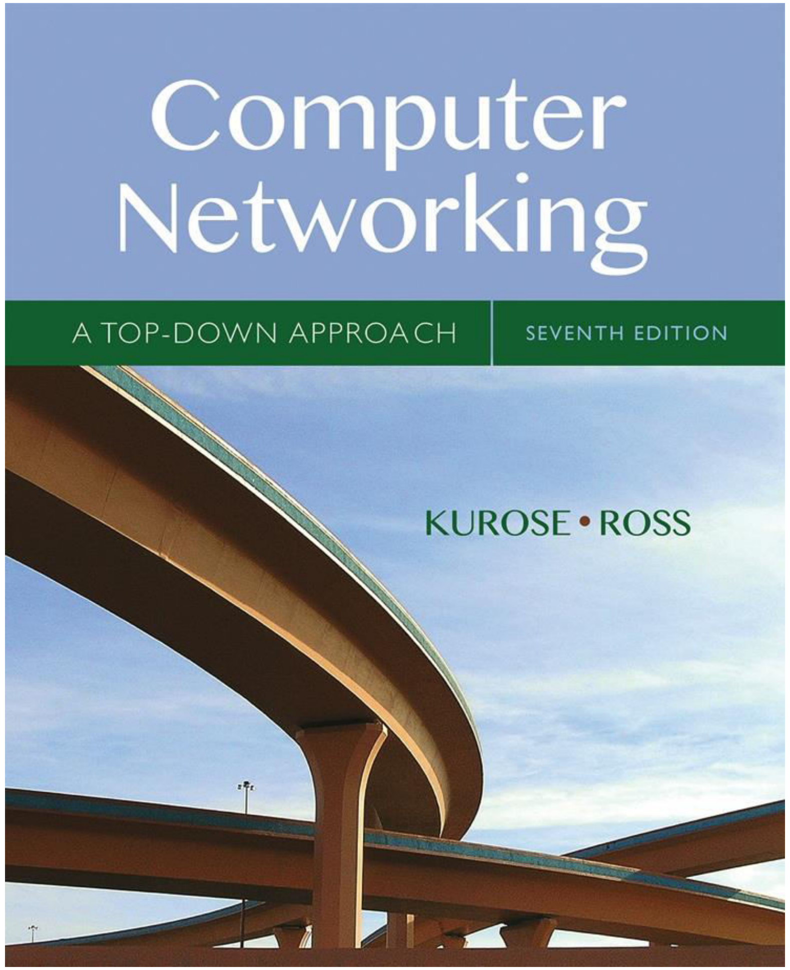
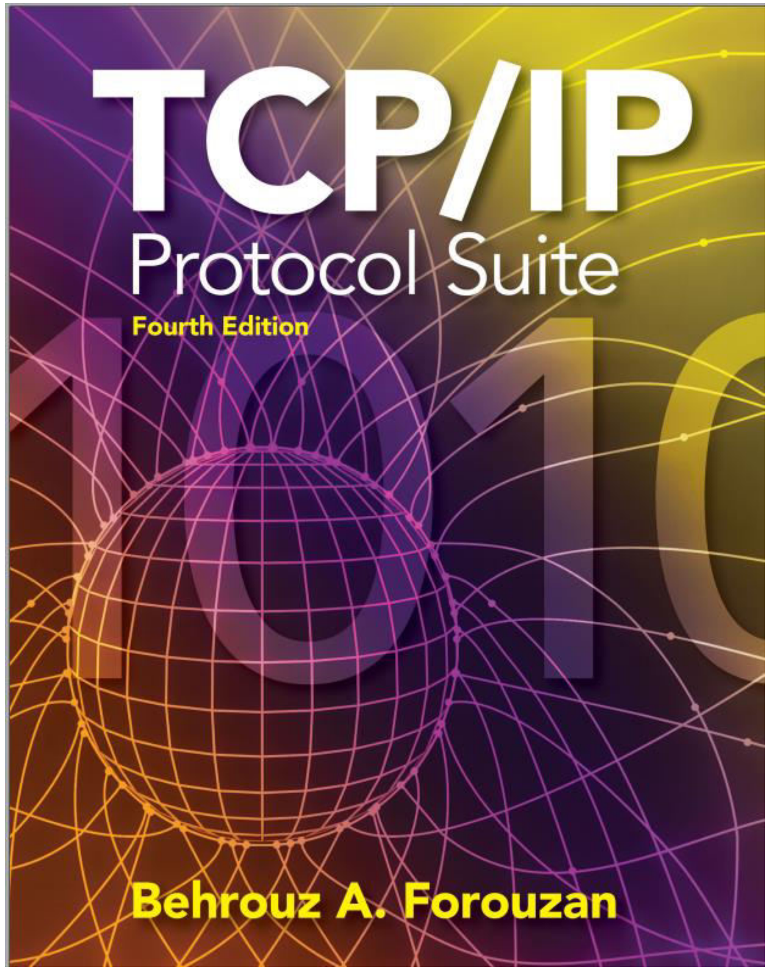
COMPUTER NETWORKS

