Fundamentals

Slides used in TDDE48 (Mobile Networks) @ LiU, Sweden, Fall 2025
Niklas Carlsson (https://www.ida.liu.se/~nikca89/)

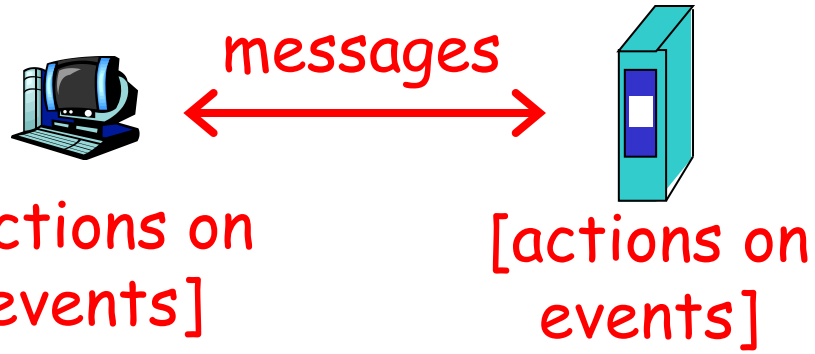
Slides in this course are adapted or based on various on-line resources (including lectures notes by Anirban Mahanti, Carey Williamson, Jim Kurose, and Keith Ross)

Background assumptions

- I will assume that you refresh your memory of what you learned in TDTS11
  - E.g., see "Computer Networks (TEN 1)" here https://www.ida.liu.se/~TDDE35/timetable/index.en.shtml

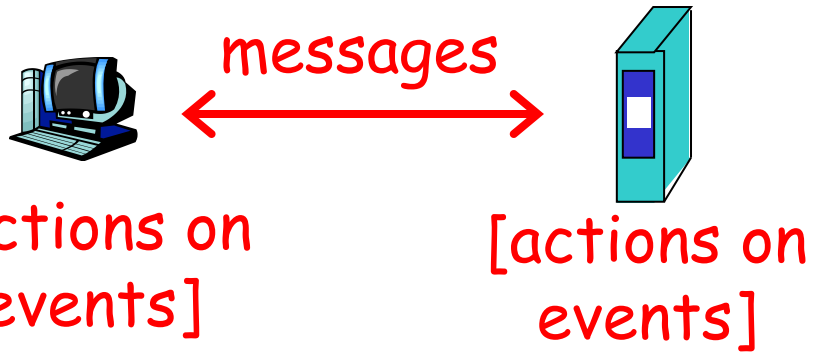
- The following slides go over some of the fundamentals again.
  - During the lecture I will put this into a wireless context and add some additional texture/depth

What's a protocol?



Need:

What's a protocol?



Network protocols:

- Define the order and format of messages exchanged
- Defines the actions to take in response to events (e.g., message arrivals, transmissions, losses, and timeouts)

What's a protocol?

Wireless vs. wired world …

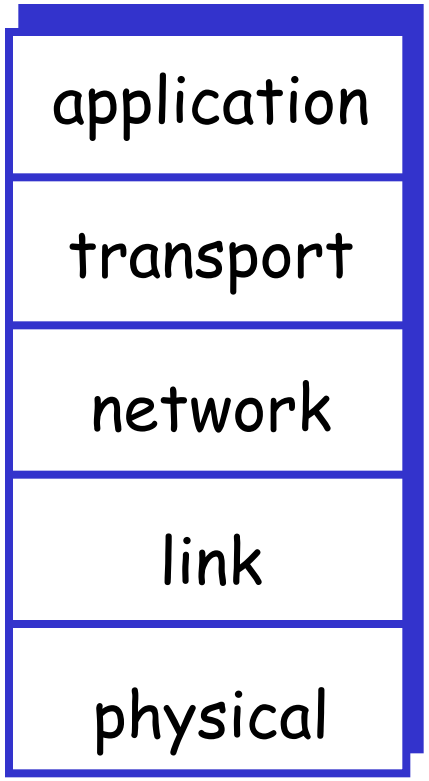
What's a protocol?

Wireless vs. wired world …

- Losses and scrambled data much more common
- Mobility, …
- Battery constrained,
- ….

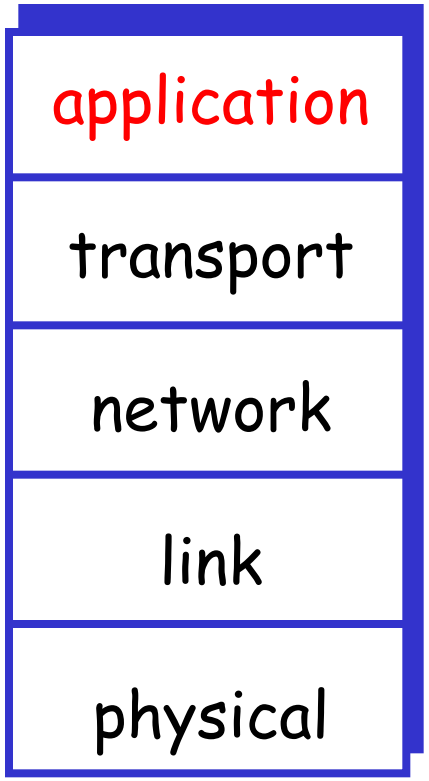
Need mechanism to coop with these events …

In this course … the Internet protocol stack…



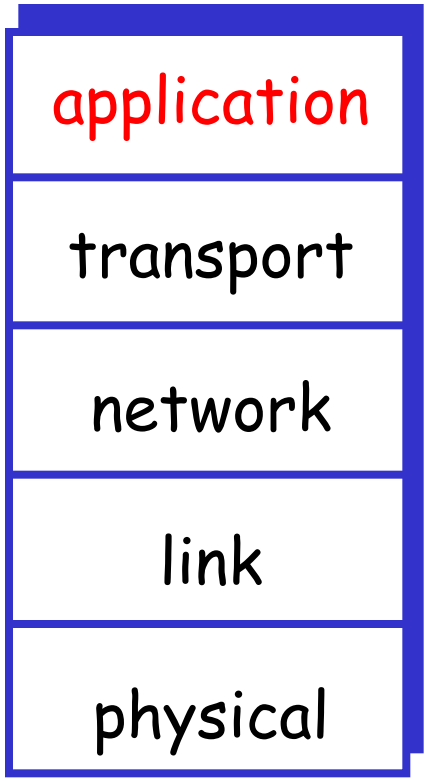
In this course … the Internet protocol stack…

- application: supporting network applications
- transport: host-host data transfer
- network: routing of datagrams from source to destination
- link: data transfer between neighboring network elements
- physical: bits "on the wire"



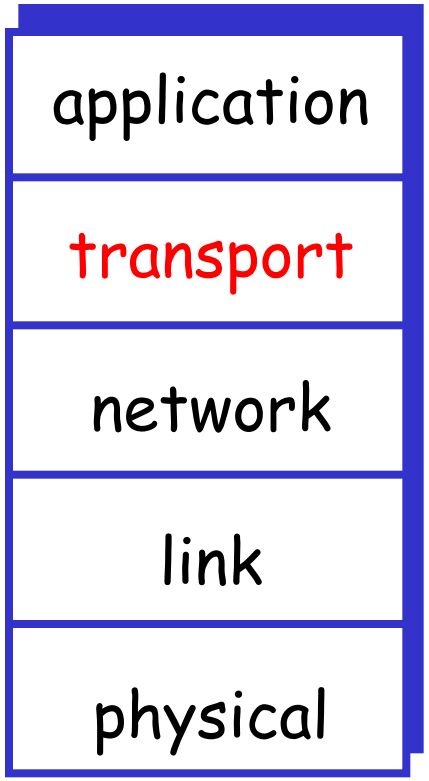
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



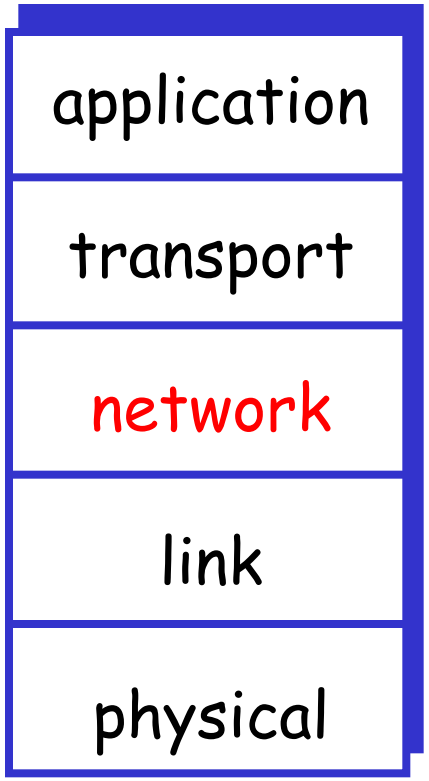
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



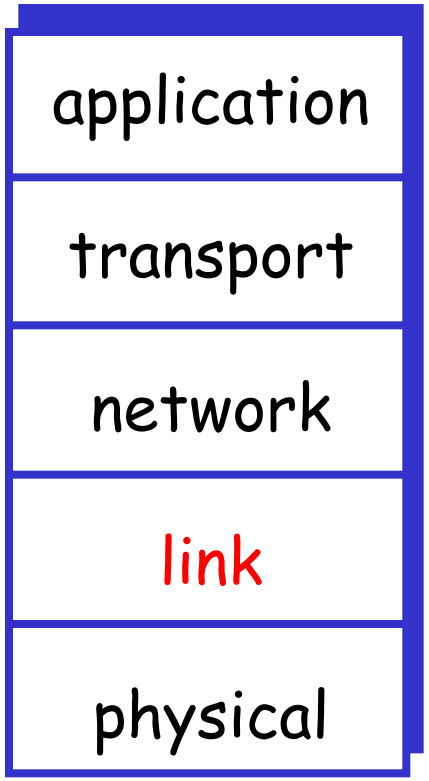
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



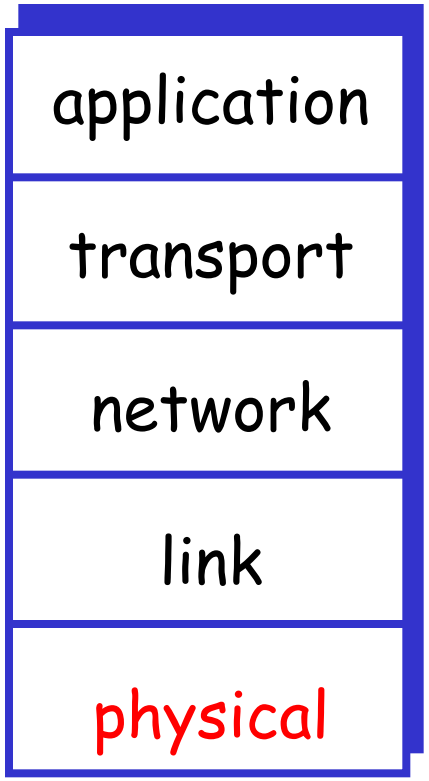
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



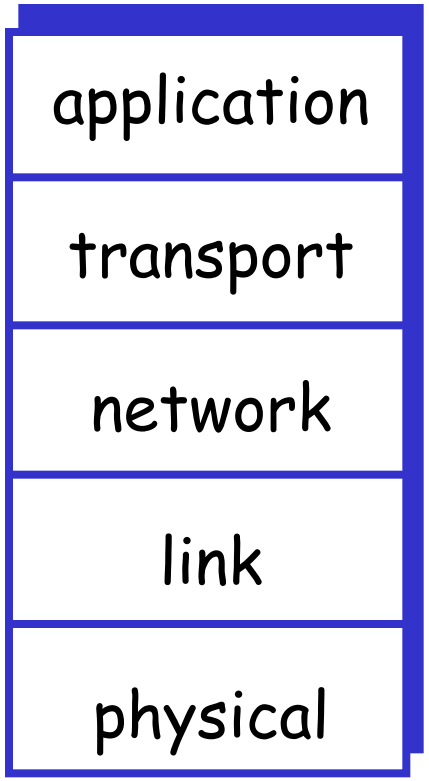
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



