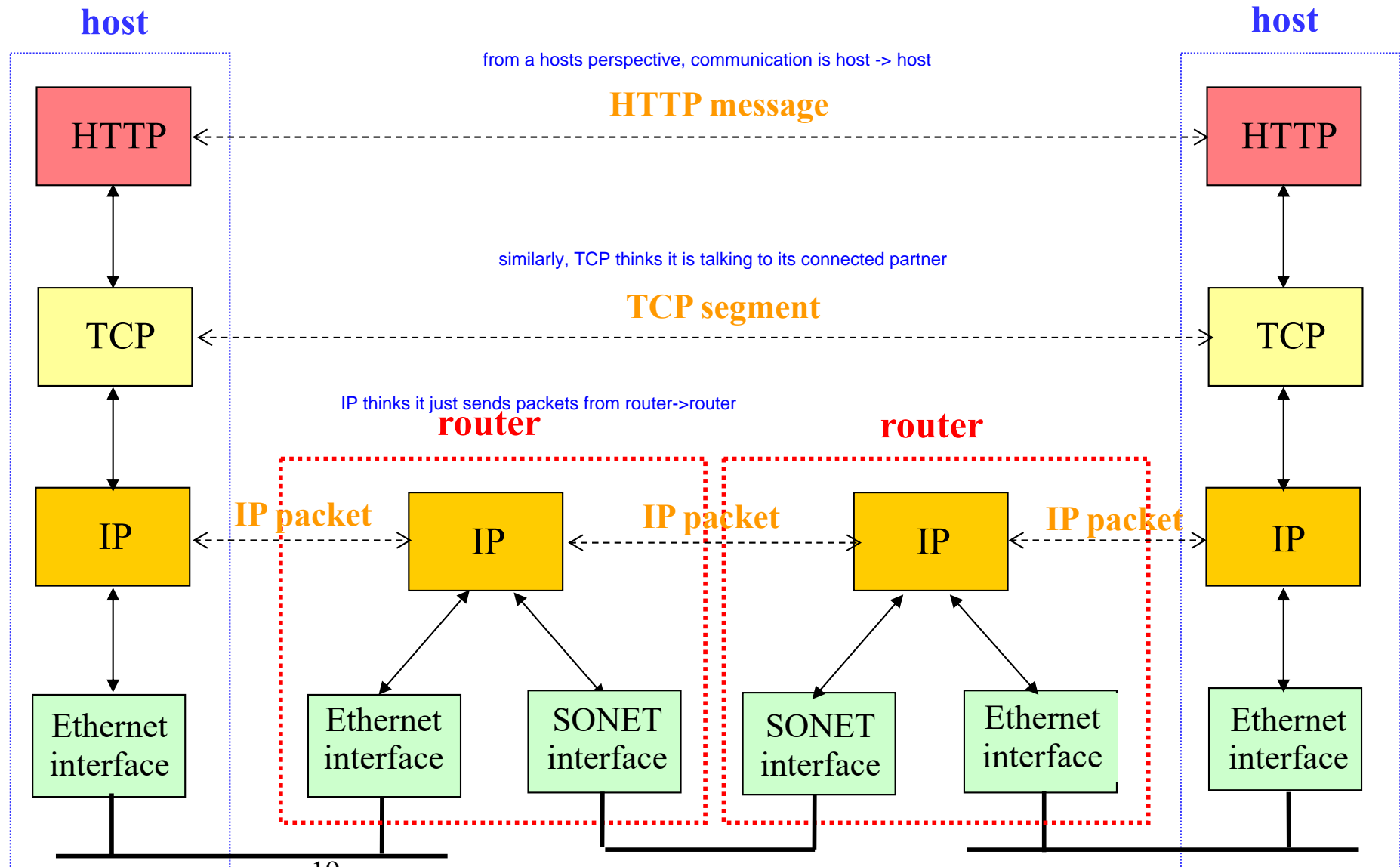
- ❖ Hosts have applications that generate data/messages that are eventually put out on wire
- ❖ At receiver host bits arrive on wire, must make it up to application
- ❖ Therefore, all layers must exist at host!

