1. Introduction: roadmap

Self study

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
    - » Aide:
      - » Puts letter in envelope with CEO B's full name
      - » Takes to FedEx

```
Dear John,

Your days are numbered.

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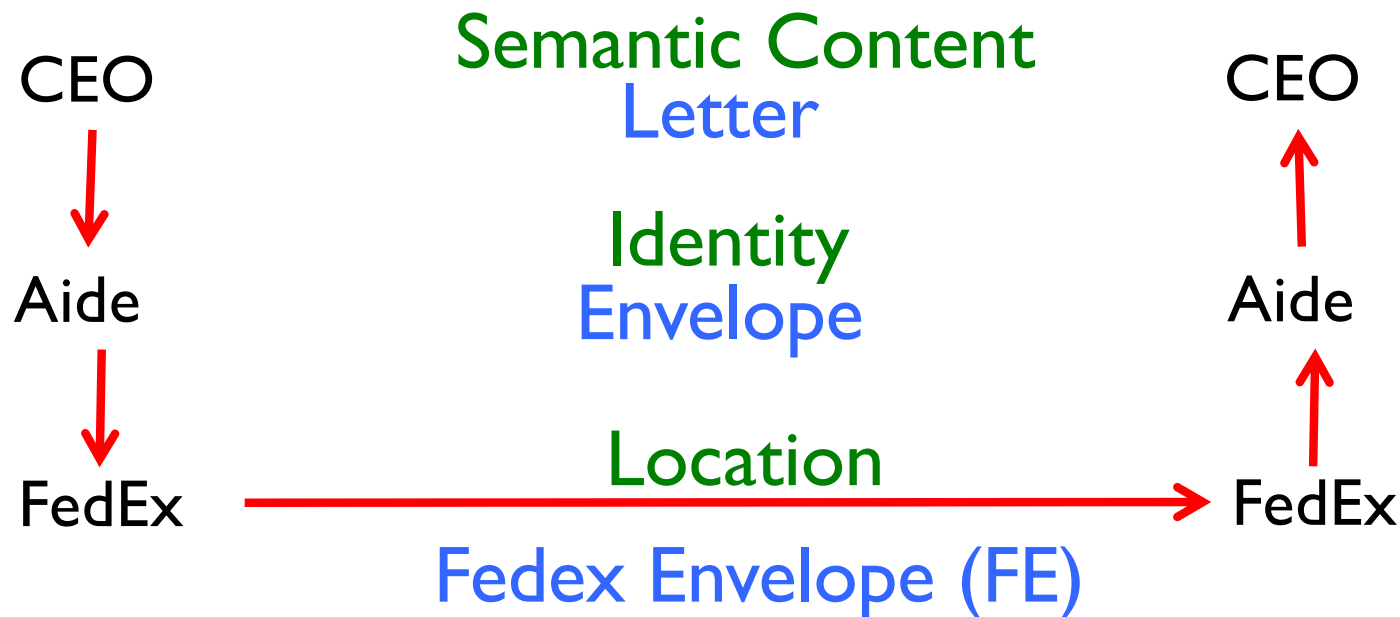
- ❖ FedEx Office
  - ▪ Puts letter in larger envelope
  - ▪ Puts name and street address on FedEx envelope
  - ▪ Puts package on FedEx delivery truck
- ❖ FedEx delivers to other company

The Path of the Letter

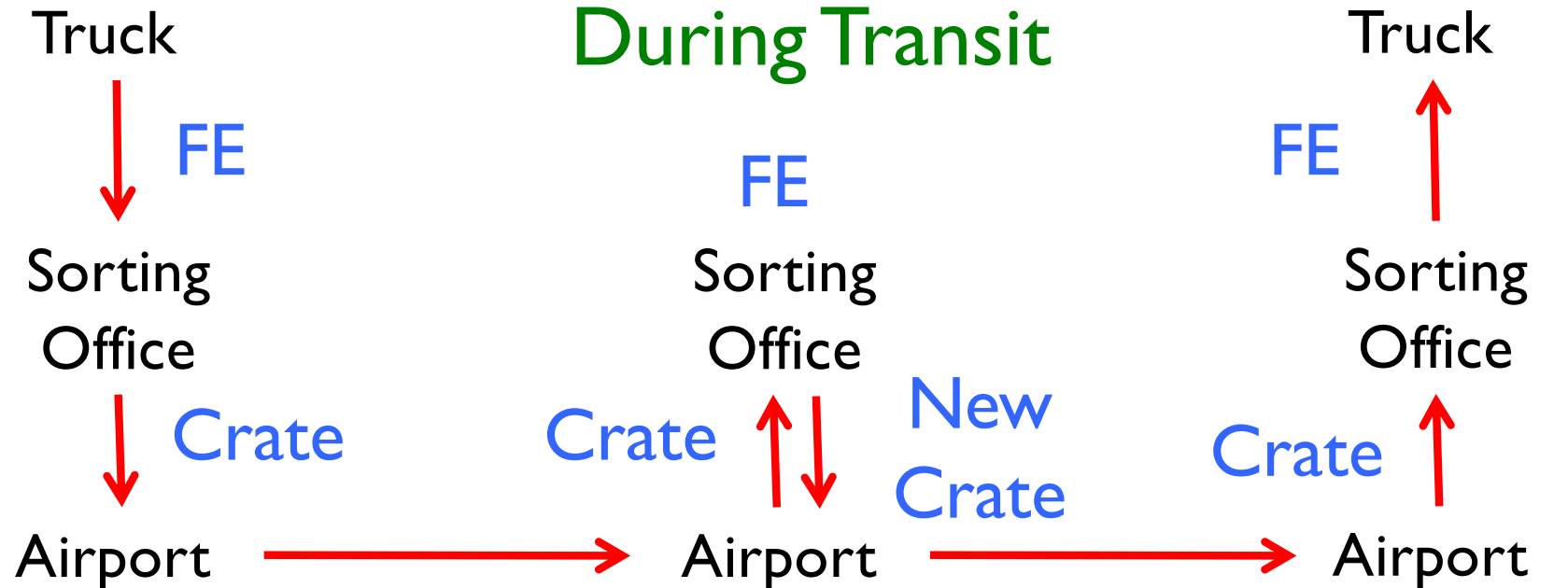"Peers" on each side understand the same things