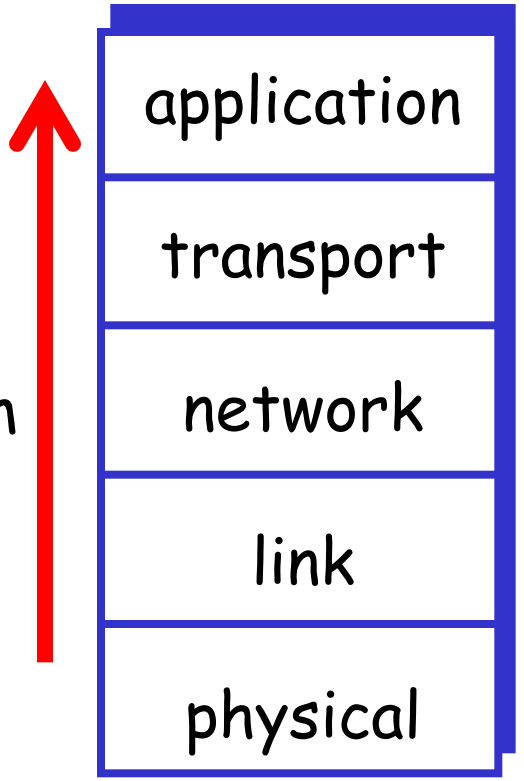
In this course … the Internet protocol stack…

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"

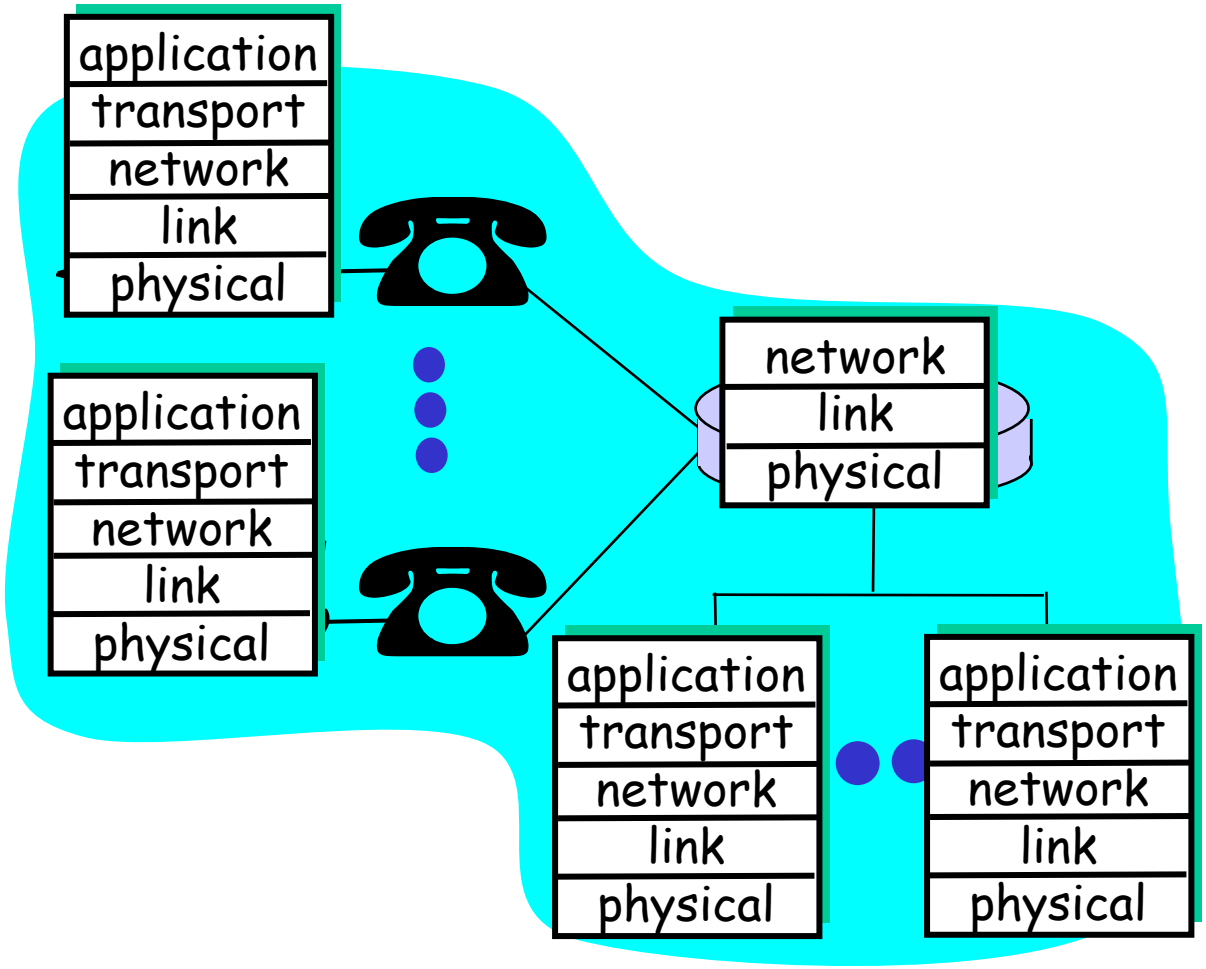


In this course … the Internet protocol stack…

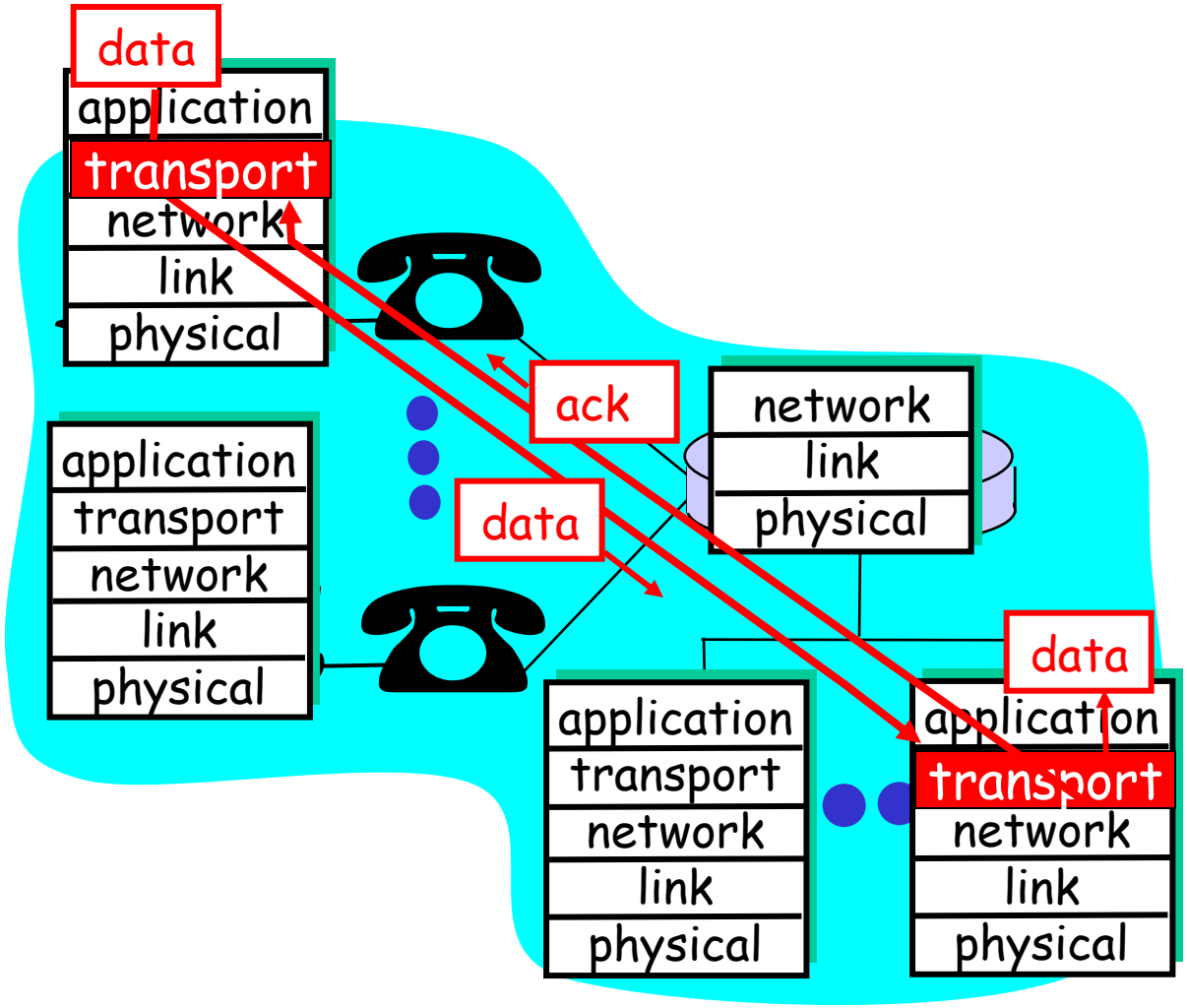
- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - WiFi, Ethernet
- physical: bits "on the wire"



