

Chapter 5

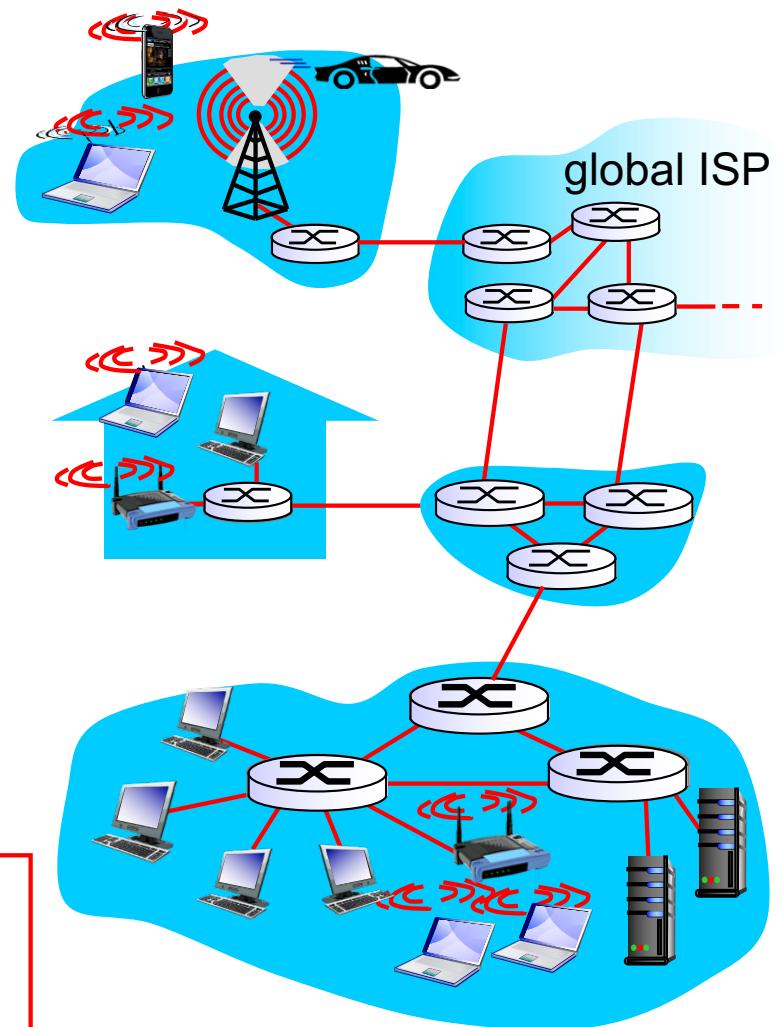
Link Layer

Link layer: introduction

terminology:

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

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



Link layer services

- ❖ *framing, link access:*

- encapsulate datagram into frame, adding header, trailer
- channel access if shared medium
- “MAC” addresses used in frame headers to identify source, dest
 - different from IP address!

- ❖ *reliable delivery between adjacent nodes*

- we learned how to do this already (chapter 3)!
- seldom used on low bit-error link (fiber, some twisted pair)
- 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 presence of errors:
 - signals sender for retransmission or drops frame

- ❖ *error correction:*

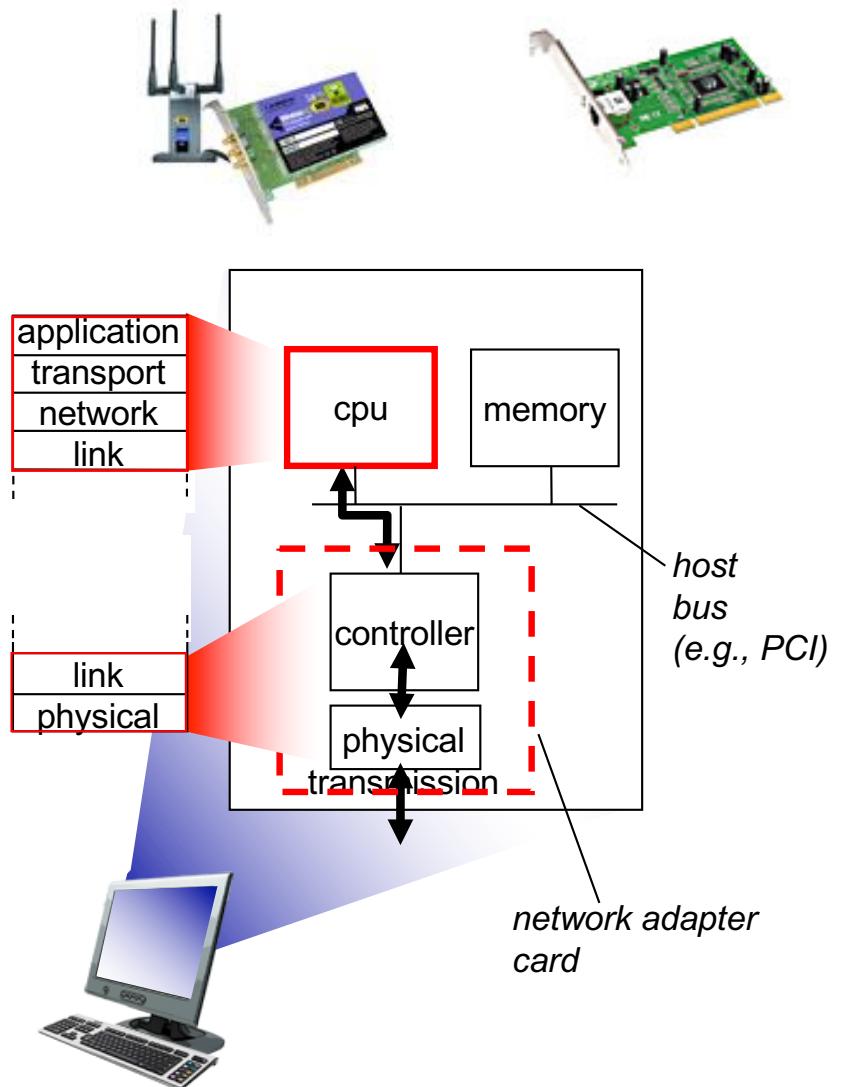
- receiver identifies *and corrects* bit error(s) without resorting to retransmission