No one else needs to (abstraction)
Lowest level has most packaging

CEO	Semantic Content / Letter	CEO
Aide	Identity / Envelope	Aide
FedEx	Location / Fedex Envelope (FE)	FedEx

The Path Through FedEx

Higher "Stack"
at Ends

Partial "Stack"
During Transit

Truck

Truck

FE

FE

FE

Sorting
Office

Sorting
Office

Sorting
Office

Crate

Crate

New
Crate

Crate

Airport

Airport

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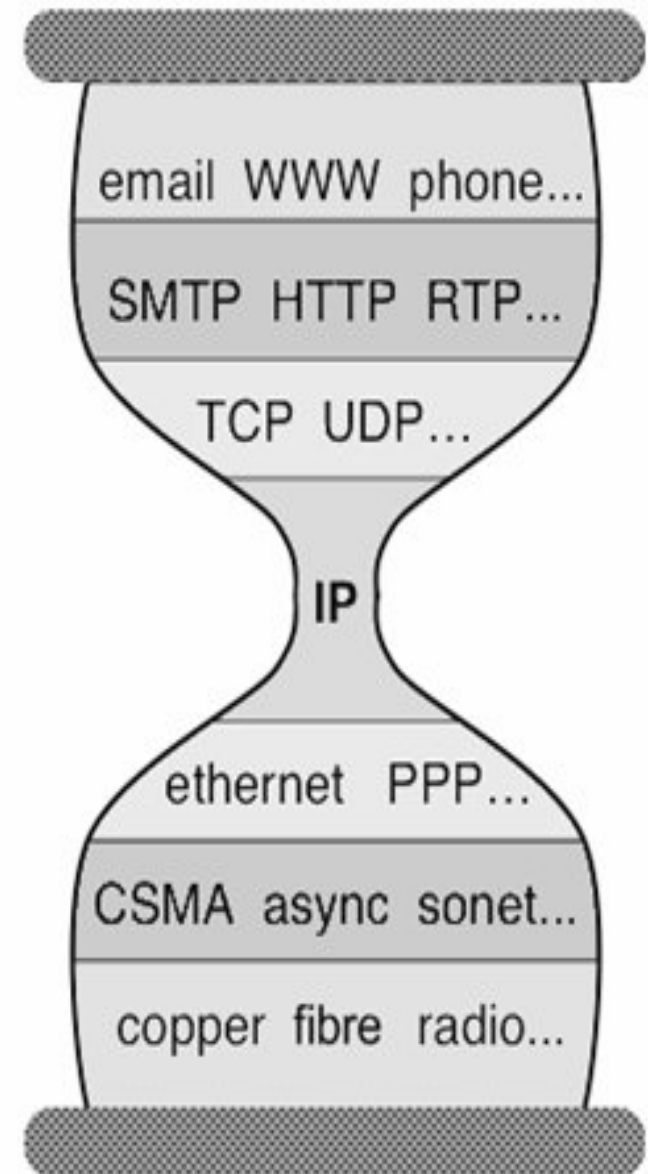
Deepest Packaging (Envelope+FE+Crate)
at the Lowest Level of Transport

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

email WWW phone...