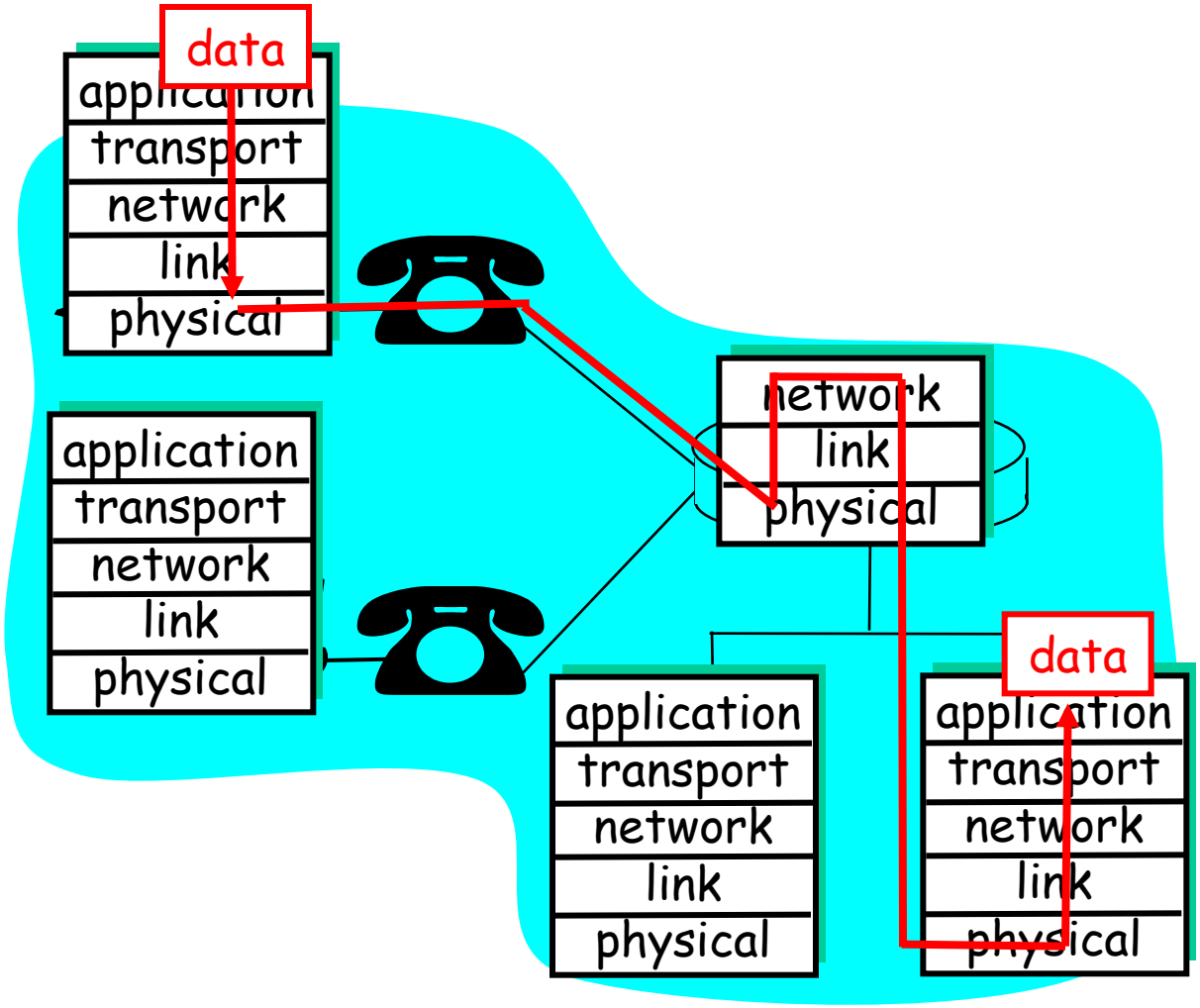
Layering: logical communication



Layering: *logical* communication



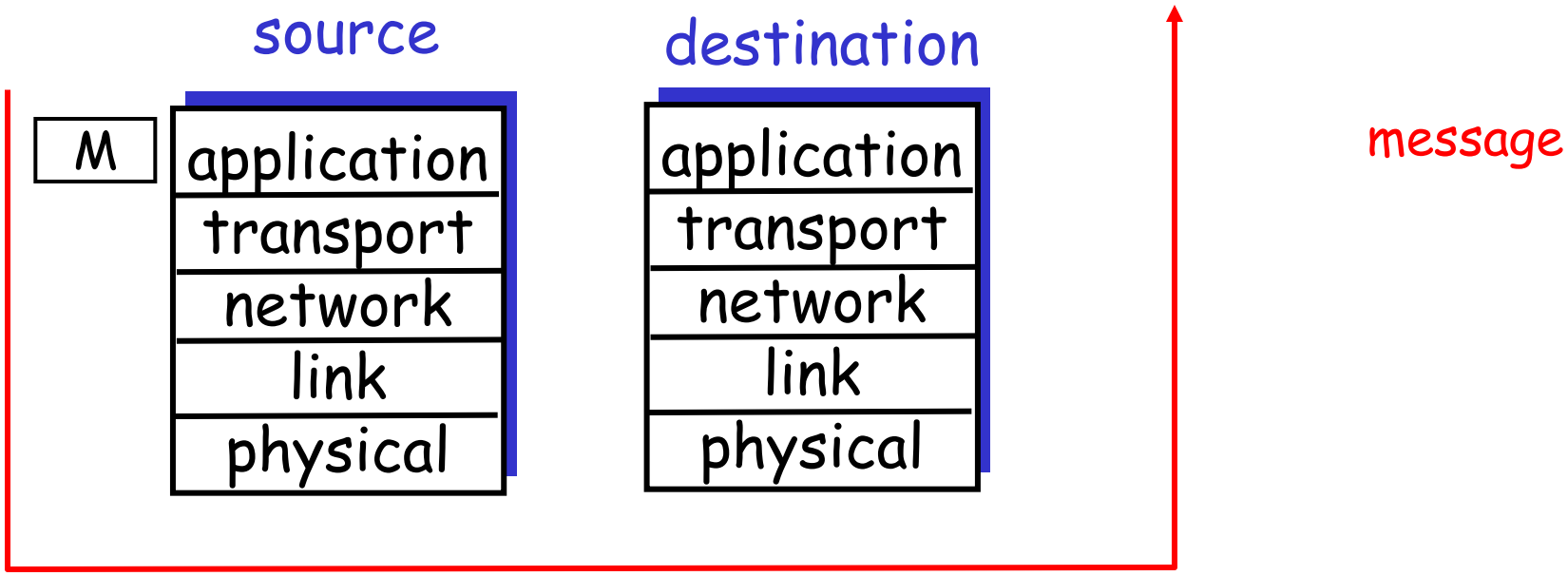
Layering: *physical* communication



Encapsulation: Layering and data

Each layer takes data from above

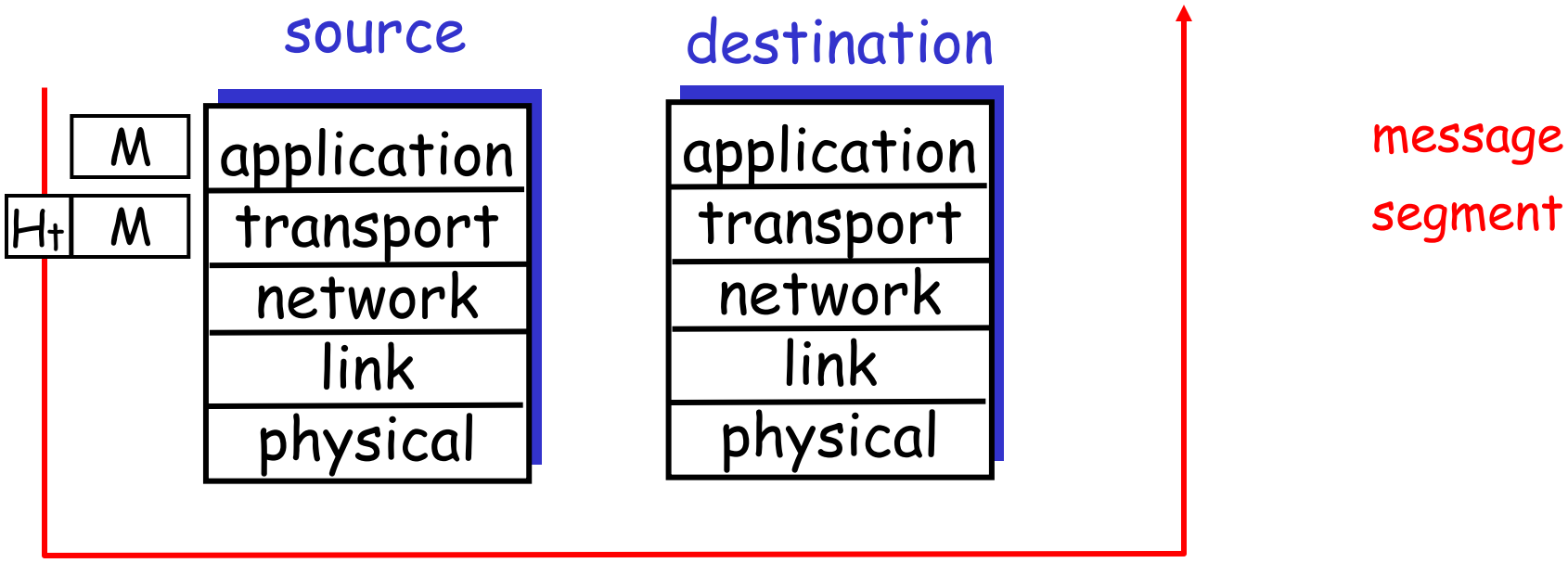
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

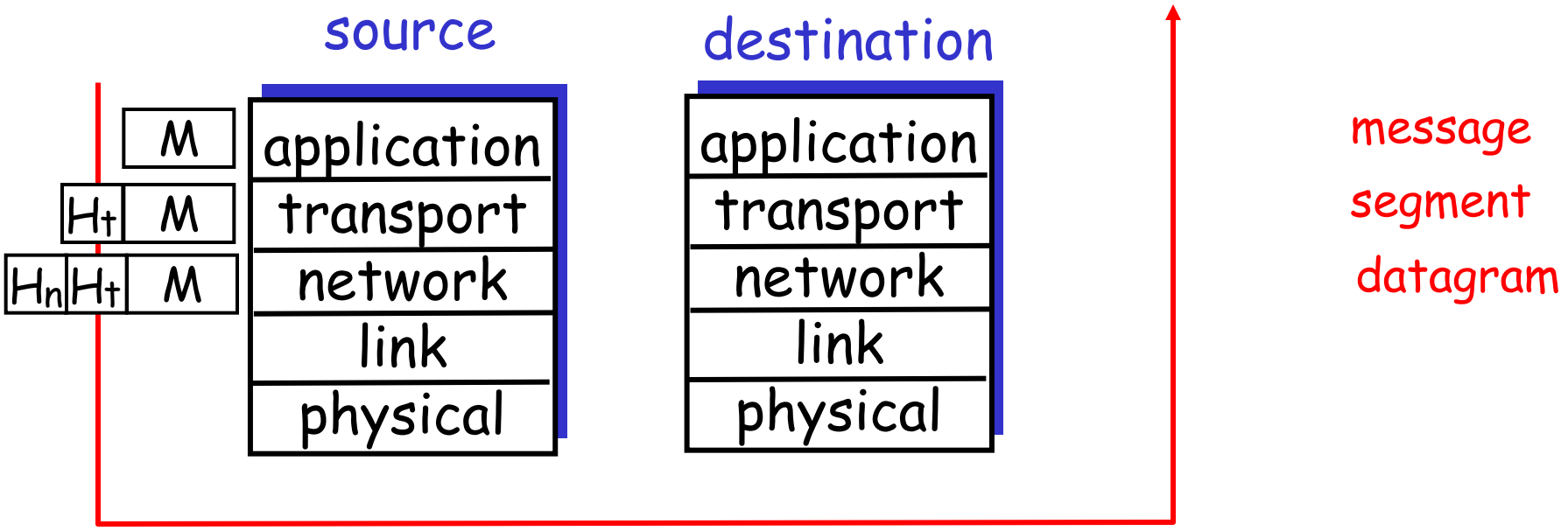
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

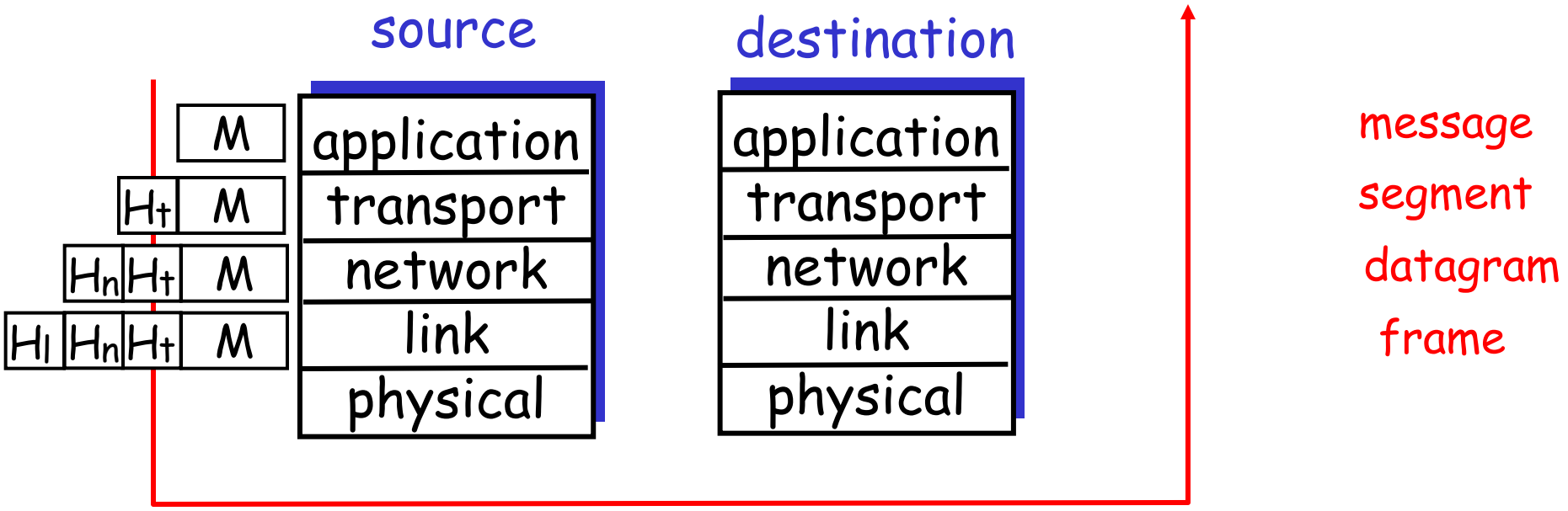
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

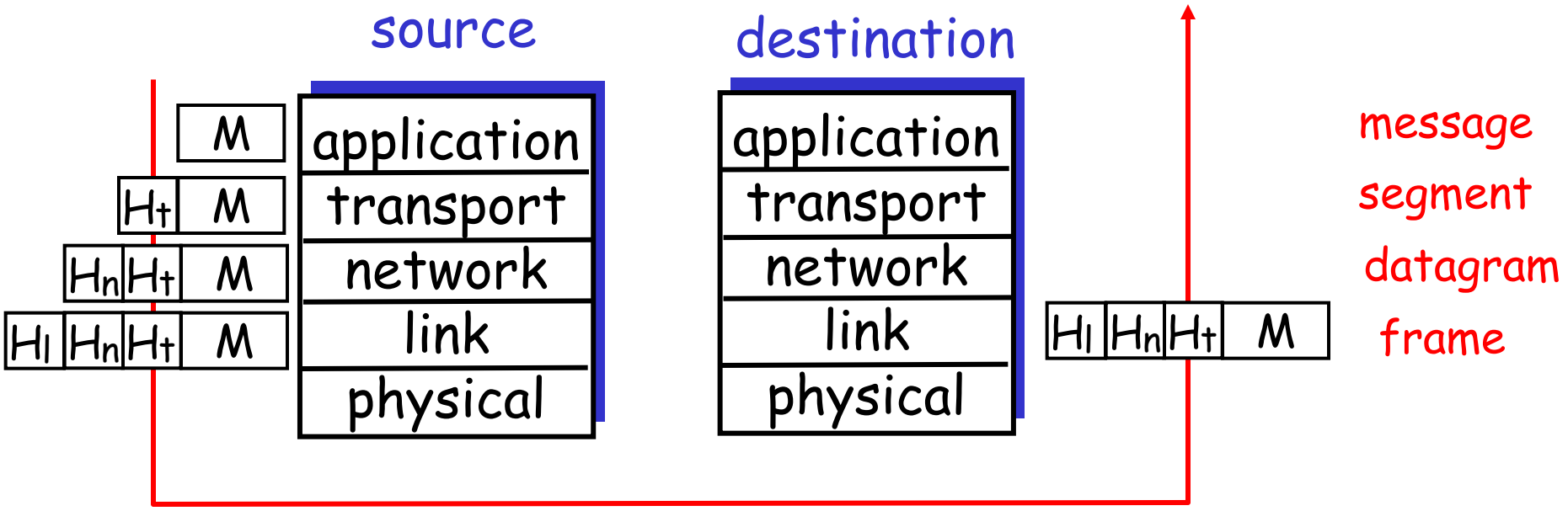
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