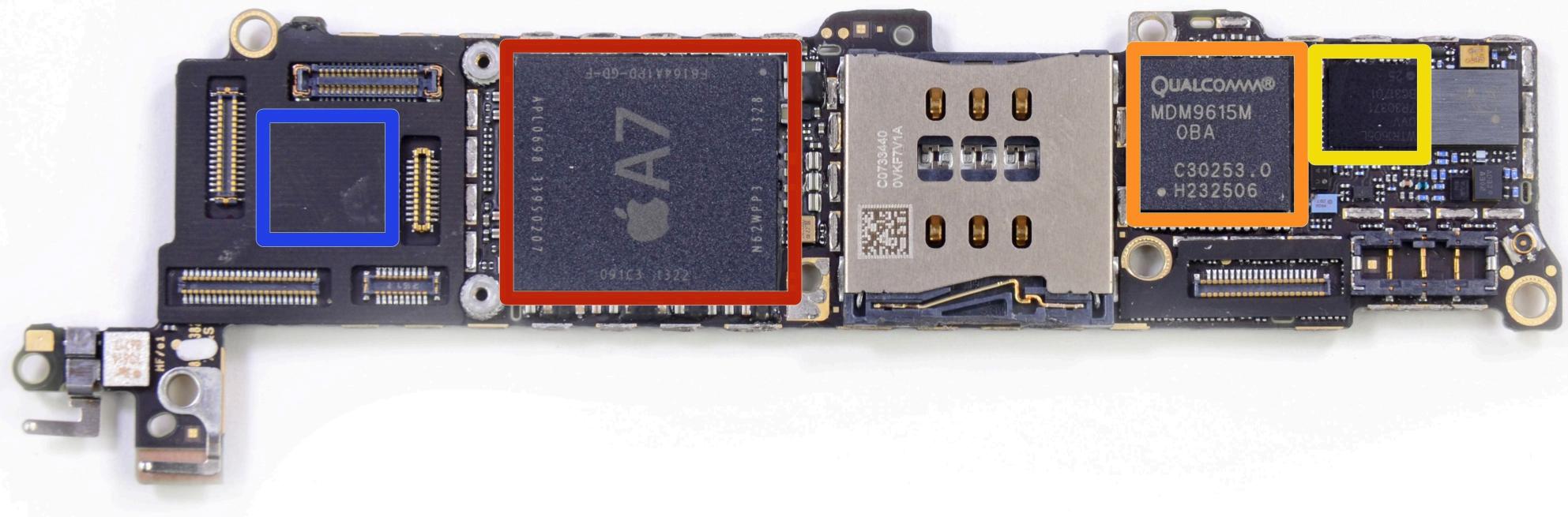
- ❖ *half-duplex and full-duplex*

- with half duplex, nodes at both ends of link can transmit, but not at same time