# 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

hence don't need transport layer and if you don't need transport layer then application layer is also out of the question

# Internet Layered Architecture

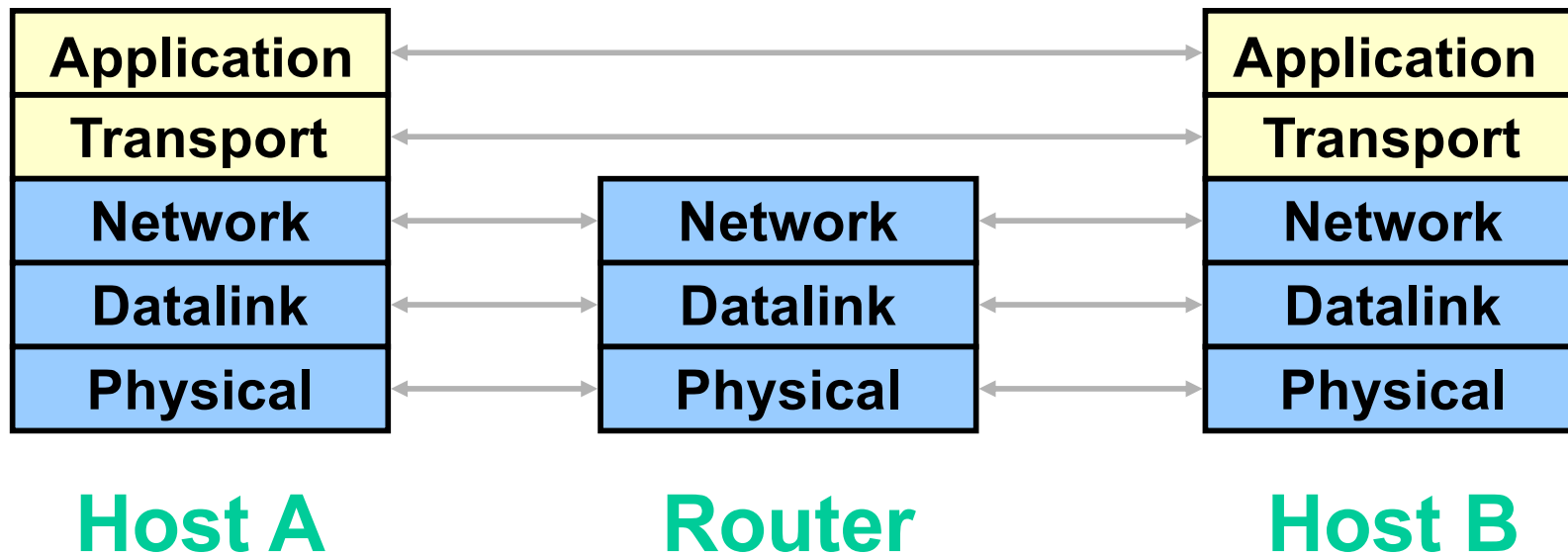


but actually ethernet needs to talk to ethernet and likewise for SONET (i.e. the implentation details are hidden)