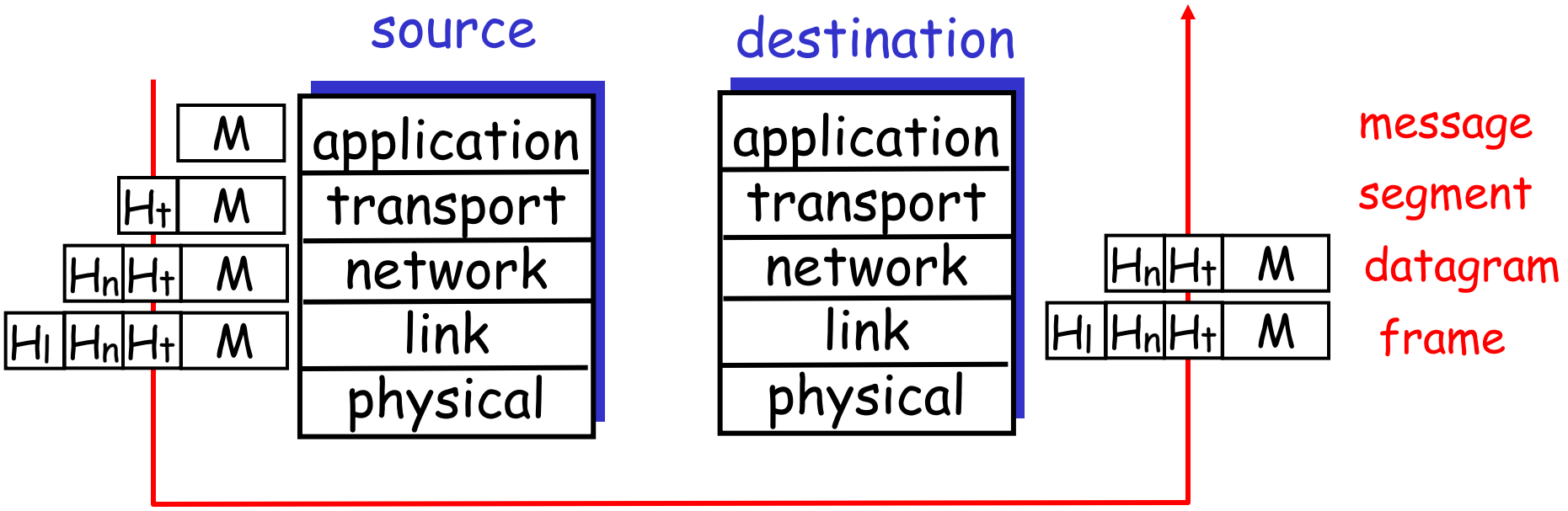
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

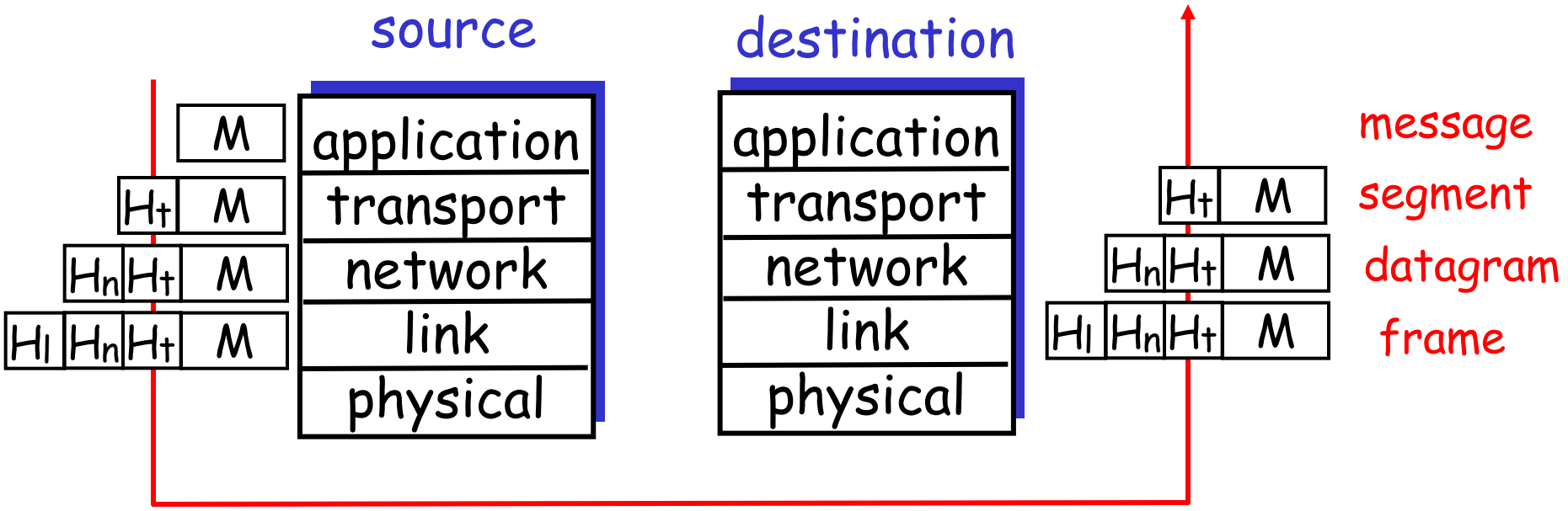
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

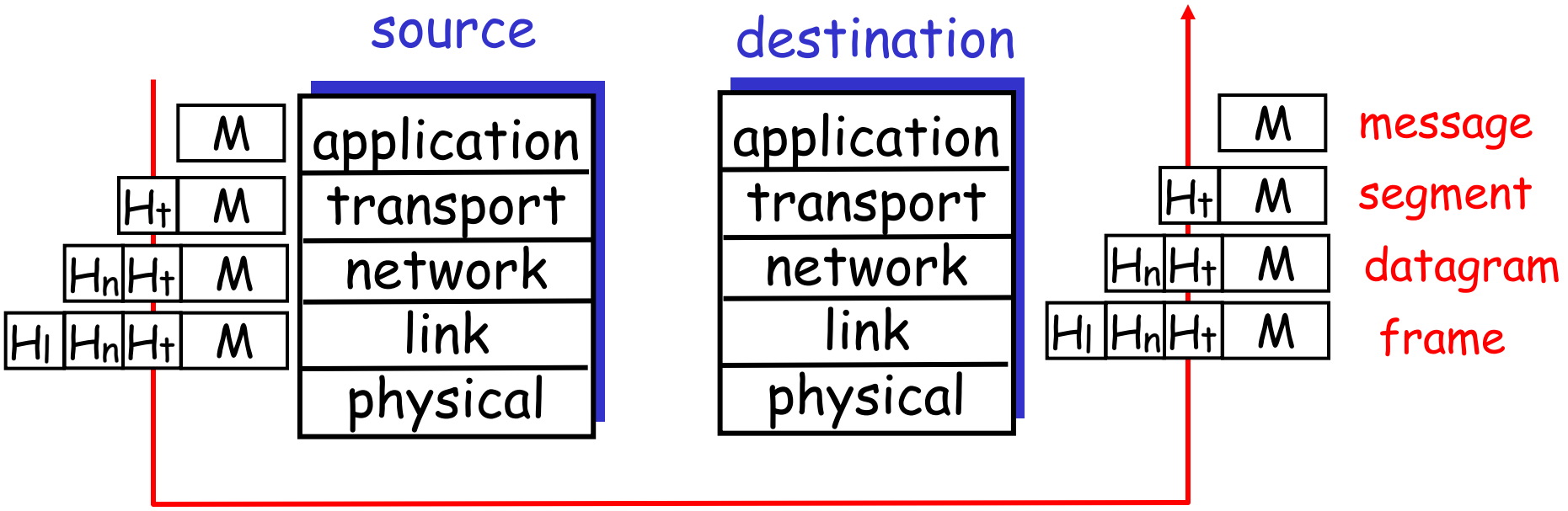
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation: Layering and data

Each layer takes data from above

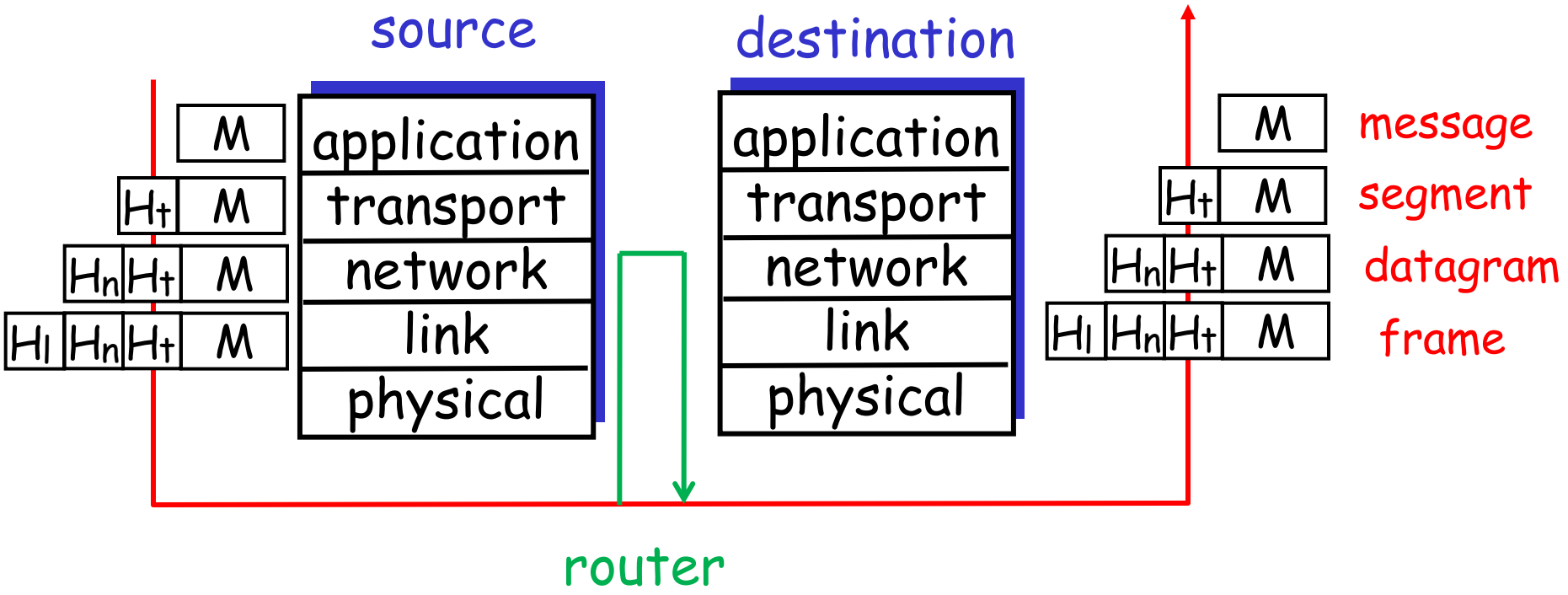
- adds header information to create new data unit
- passes new data unit to layer below