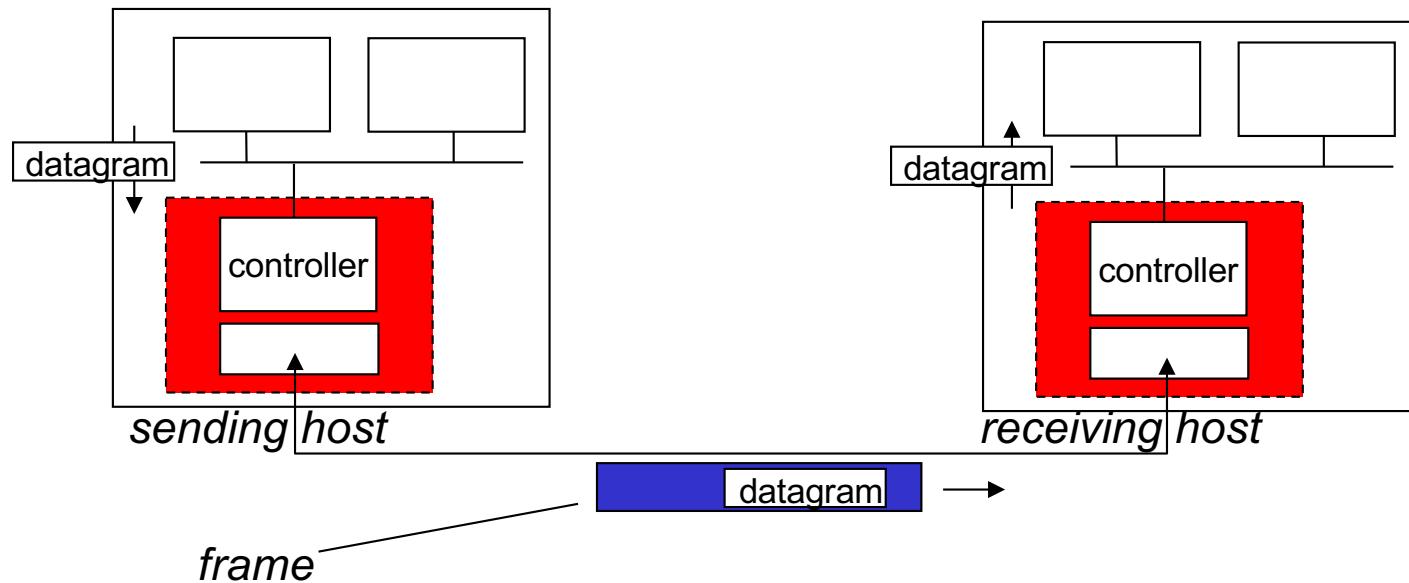
Where is the link layer implemented?

- ❖ in each and every host
- ❖ link layer implemented in “adaptor” (aka *network interface card* NIC) or on a chip
 - Ethernet card, 802.11 card; Ethernet chipset
 - implements link, physical layer
- ❖ attaches into host's system buses
- ❖ combination of hardware, software, firmware





Adaptors communicating



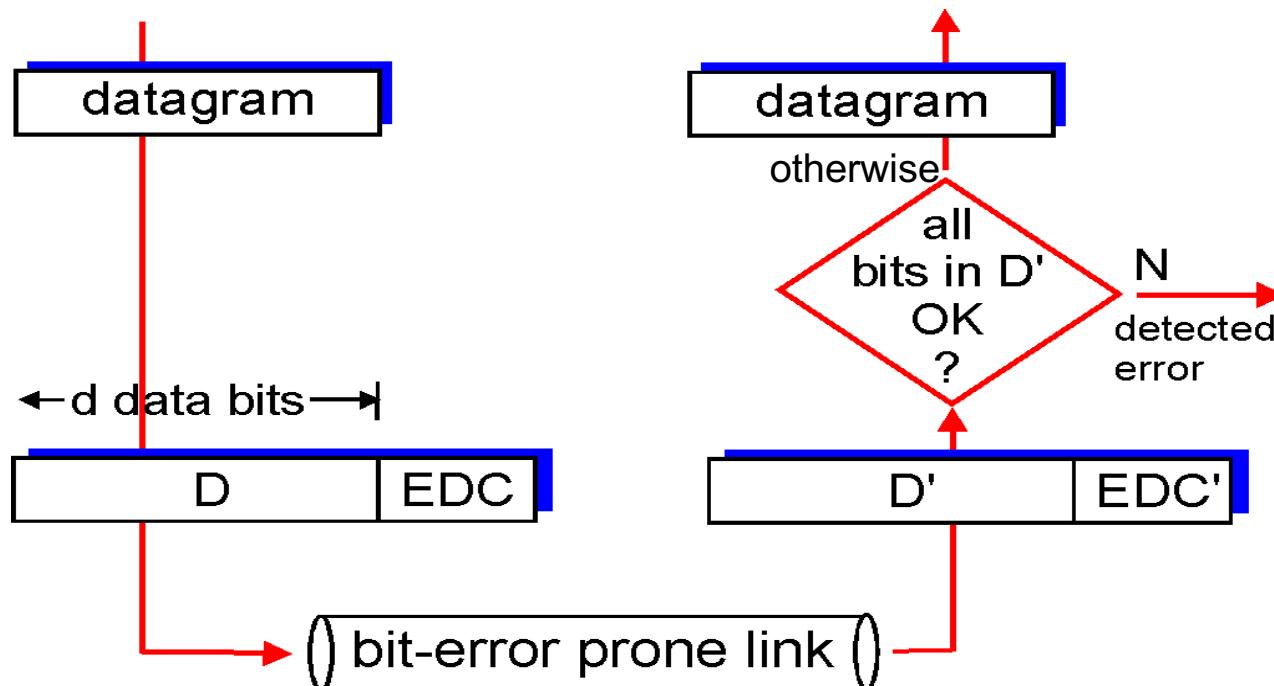
- ❖ **sending side:**
 - encapsulates datagram in frame
 - adds error checking bits, rdt, flow control, etc.
- ❖ **receiving side**
 - looks for errors, rdt, flow control, etc
 - extracts datagram, passes to upper layer at receiving side