S Nagasundari
Department of Computer Science and Engineering

Text Book





Slides adapted from

Computer Networking: A Top-Down Approach
Jim Kurose, Keith Ross
Pearson, 2017, 8th Ed.

TCP/IP protocol suite ,
Behrouz A. Forouzan.,4th Ed.

COMPUTER NETWORKS

Link Layer and LAN

S Nagasundari
Department of Computer Science and Engineering

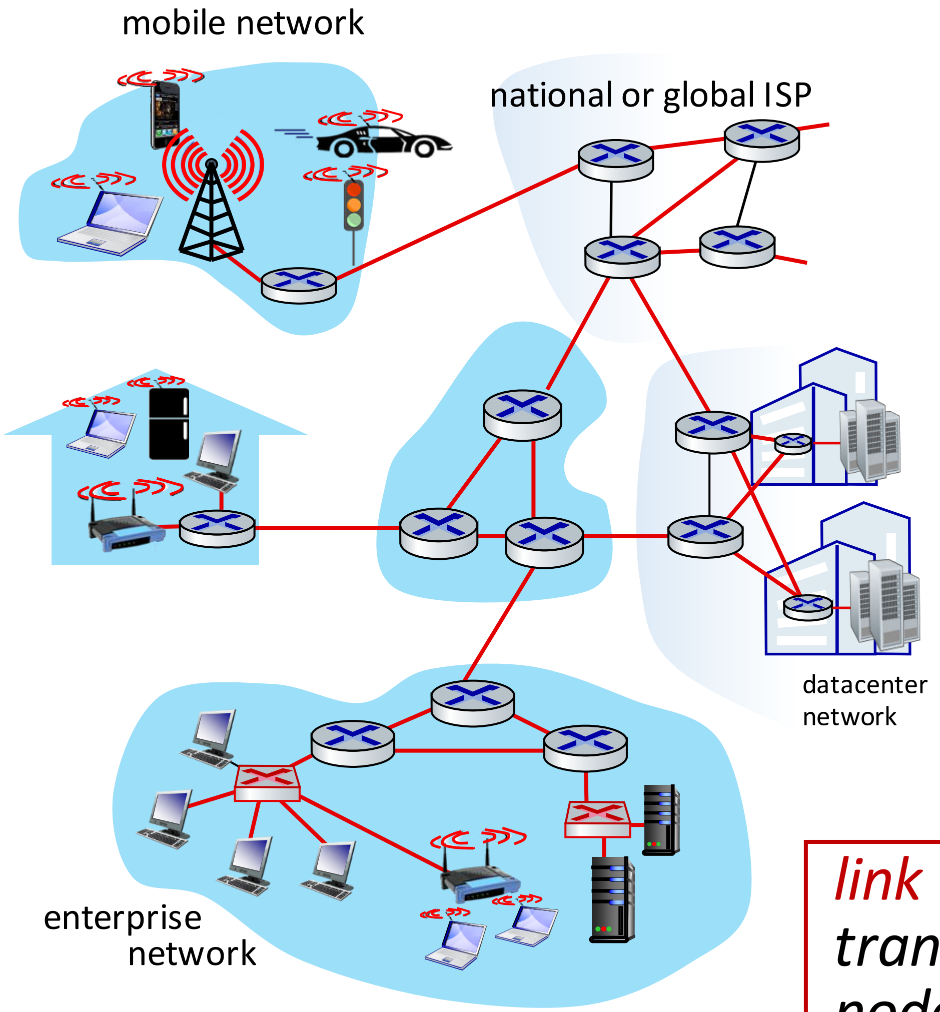
Unit – 5 Link Layer and LAN Roadmap

- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
- Physical layer
- Wireless LANs: IEEE 802.11
- A day in the life of a web request