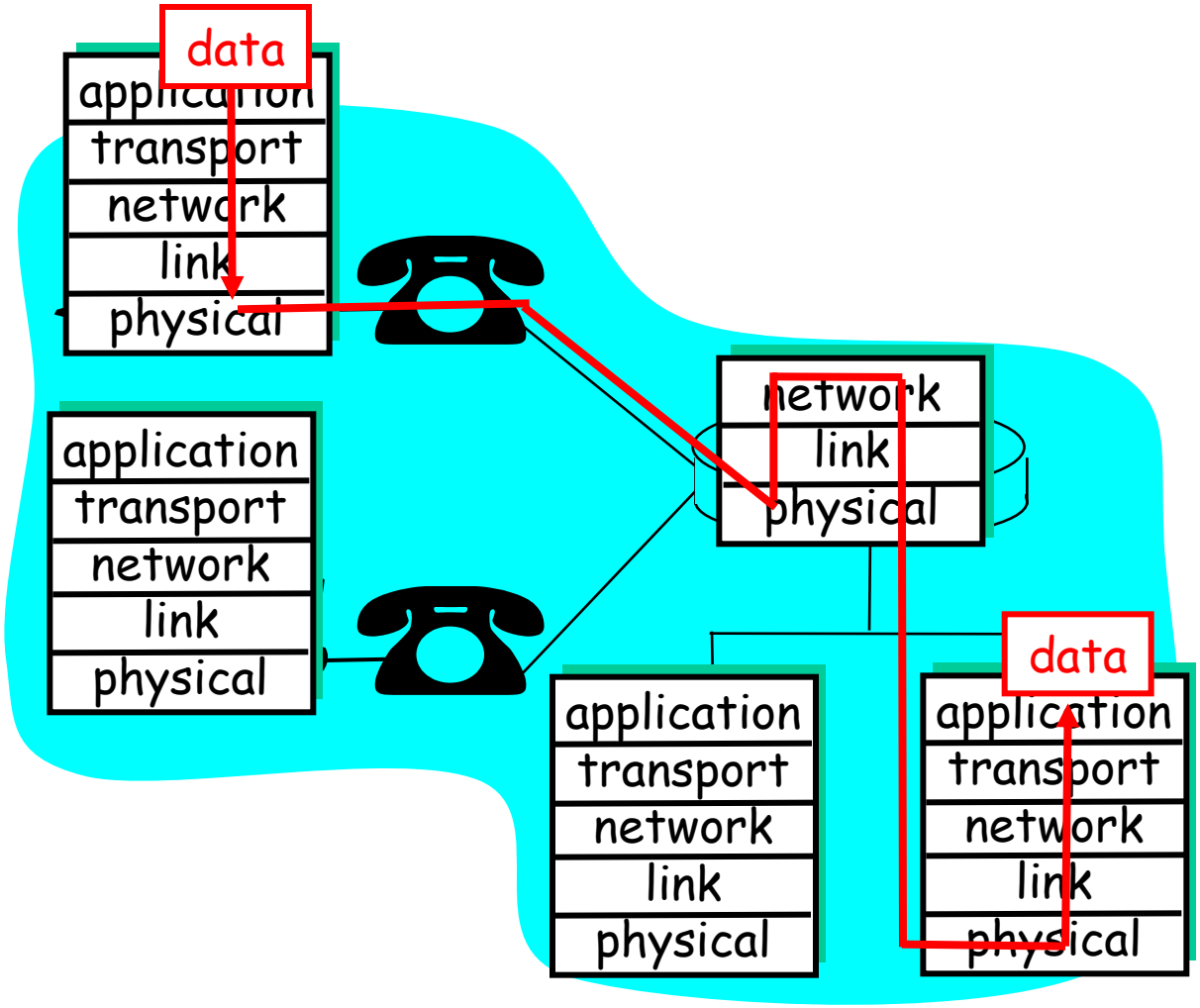
Encapsulation: Layering and data

Each layer takes data from above

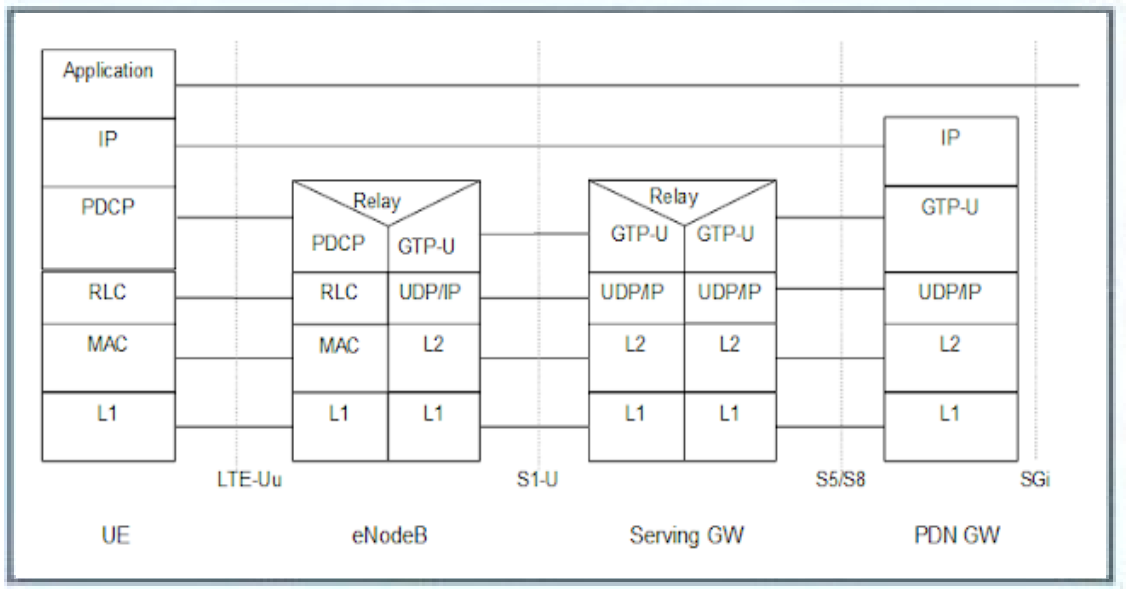
- adds header information to create new data unit
- passes new data unit to layer below

