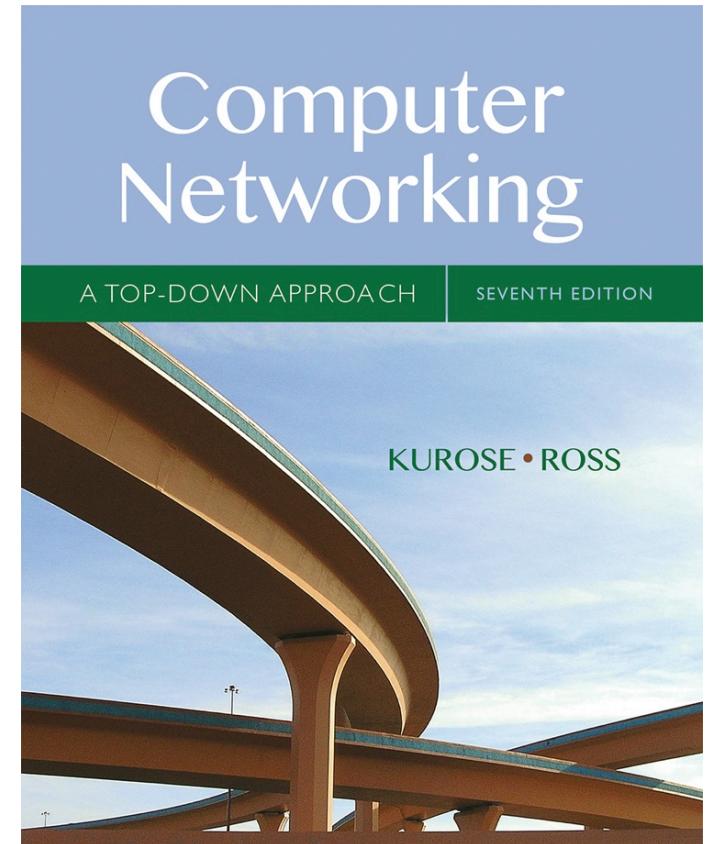


Chapter 6

The Link Layer and LANs



Computer Networking: A Top Down Approach

7th edition

Jim Kurose, Keith Ross

Pearson/Addison Wesley

April 2016

© Slides adapted from J.F Kurose and K.W. Ross
Copyright 1996-2016
All Rights Reserved