Class 47 : Intro to Link layer : Learning Objectives

- Introduction to link layer
- Error detection and correction techniques
  - Parity Checks
  - Internet Checksum
  - Cyclic Redundancy Check

Introduction to Link layer



Terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram

link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

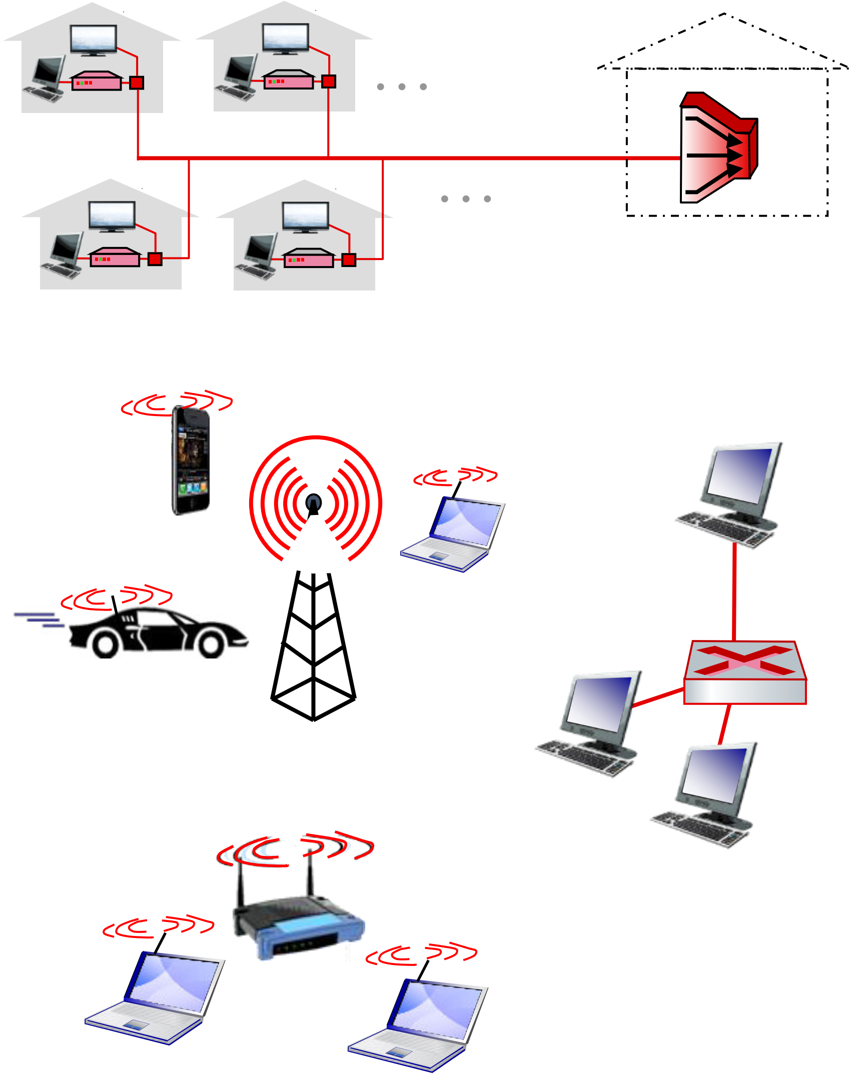
Link layer : context

- Datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link
- Each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

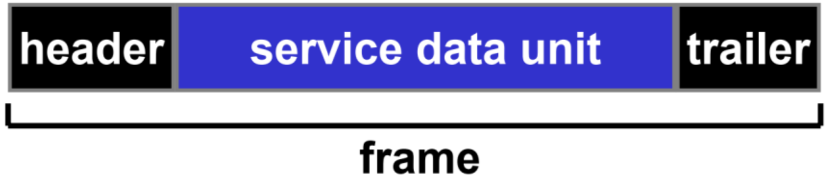
transportation analogy:

- trip from Mysore to Jaipur
  - Car: Mysore to Bangalore
  - plane: Bangalore to Delhi
  - train: Delhi to Jaipur
- tourist = datagram
- transport segment = communication link
- transportation mode = link-layer protocol
- travel agent = routing algorithm

Link layer : services

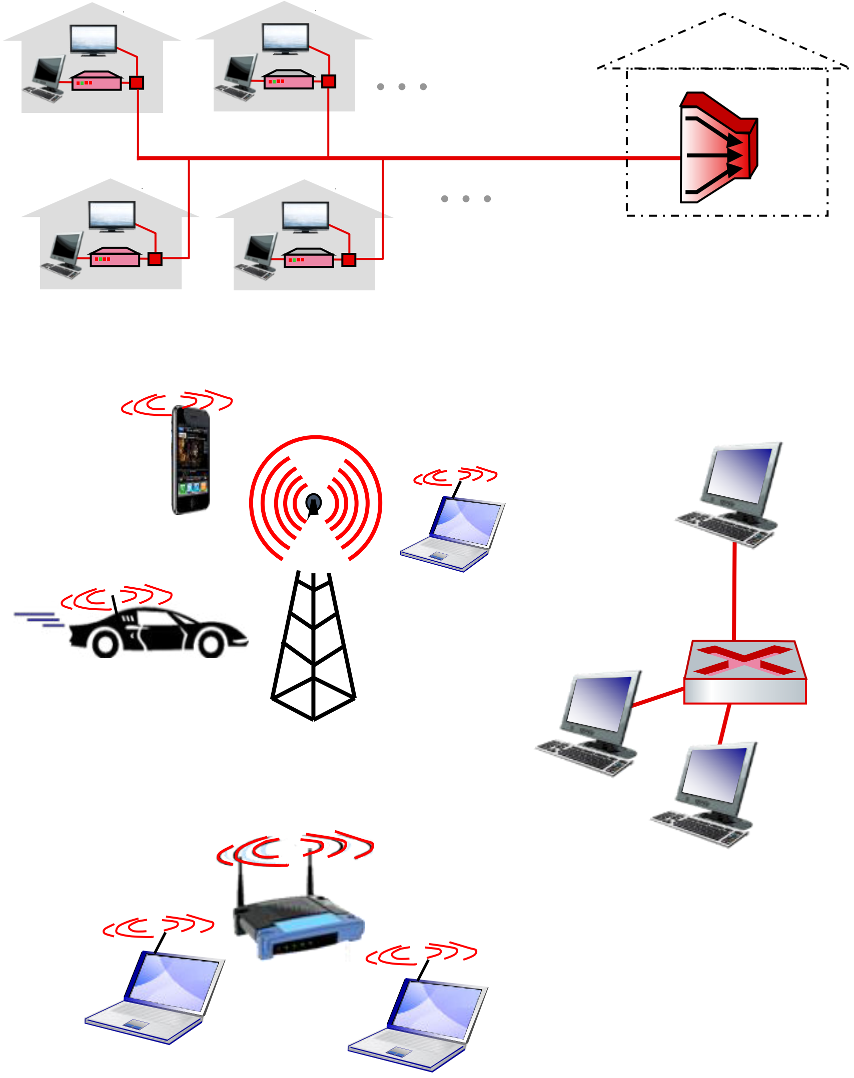


- **Framing:**
  - encapsulate datagram into frame, adding header, trailer



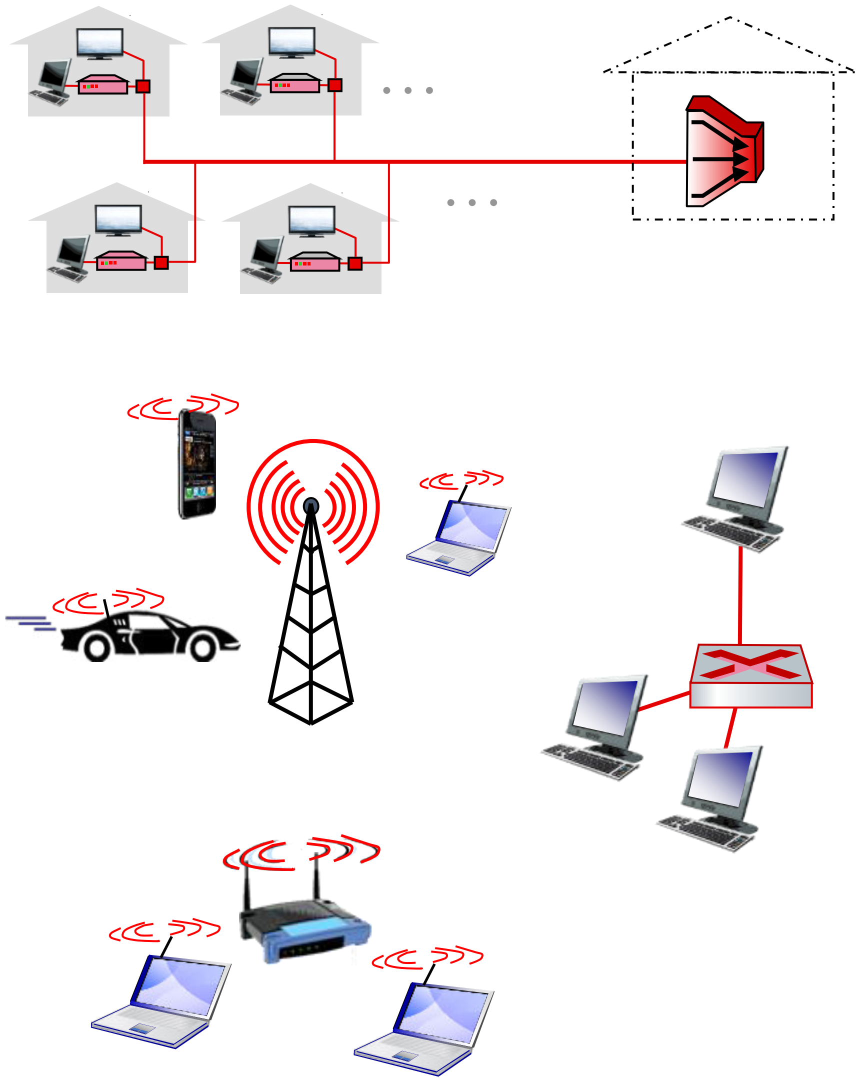