SMTP HTTP RTP...

TCP UDP...

IP

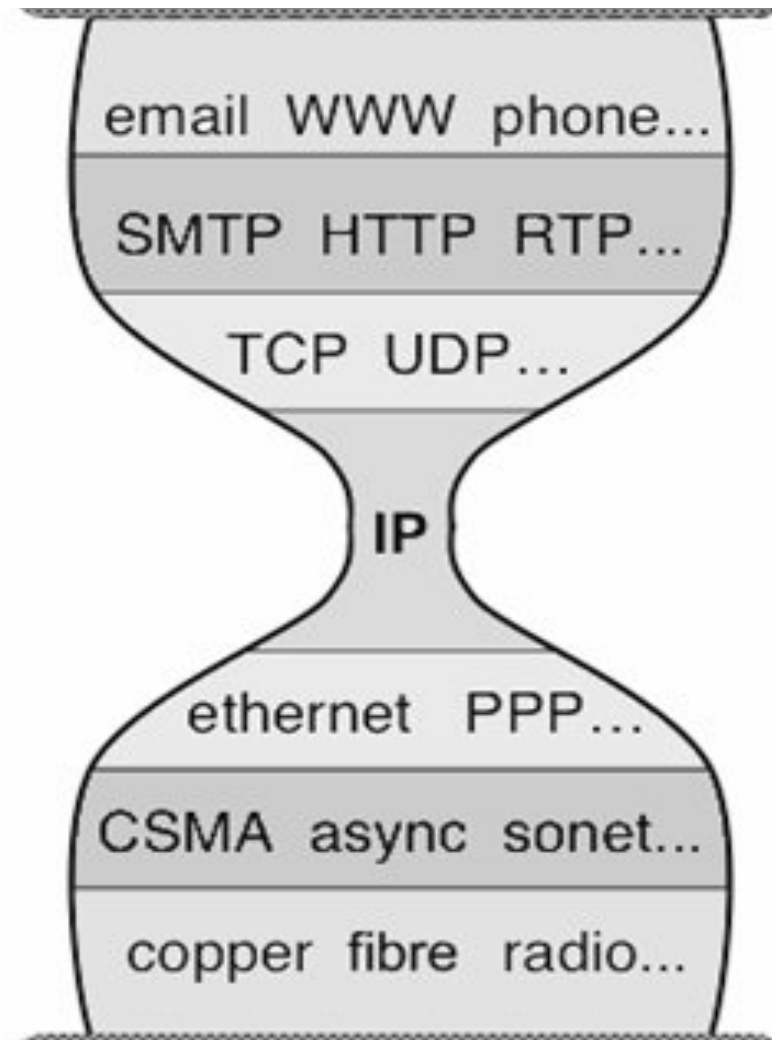ethernet PPP…

CSMA async sonet...

copper fibre radio...
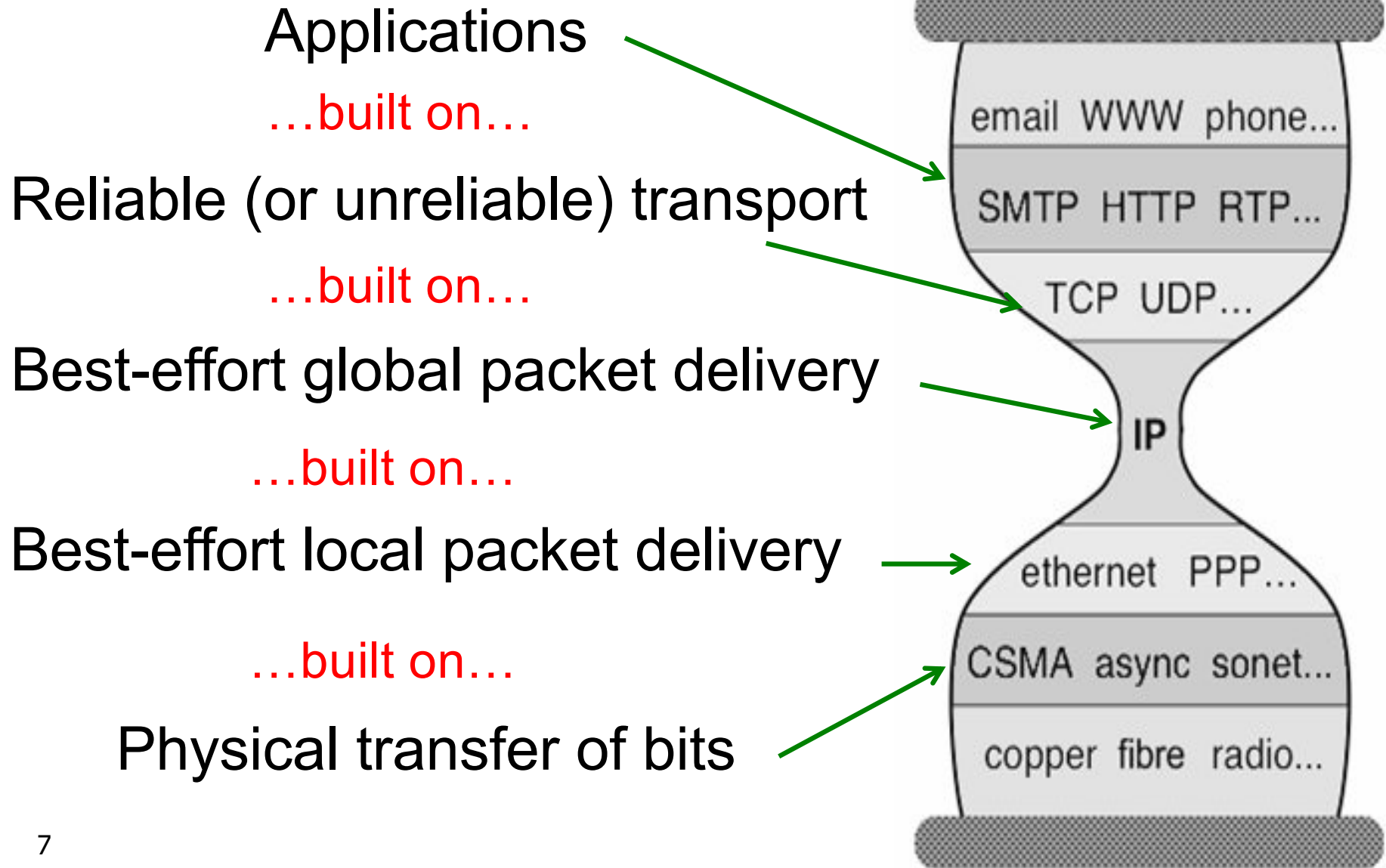
Internet protocol stack

❖ *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"