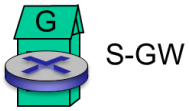


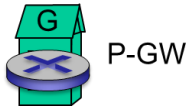
Layering: *physical* communication



Layering: *physical* communication

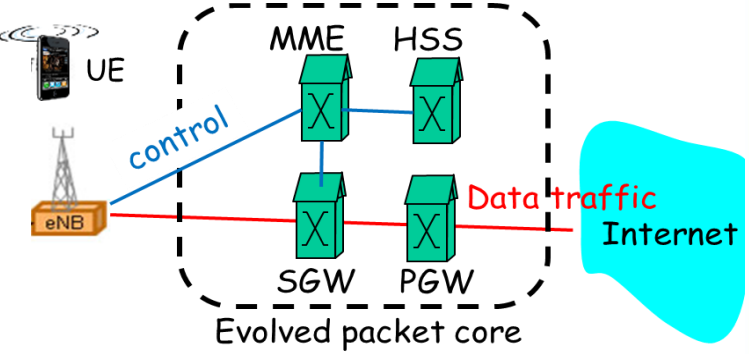


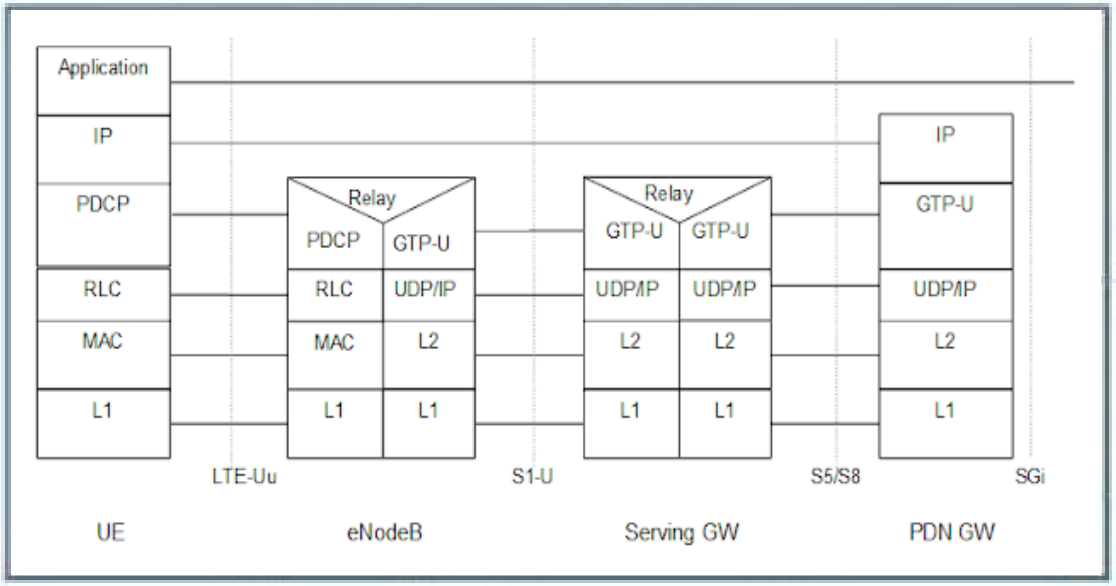




Internet

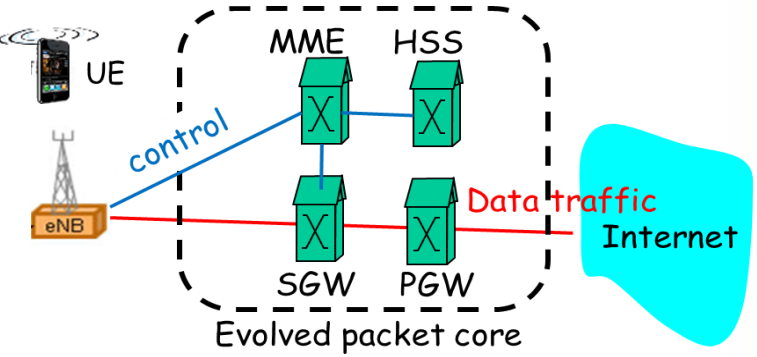
Layering: *physical* communication

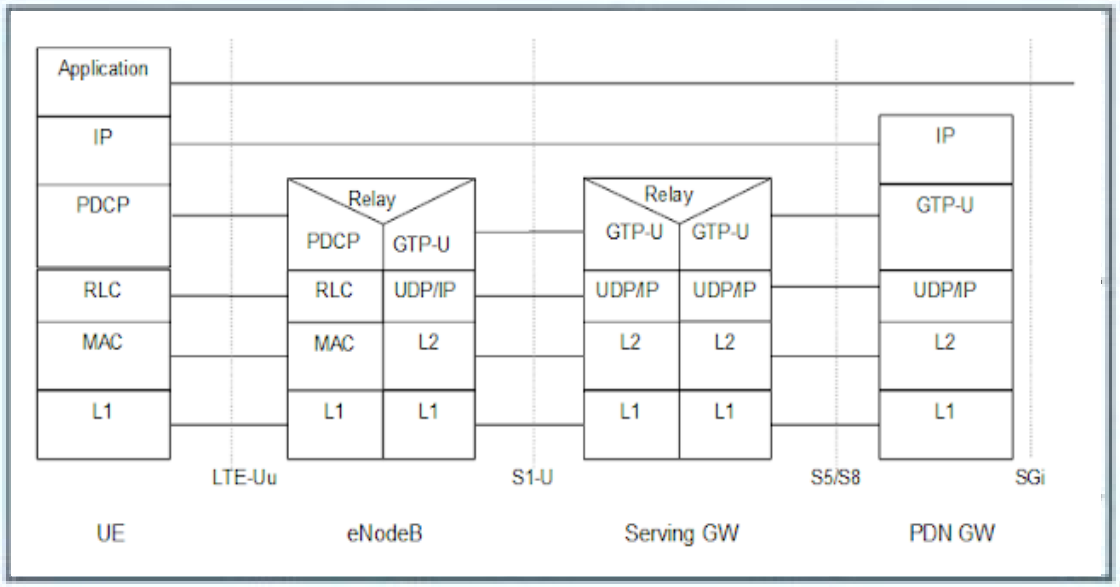




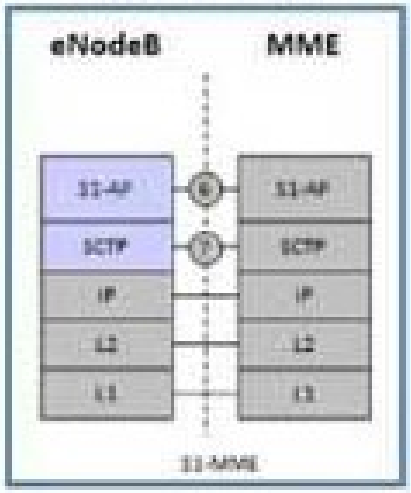
Layering: *physical* communication

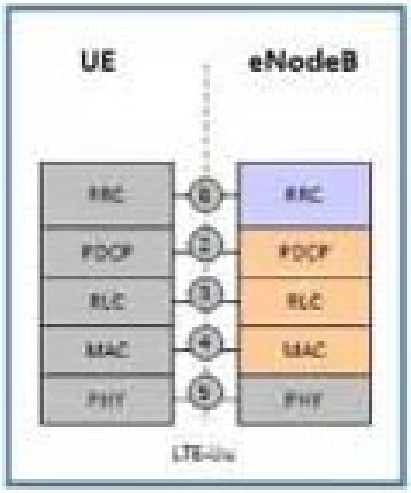
User plane

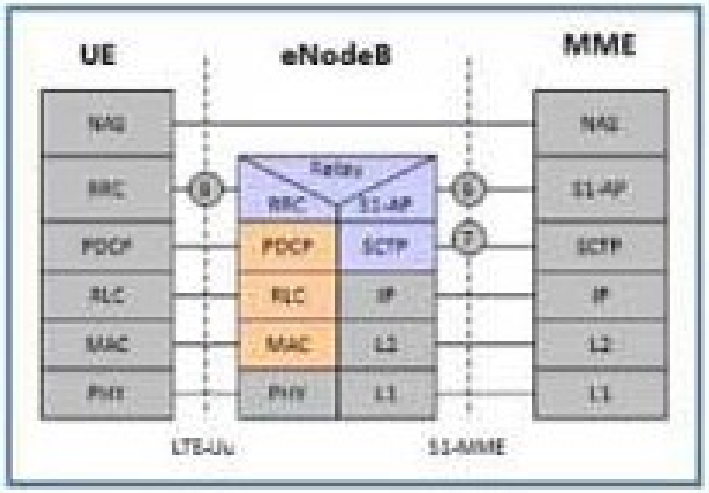




Control plane

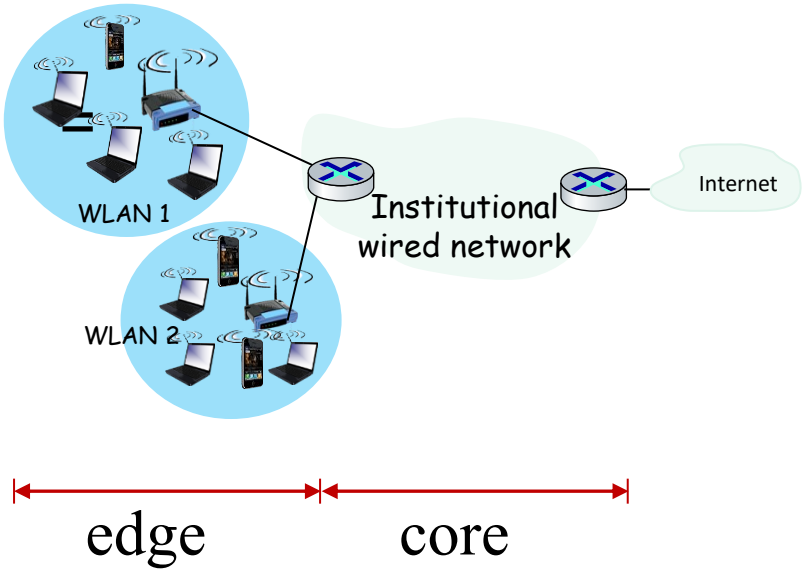




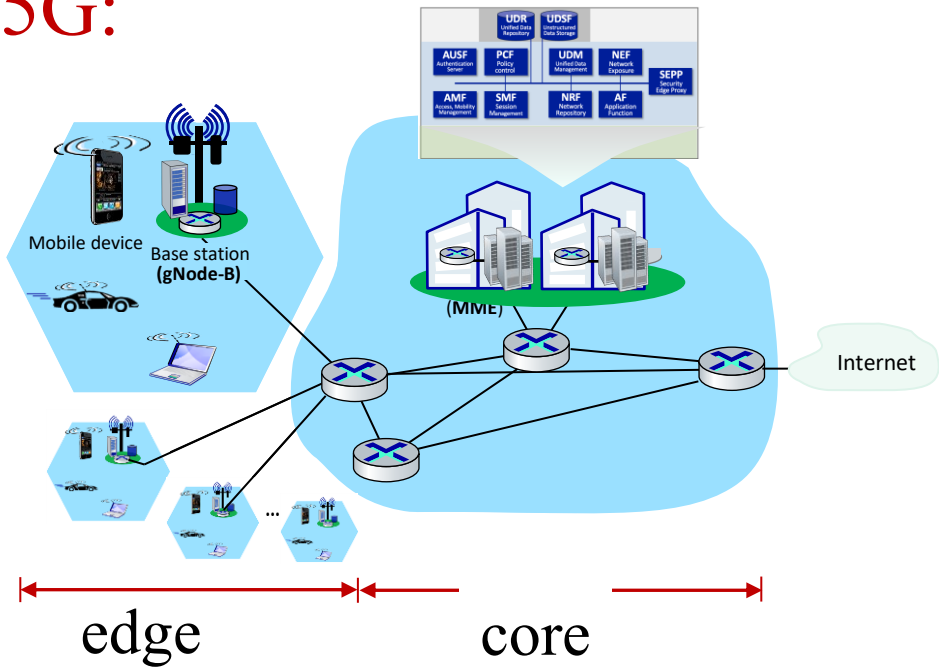


Wireless networks: edge and core networks

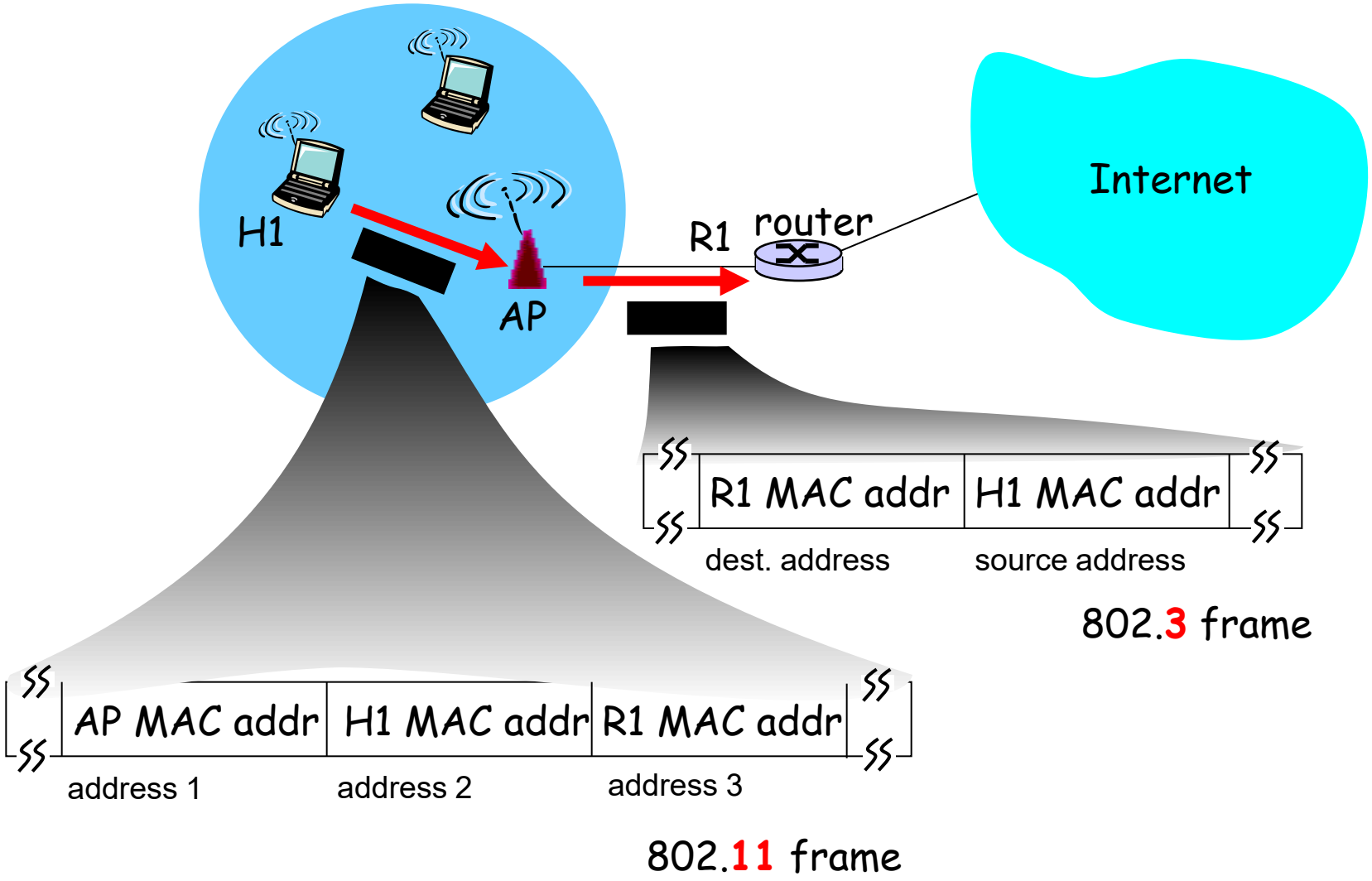
WiFi:



5G:



802.11 frame: addressing

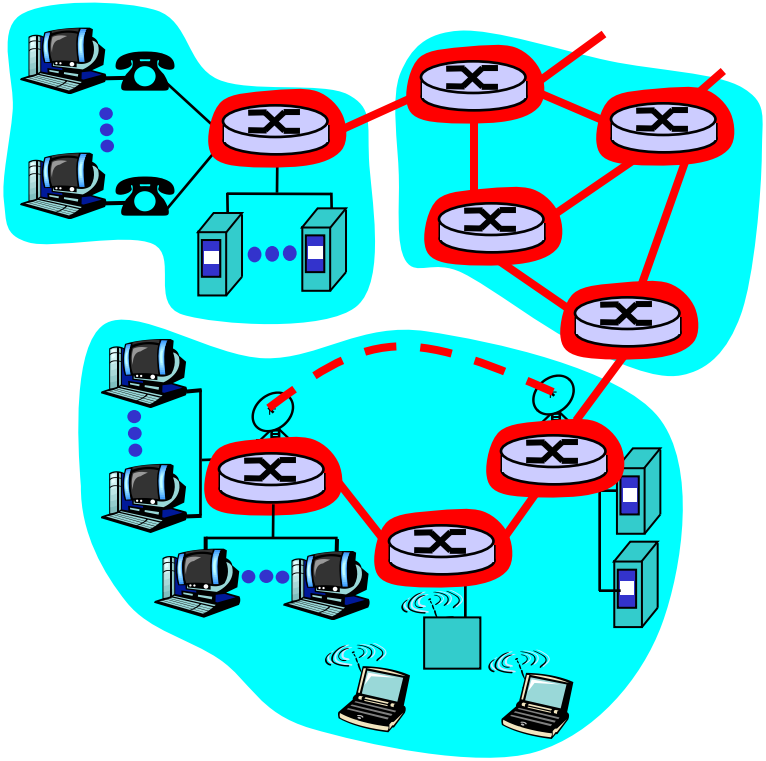


- *the* fundamental question: how is data transferred through net?
  1. circuit-switching:

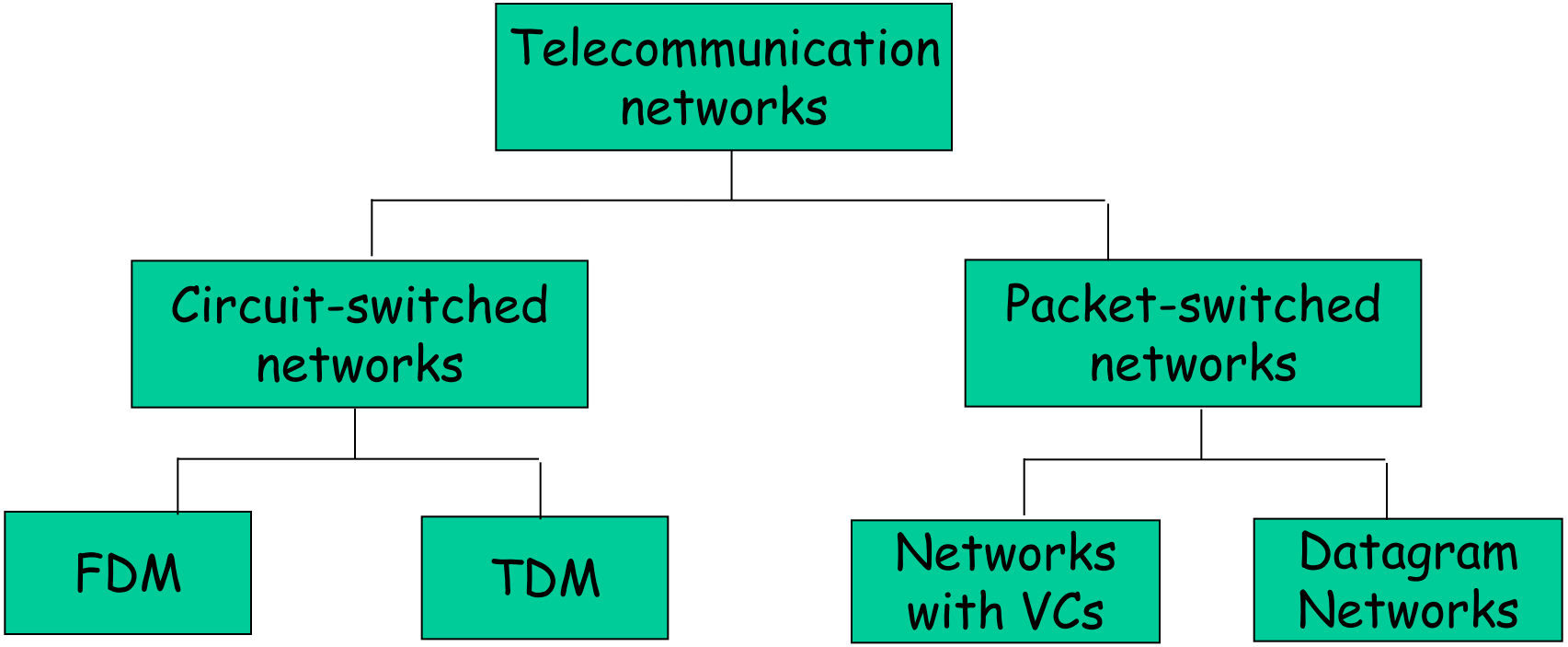
     dedicated circuit per call: telephone net

  2. packet-switching:

     data sent thru net in discrete "chunks"



Network Taxonomy



Mobile networks

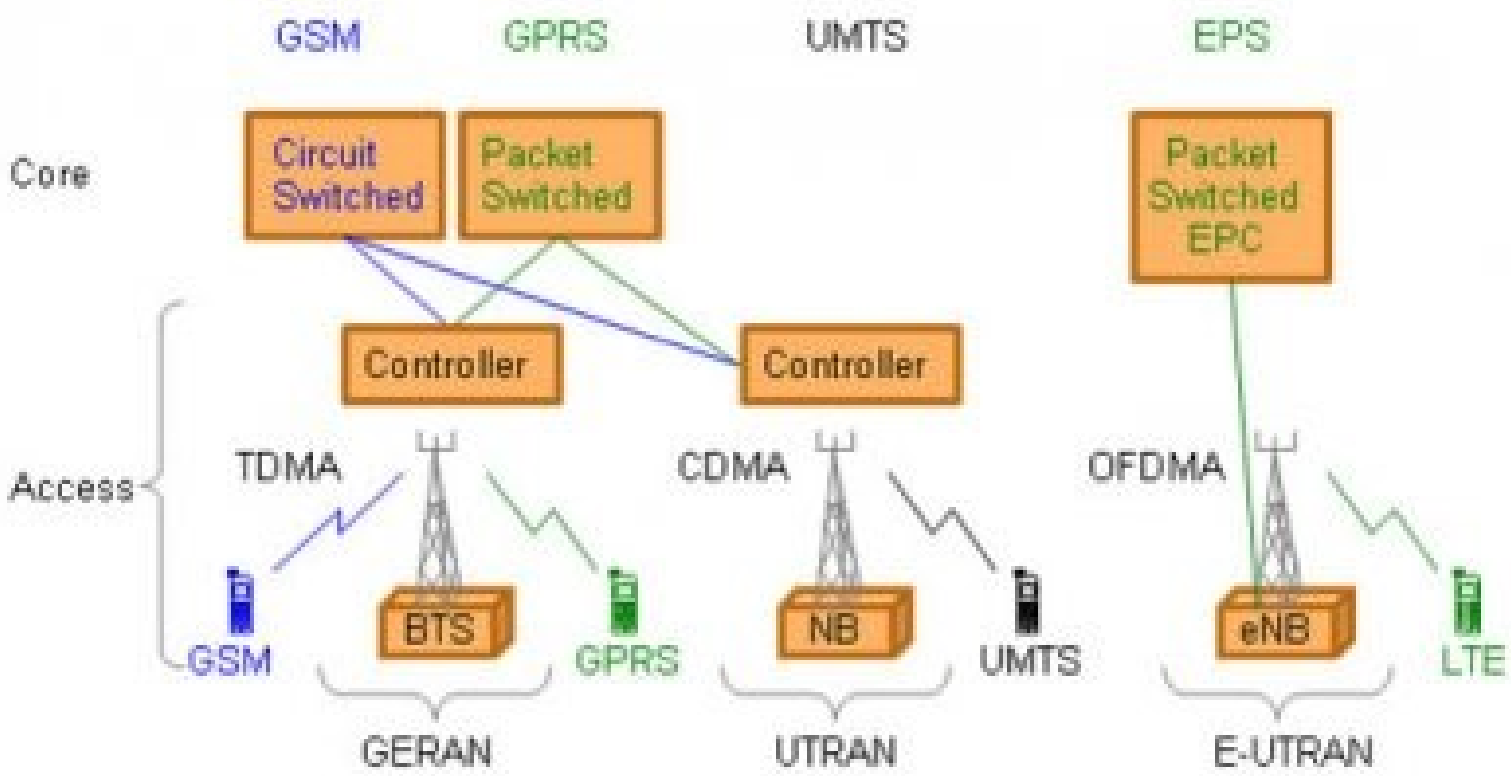
The Internet: Packet switching or circuit switching ??

Does use of Ethernet vs WiFi matter ??

How about 1G 2G, 3G, 4G, 5G... ??

When did mobile switch from “CS” to “PS”?

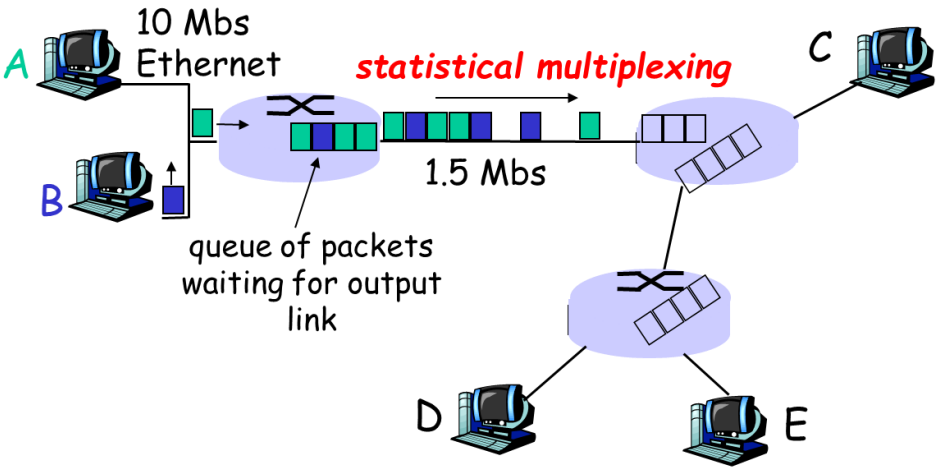
Mobile networks



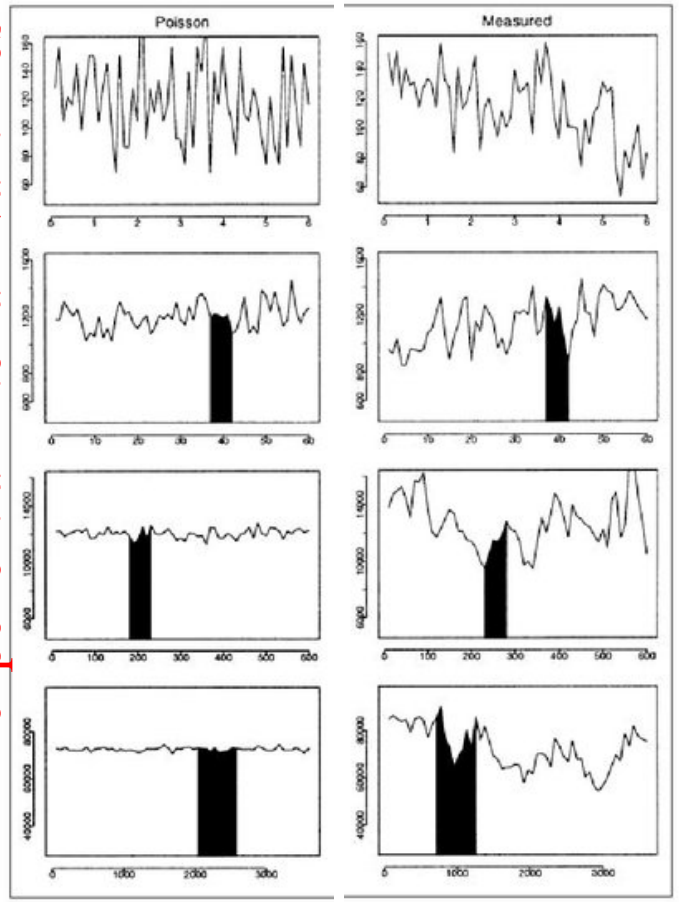
- Also, mobile networks are going towards IP-based packet switching …

[Fig. from 3GPP website]

Packet-Switching: Statistical Multiplexing

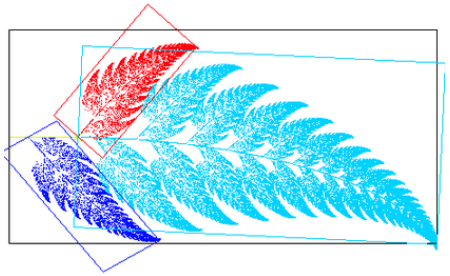


Poisson model: Assumes exponential inter-arrival times

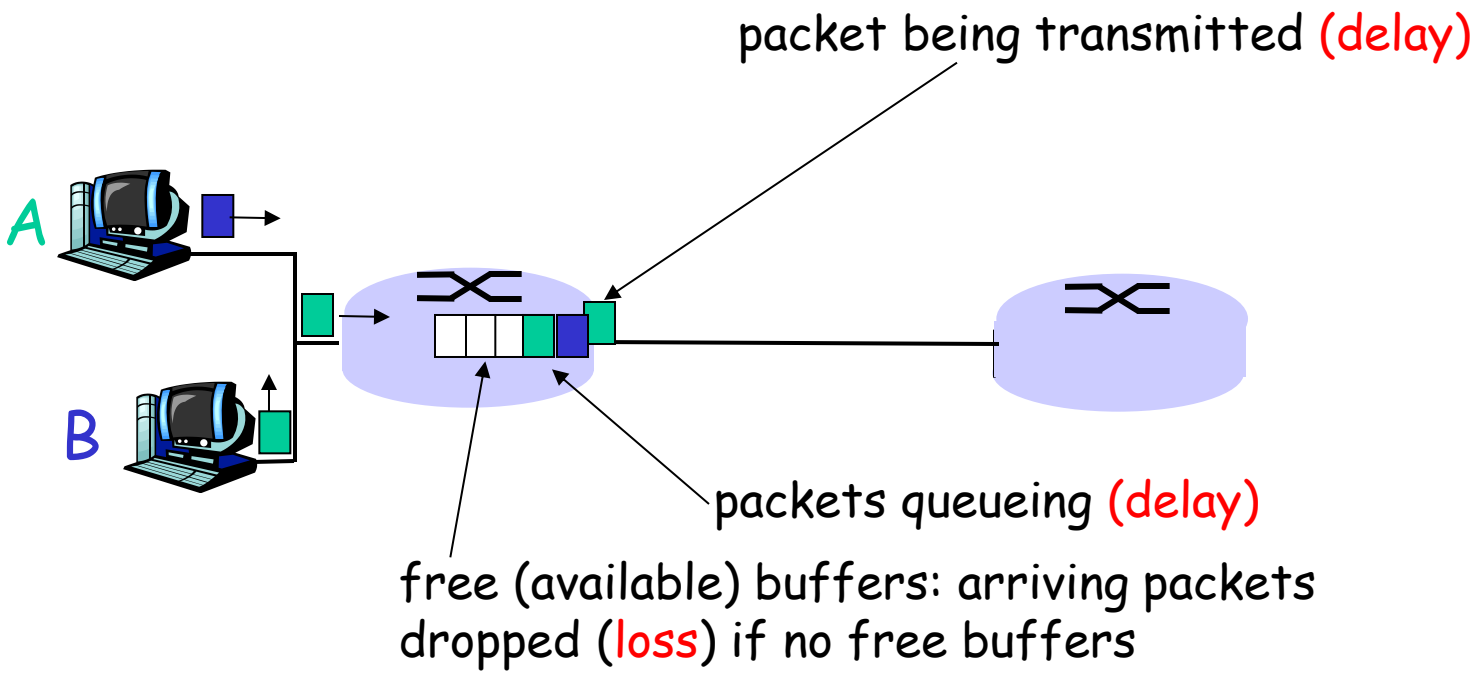


Measured: Bursty + Self similar

- Resource sharing great for bursty traffic
  - E.g., Sequence of A & B packets does not have fixed pattern - ***statistical multiplexing***.
  - In contrast: In TDM, each host gets same slot in revolving TDM frame.