- **Link access:**
  - channel access if shared medium
  - "MAC" addresses in frame headers identify source, destination (different from IP address!)

Link layer : services



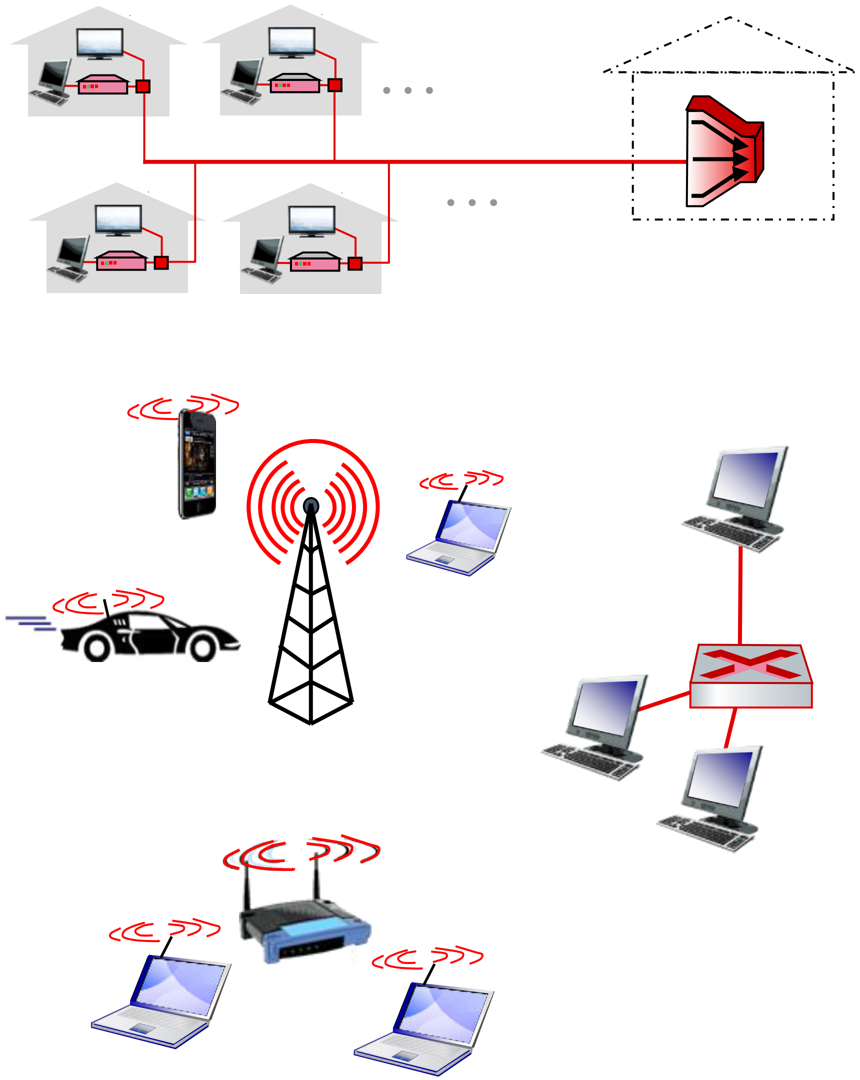
- Reliable delivery between adjacent nodes
  - we already know how to do this!
  - seldom used on low bit-error links
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link layer : services (more)