email WWW phone...

SMTP HTTP RTP...

TCP UDP...

IP

ethernet PPP...
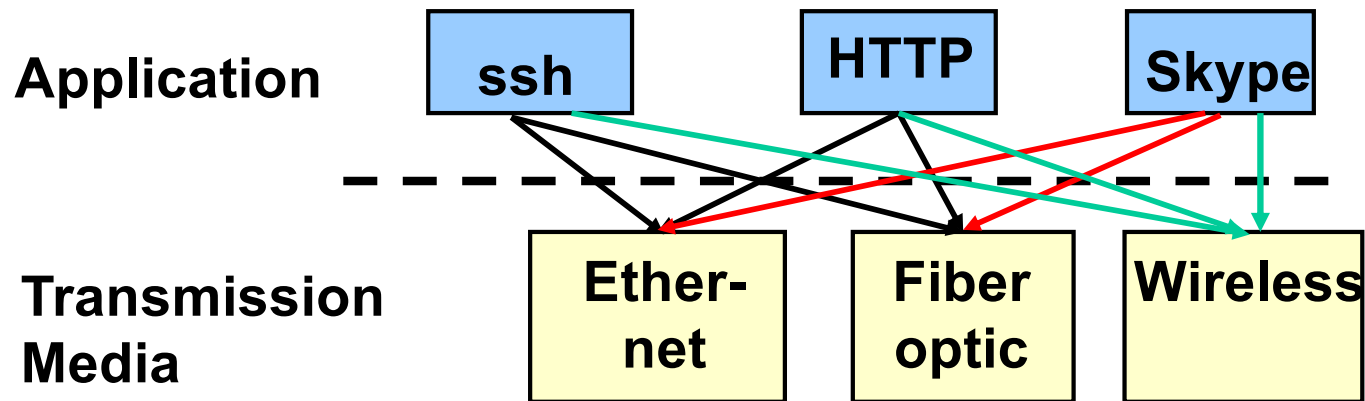
CSMA async sonet...

copper fibre radio...

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

email WWW phone...

SMTP HTTP RTP...