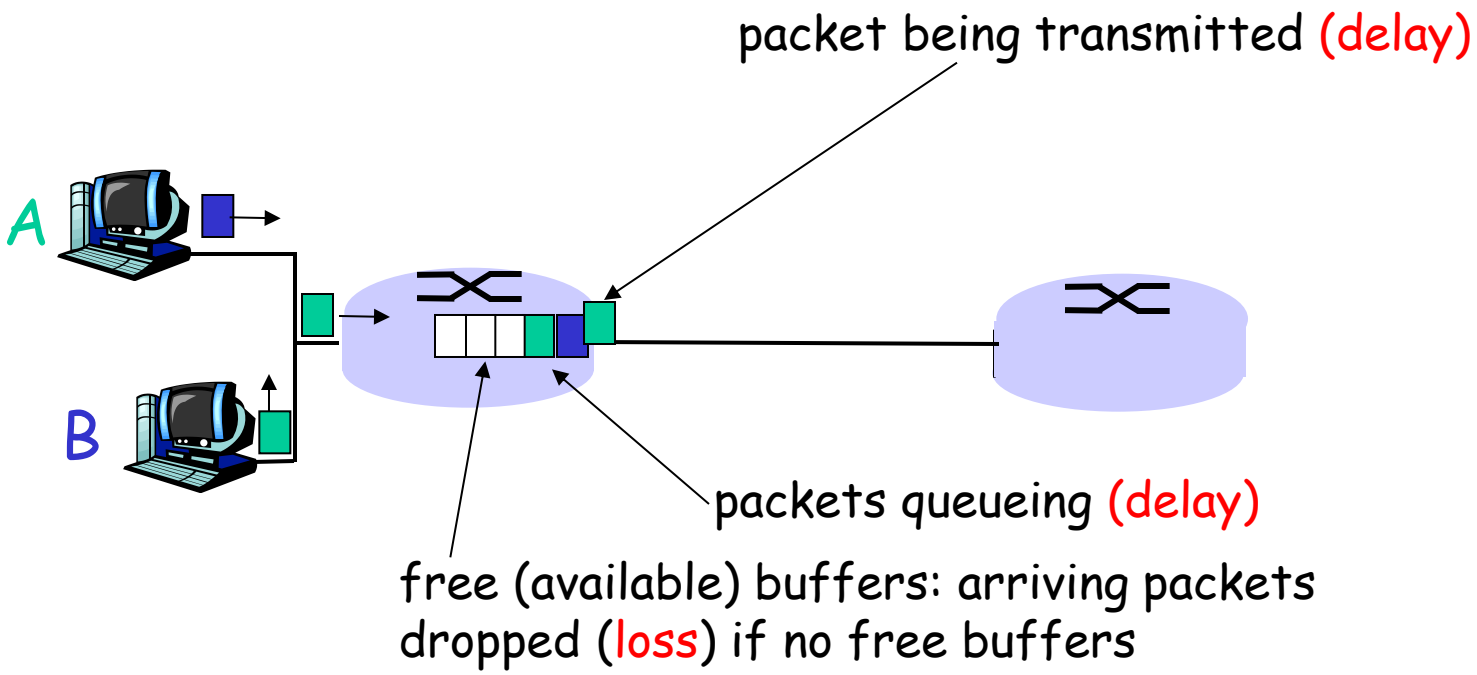
How do loss and delay occur?



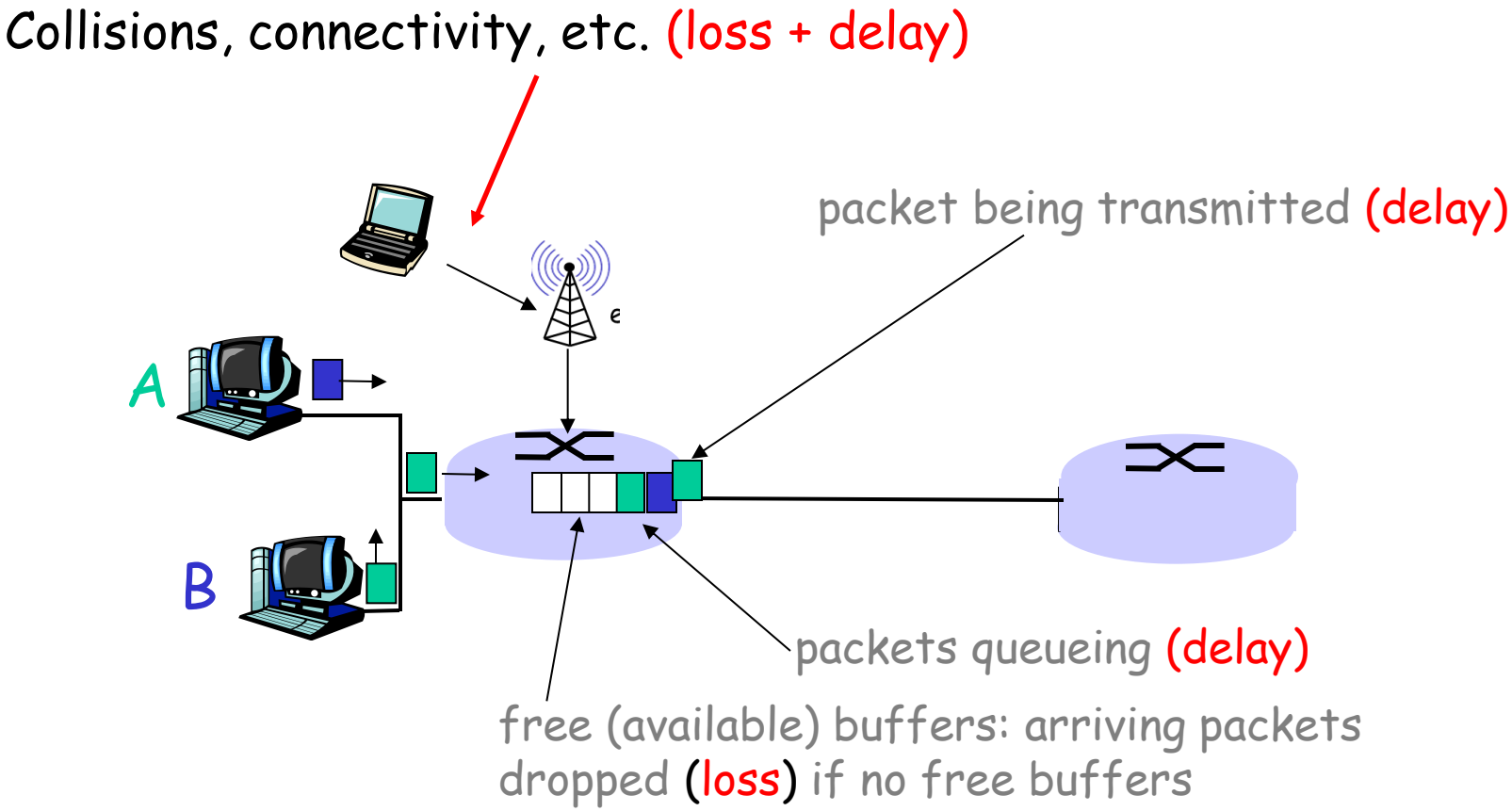
How do loss and delay occur?

packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn
- if queue is full, arriving packets dropped (Drop-Tail)



How do loss and delay occur?



Wireless, mobility: impact on higher layer protocols

- logically, impact *should* be minimal …
  - Best-effort service model remains unchanged
  - TCP and UDP can (and do) run over wireless, mobile
- … but performance-wise:
  - packet loss/delay due to bit-errors (discarded packets, delays for link-layer retransmissions), handoffs from mobility and transient connectivity
  - TCP interprets loss as congestion, will decrease congestion window un-necessarily
  - delay impairments for real-time traffic
  - limited bandwidth of wireless links

Connection oriented or not?

Connection oriented:

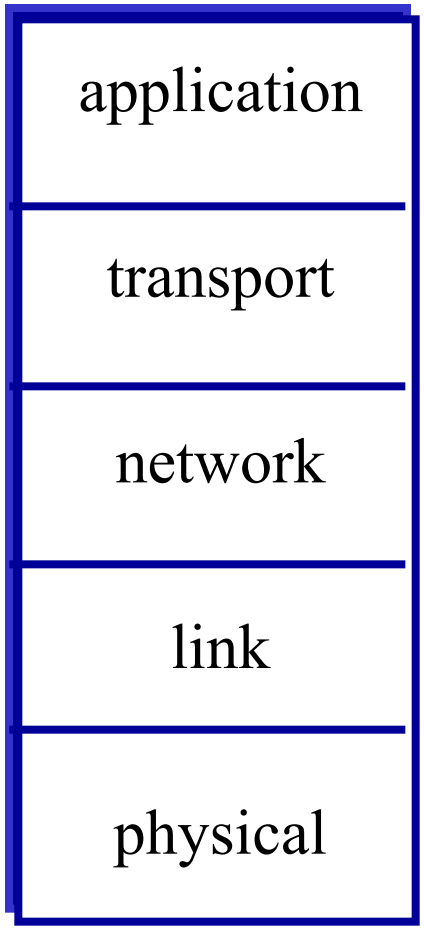
- *Hand shaking*
  - *Explicit setup phase for logical connection*
  - *Connection release afterwards*
- *Establishes state information about the connection*
- *Mechanisms for*
  - *reliable data transfer, error control, flow control, etc.*
- *Guarantees that data will arrive (eventually)*

Connection less:

- No handshaking
- No (significant) state information (at end points or in network)
- No guarantees of arrival (or when)
- No mechanisms for flow control etc.
- Simpler (and faster?)

Which is the best?
… It depends on (i) what it is used for, and (ii) what it is built on-top of …

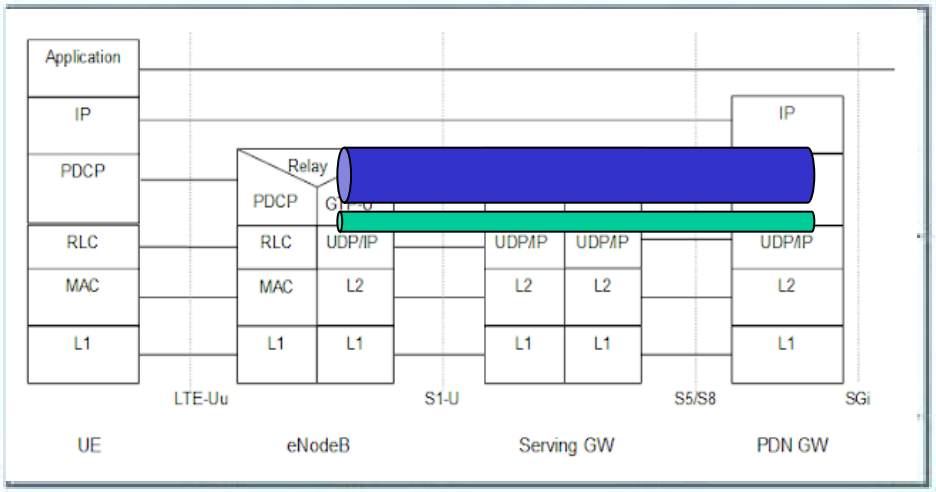
Internet protocol stack



E.g., TCP (CO)
UDP (CL)

E.g., IP (CL)

E.g., Ethernet (CL)



May use different bearers and/or treat applications differently
(e.g., based on QoS requirements)

Physical layer:

- Guided (e.g., coaxial cable, fiber, etc) vs. unguided (satellite, wireless, etc.)
- Signaling, modulation, encoding, etc,