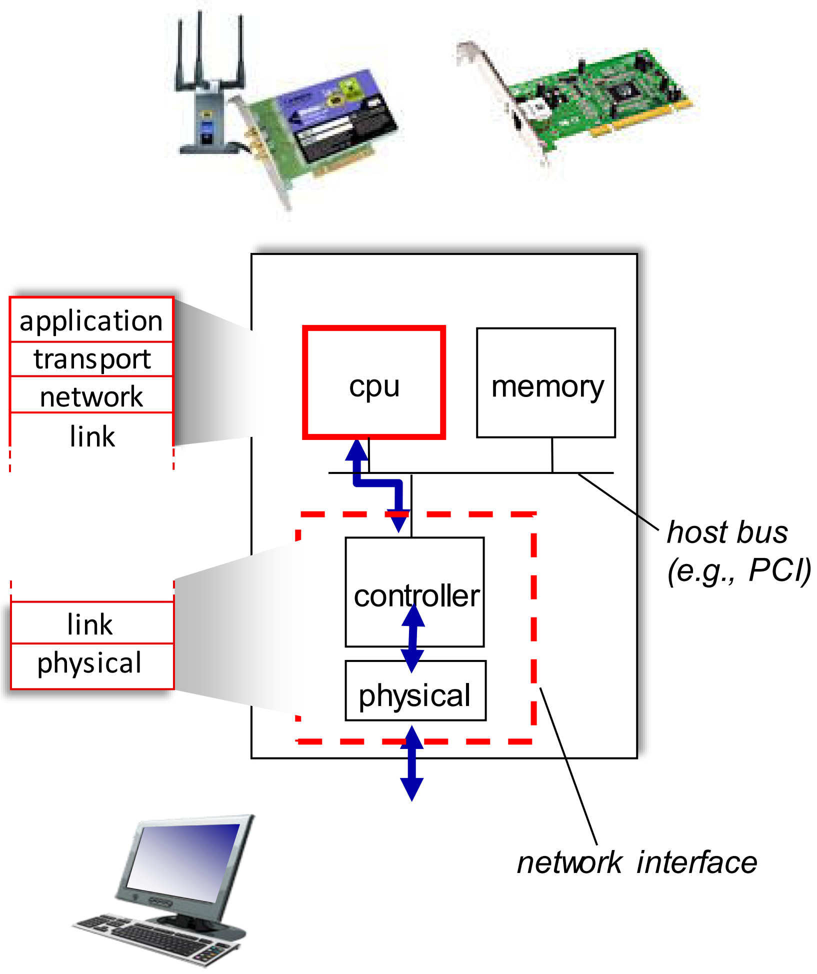
- **Flow control:**
  - pacing between adjacent sending and receiving nodes
- **Error detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame

Link layer : services (more)