TCP UDP...

IP

ethernet PPP...

CSMA async sonet...

copper fibre radio...

An Example: No Layering

Application

ssh   HTTP   Skype

Transmission Media

Ether-net   Fiber optic   Wireless

❖ 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

Application   ssh   HTTP   Skype

Transport & Network

Transmission Media   Ethernet   Fiber optic   Wireless
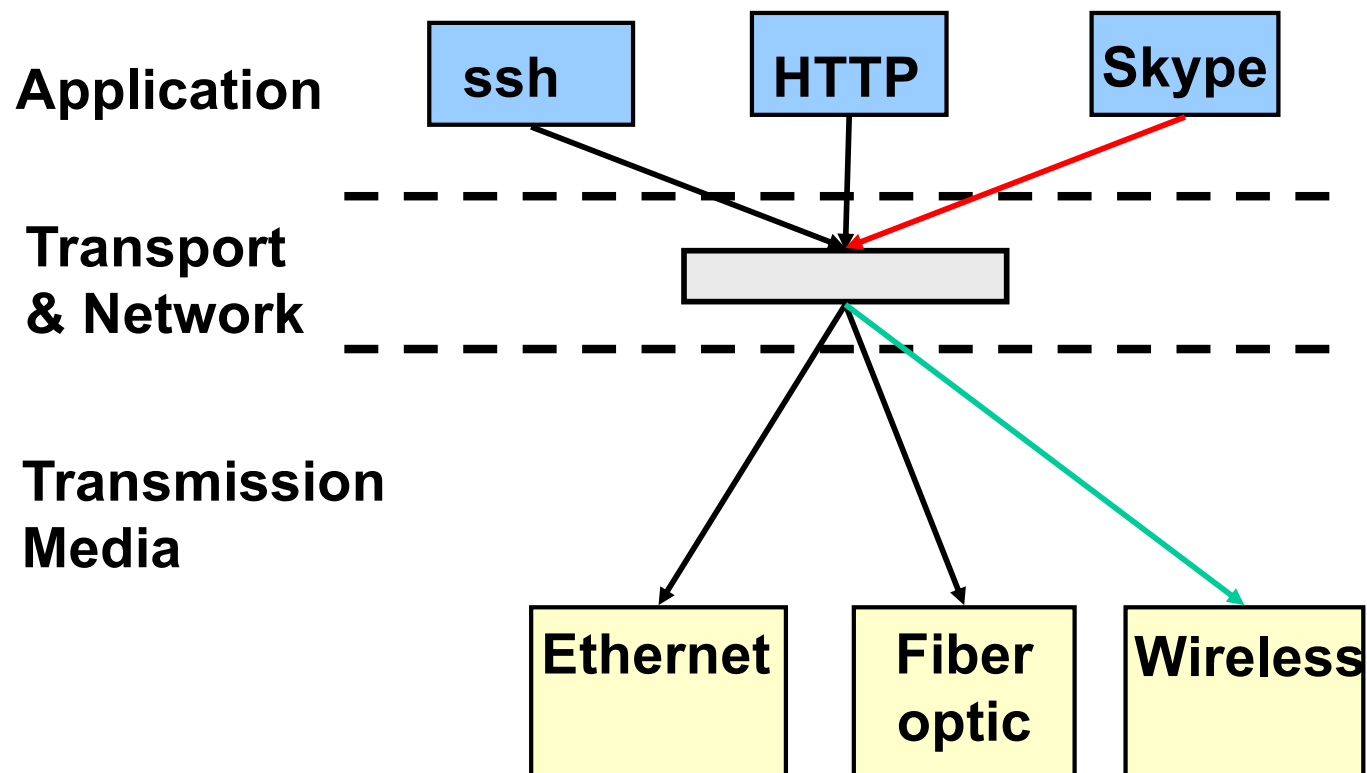
Is Layering Harmful?