Error detection

EDC= Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

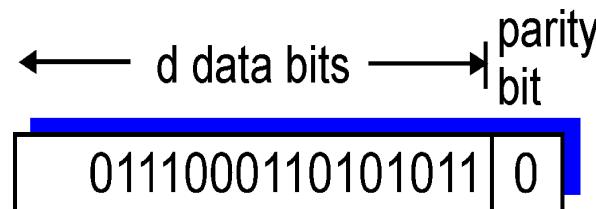
- Error detection not 100% reliable!
 - protocol may miss some errors, but rarely
 - larger EDC field yields better detection and correction



Parity checking

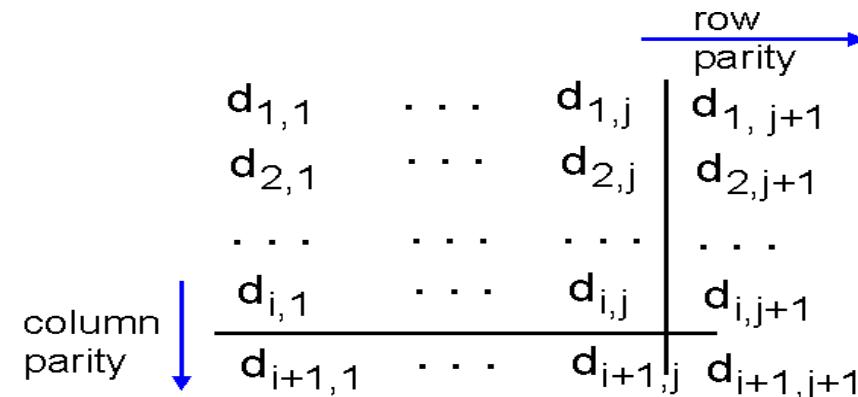
single bit parity:

- ❖ detect single bit errors



two-dimensional bit parity:

- ❖ detect and correct single bit errors



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
<hr/>					
0	0	1	0	1	0

no errors

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
<hr/>					
0	0	1	0	1	0

parity error

*correctable
single bit error*

Internet checksum (review)

goal: detect “errors” (e.g., flipped bits) in transmitted packet
(note: used at transport layer only)

sender:

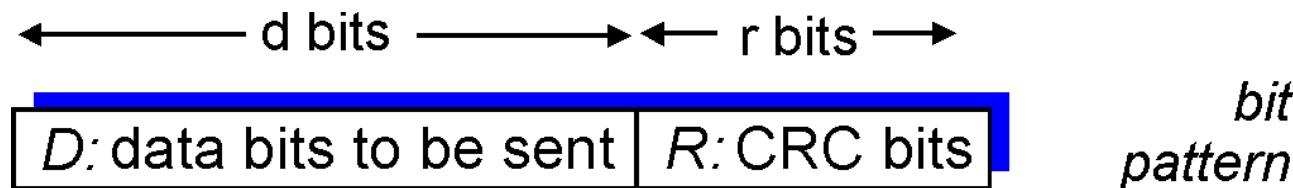
- ❖ treat segment contents as sequence of 16-bit integers
- ❖ checksum: addition (I's complement sum) of segment contents
- ❖ sender puts checksum value into UDP checksum field

receiver:

- ❖ compute checksum of received segment
- ❖ check if computed checksum equals checksum field value:
 - NO - error detected
 - YES - no error detected.
But maybe errors nonetheless?

Cyclic redundancy check

- ❖ more powerful error-detection coding
- ❖ view data bits, **D**, as a binary number
- ❖ choose $r+1$ bit pattern (generator), **G**
- ❖ goal: choose r CRC bits, **R**, such that
 - $\langle D, R \rangle$ exactly divisible by G (modulo 2)
 - receiver knows G, divides $\langle D, R \rangle$ 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, ATM)



$$D * 2^r \text{ XOR } R$$

mathematical formula