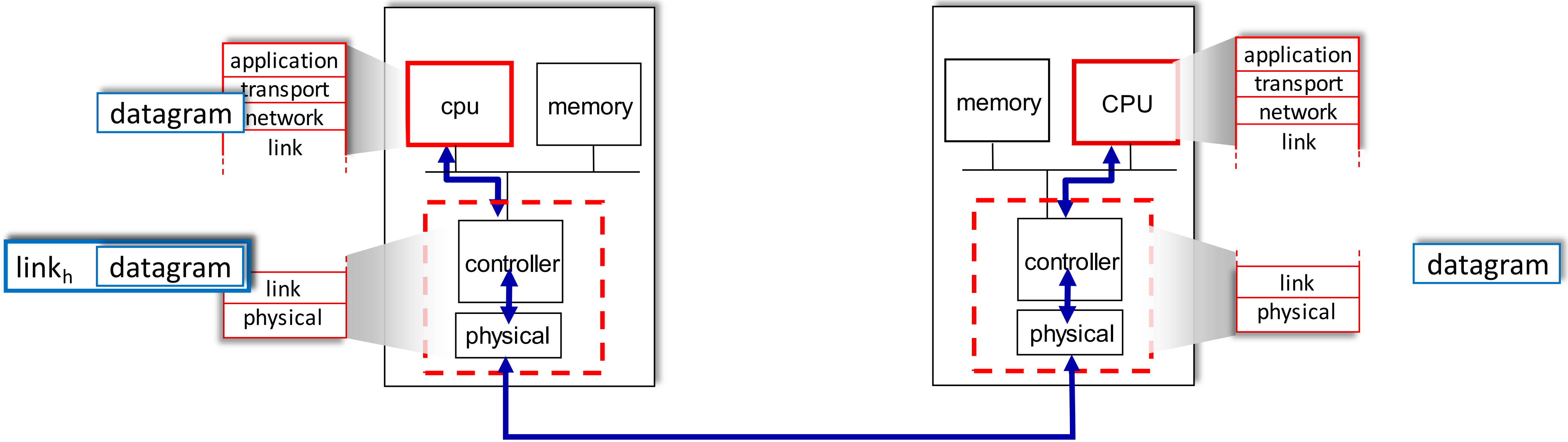
- Error correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- Half-duplex and Full-duplex:
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Where is the link layer implemented?



- In each-and-every host
- Link layer implemented in *network interface card* (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- Attaches into host's system buses
- Combination of hardware, software, firmware

Interfaces communicating



Sending side:

- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

Receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

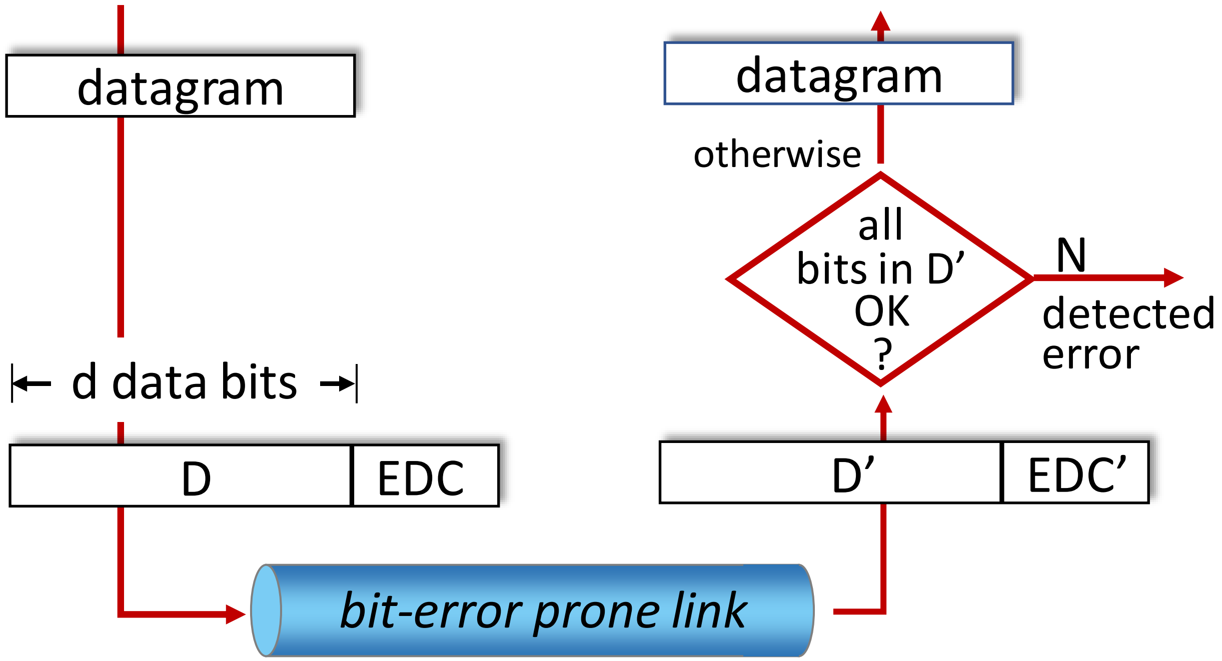
Unit – 5 Link Layer and LAN Roadmap

- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
- A day in the life of a web request
- Physical layer
- Wireless LANs: IEEE 802.11

Error detection

EDC: error detection and correction bits (e.g., redundancy)

D: data protected by error checking, may include header fields



Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Parity checking

Single bit parity:

- detect single bit errors

0111000110101011	1

d data bits — parity bit

Even parity: set parity bit so there is an even number of 1's

Two-dimensional bit parity:

- detect *and correct* single bit errors

row parity

column parity

$$
\begin{array}{cccc|c}
d_{1,1} & \cdots & d_{1,j} & & d_{1,j+1} \\
d_{2,1} & \cdots & d_{2,j} & & d_{2,j+1} \\
\cdots & \cdots & \cdots & & \cdots \\
d_{i,1} & \cdots & d_{i,j} & & d_{i,j+1} \\
\hline
d_{i+1,1} & \cdots & d_{i+1,j} & & d_{i+1,j+1}
\end{array}
$$

no errors:

$$
\begin{array}{ccccc|c}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
\hline
0 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

detected and correctable single-bit error:

$$
\begin{array}{ccccc|c}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
\hline
0 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

parity error

parity error

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

Internet checksum (review)

Goal: detect errors (*i.e.,* flipped bits) in transmitted segment

Sender:

- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment content
- checksum value put into UDP checksum field

Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - not equal - error detected
  - equal - no error detected. *But maybe errors nonetheless?* More later ....

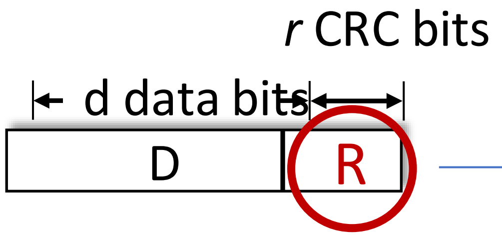
Cyclic Redundancy Check (CRC)

- More powerful error-detection coding
- D: data bits (given, think of these as a binary number)
- G: bit pattern (generator), of *r+1* bits (given)

Goal: choose *r* CRC bits, R, such that <D,R> exactly divisible by G (mod 2)

- receiver knows G, divides <D,R> by G.

  If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi)



bit pattern

$$<D,R> = D * 2^r \text{ XOR } R$$

formula for bit pattern

Cyclic Redundancy Check (CRC) : example

We want:

$$D\cdot 2^r \;\; XOR \;\; R = nG$$

or equivalently:

$$D\cdot 2^r = nG \;\; XOR \;\; R$$

or equivalently:

if we divide $D\cdot 2^r$ by G, want remainder R to satisfy:

$$R = remainder\left[\frac{D\cdot 2^r}{G}\right]$$

```
     G                  1 0 1 0 1 1
 _________          _________________
 1 0 0 1   )   1 0 1 1 1 0 0 0 0      D*2^r
               1 0 0 1
                   1 0 1
                   0 0 0
                   1 0 1 0
                   1 0 0 1
                       1 1 0
                       0 0 0
                       1 1 0 0
                       1 0 0 1
                         1 0 1 0
                         1 0 0 1
                           0 1 1
                           _____
                             R
```

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

THANK YOU

S Nagasundari
Department of Computer Science and Engineering
nagasundaris@pes.edu