- ❖ 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
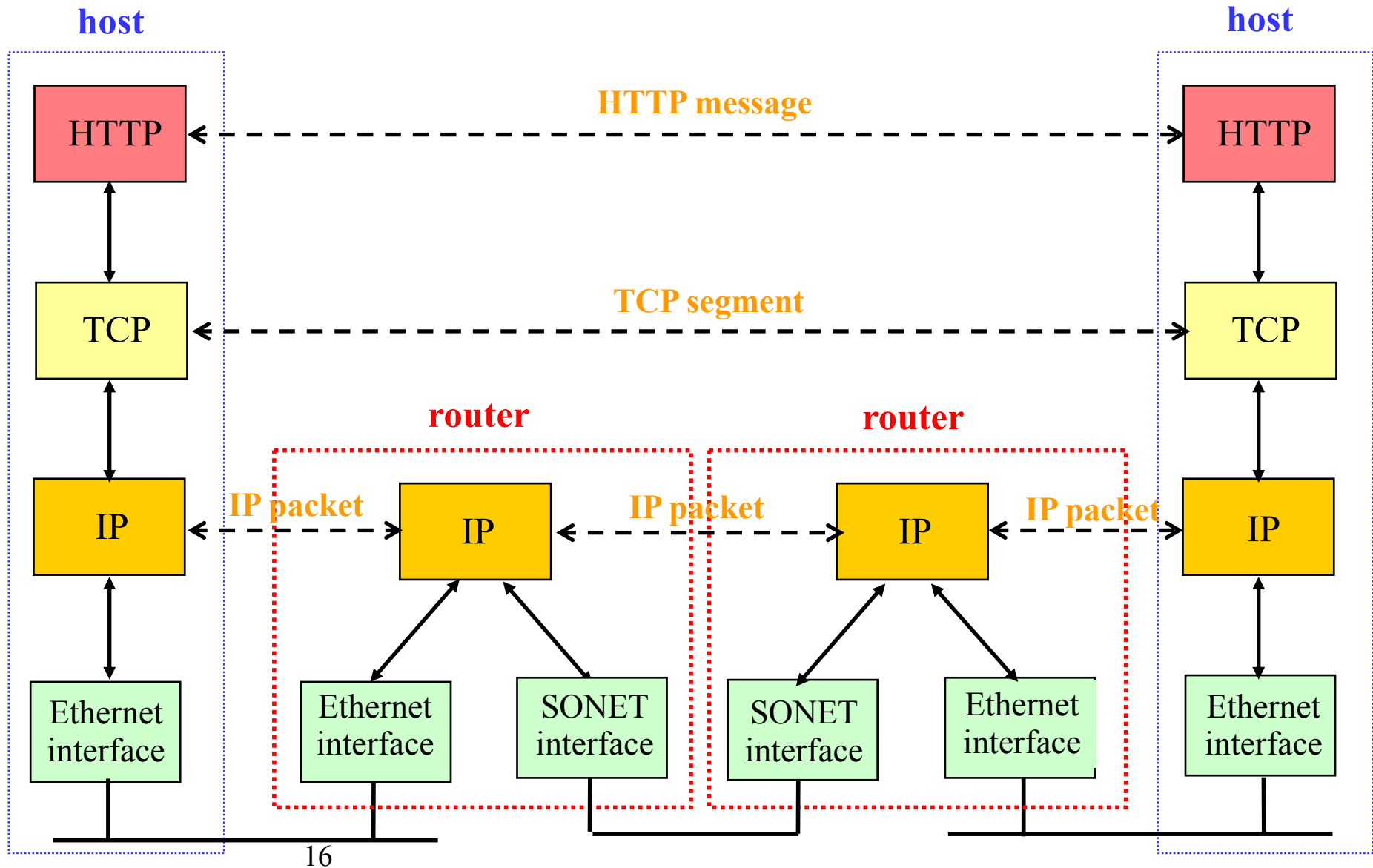 ▪ Routers
  ▪ Switches

❖ 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!