# Logical Communication

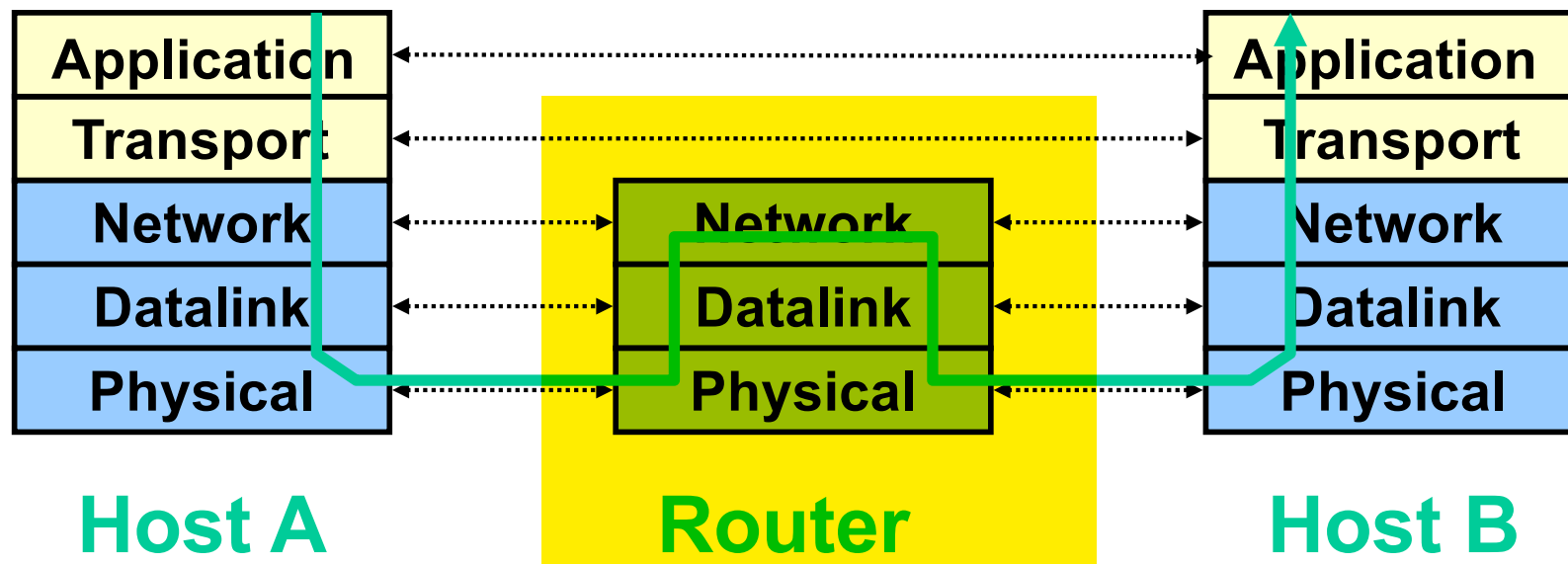
- ❖ Layers interacts with peer's corresponding layer

this is what i was communicating above!

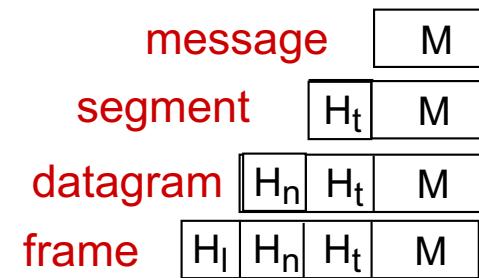


# Physical Communication

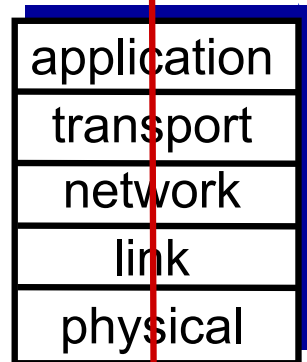
- ❖ Communication goes down to physical network
- ❖ Then from network peer to peer
- ❖ Then up to relevant layer



# Encapsulation

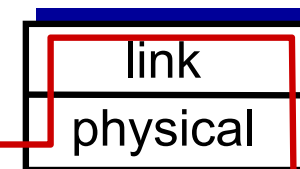


*source*



go down to physical

feed up to link and then back out to physical

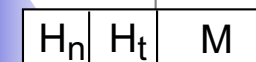
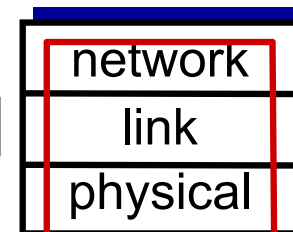
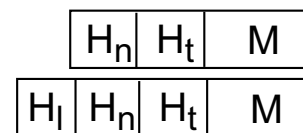
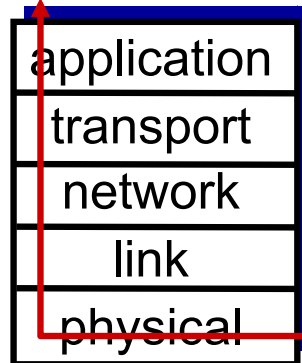
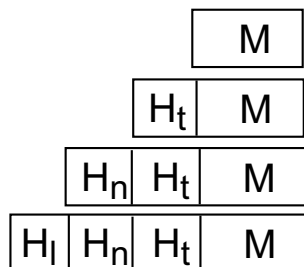


now physical talks to physical



**switch**

*destination*



**router**

physical talking to physical

## 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

# I. Introduction: roadmap

I.1 what is the Internet?

I.2 network edge

- end systems, access networks, links

I.3 network core

- packet switching, circuit switching, network structure

I.4 delay, loss, throughput in networks

I.5 protocol layers, service models

I.6 networks under attack: security

I.7 history



Self study