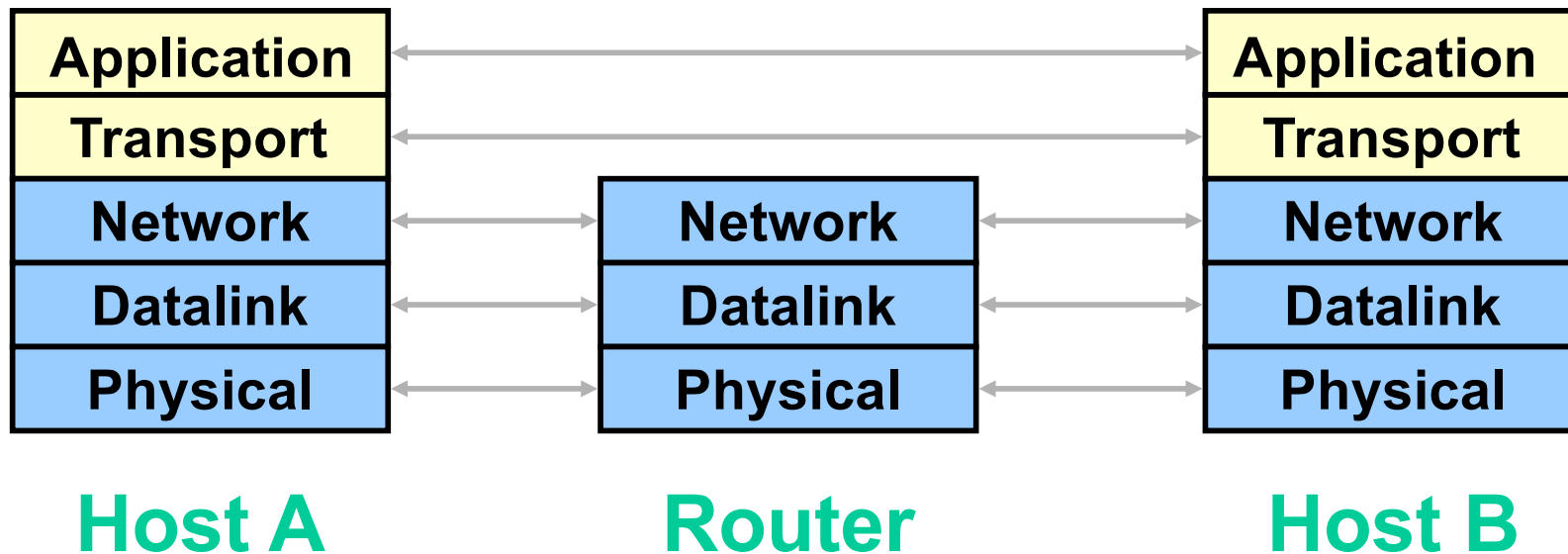
What Gets Implemented on Router?

❖ 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
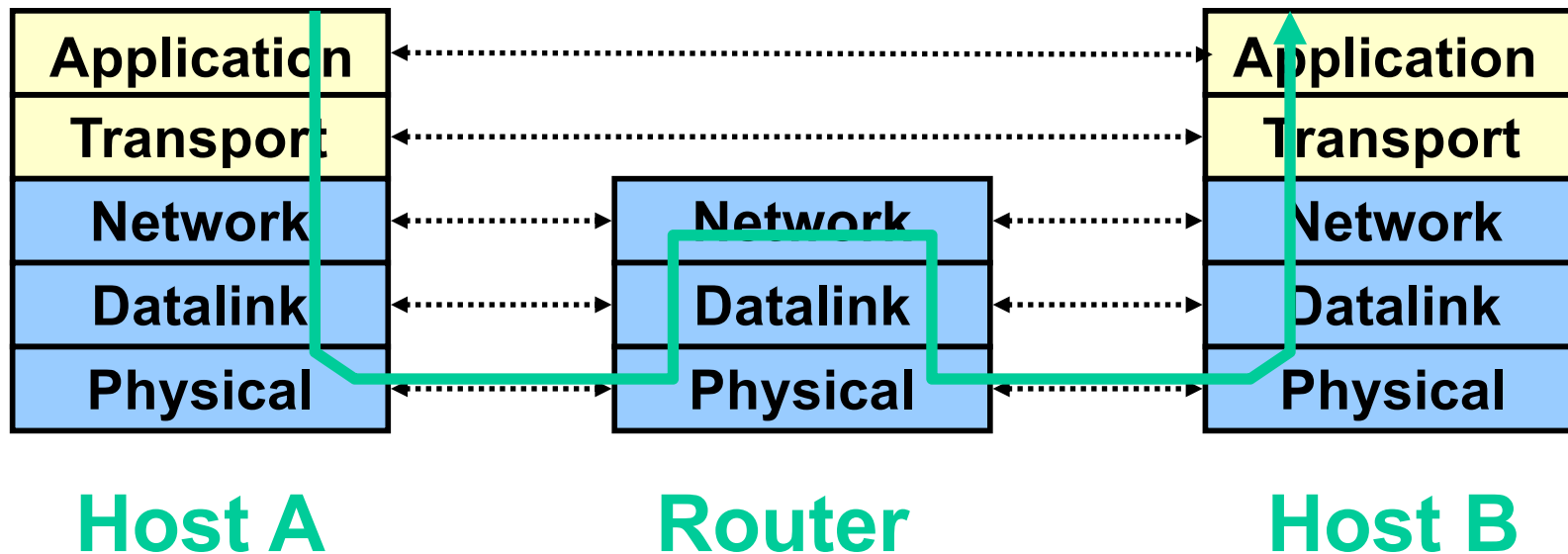
Internet Layered Architecture

Logical Communication

❖ Layers interacts with peer's corresponding layer
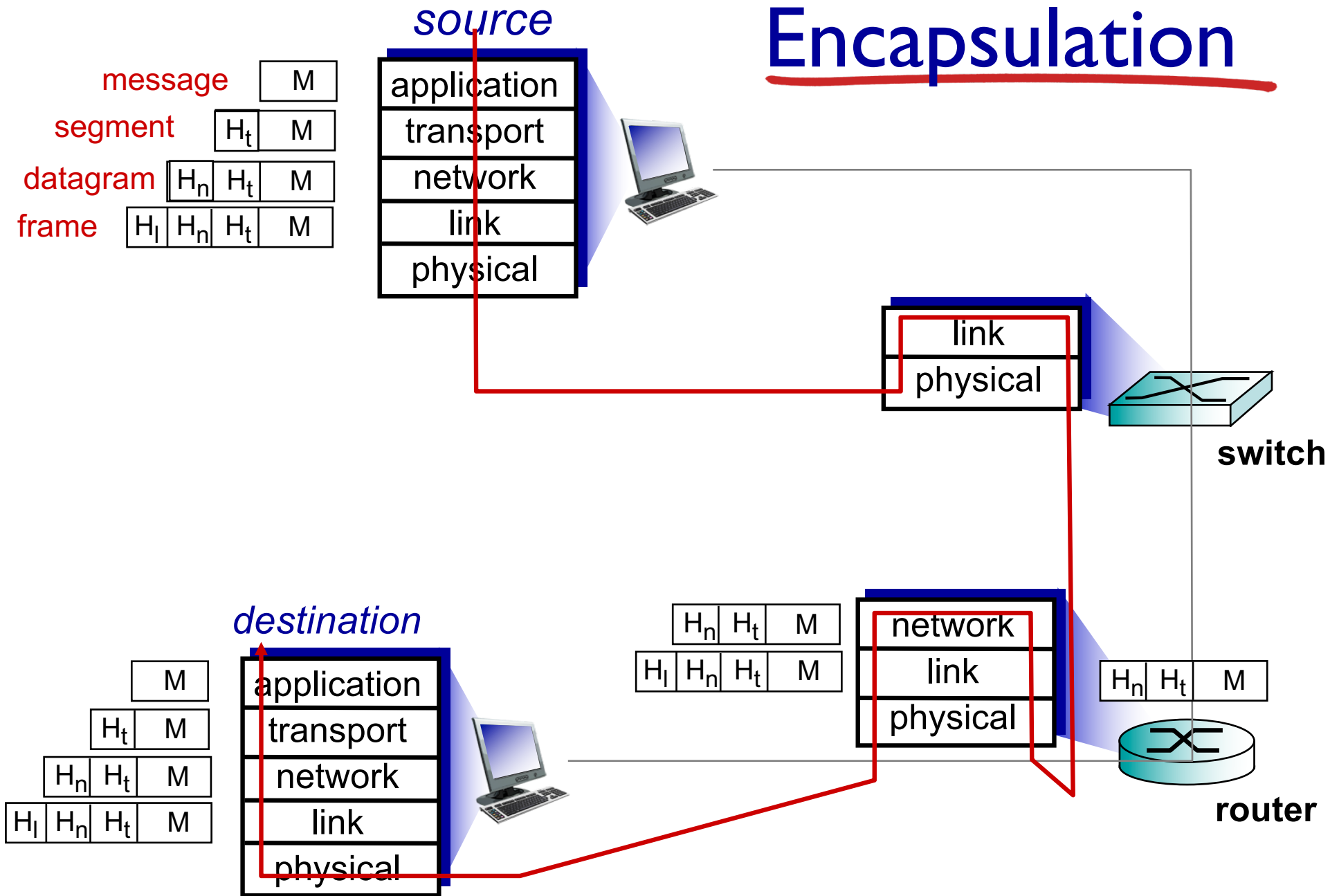
Application		Application
Transport		Transport
Network	Network	Network
Datalink	Datalink	Datalink
Physical	Physical	Physical
Host A	**Router**	**Host B**

Physical Communication

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

Host A	Router	Host B
Application		**Application**
Transport		**Transport**
Network	**Network**	**Network**
Datalink	**Datalink**	**Datalink**
Physical	**Physical**	**Physical**

Encapsulation

message   M

segment H_t M

datagram H_n H_t M

frame H_l H_n H_t M

source

application
transport
network
link
physical

link
physical

switch

destination

application
transport
network
link
physical

M

H_t M

H_n H_t M

H_l H_n H_t M

H_n H_t M

H_l H_n H_t M

network
link
physical

H_n H_t M

router

1. Introduction: roadmap

Self study