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,
correction

6.3 multiple access
protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANs

6.5 link virtualization:
MPLS

6.6 data center
networking

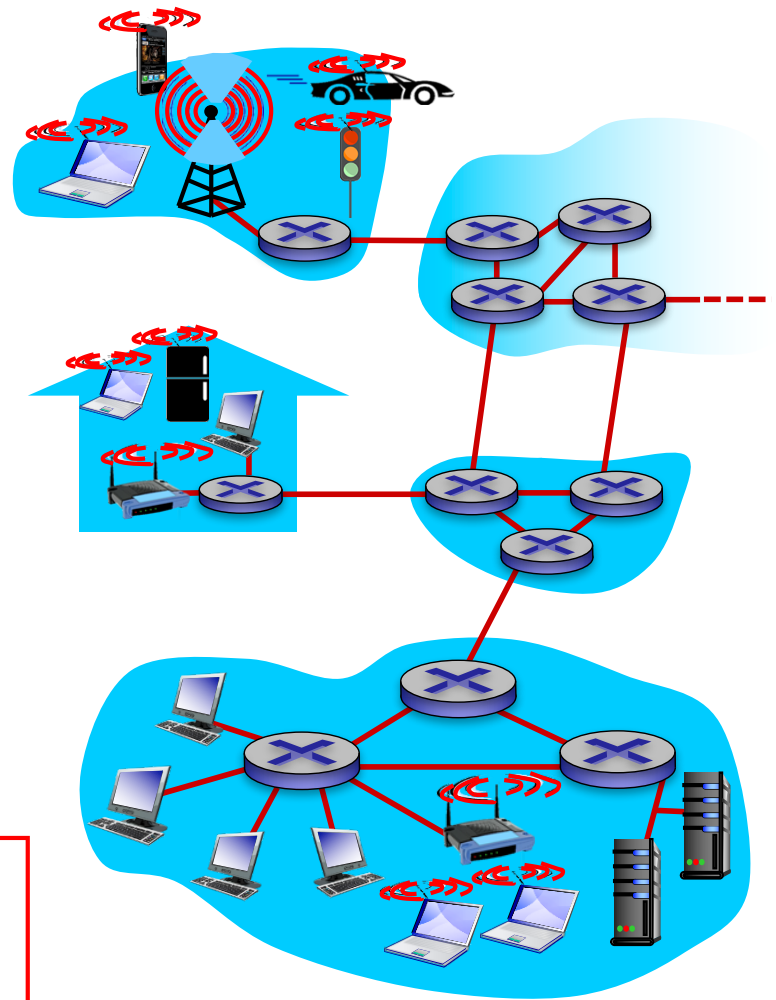
6.7 a day in the life of a
web request

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: context

- datagram transferred by different link protocols over different links:
 - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services
 - e.g., may or may not provide rdt over link

transportation analogy:

- trip from Princeton to Lausanne
 - limo: Princeton to JFK
 - plane: JFK to Geneva
 - train: Geneva to Lausanne
- tourist = **datagram**
- transport segment = **communication link**
- transportation mode = **link layer protocol**
- travel agent = **routing algorithm**

Link layer services

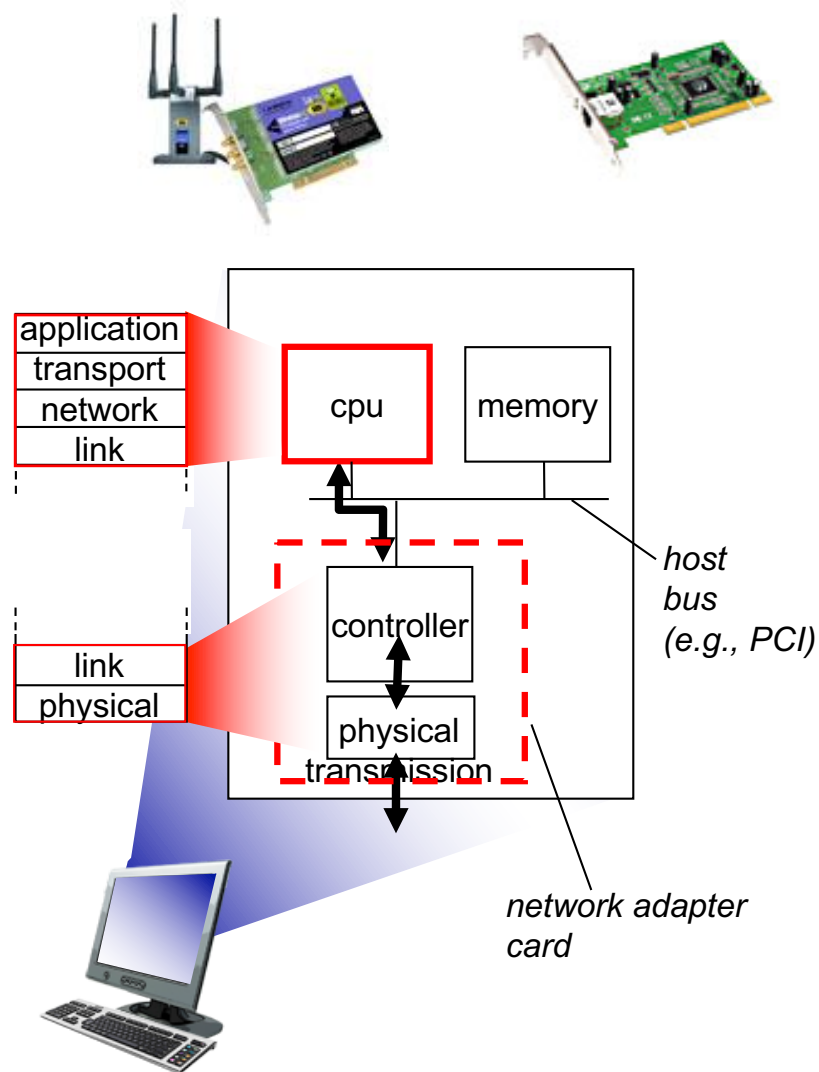
- *framing, link access:*
 - encapsulate datagram into frame, adding header, trailer
 - channel access if shared medium
 - “MAC” addresses used in frame headers to identify source, destination
 - different from IP address!
 - `ifconfig -a`
- *reliable delivery between adjacent nodes*
 - we learned how to do this already (TCP chapter 3)!
 - wireless links: high error rates
 - *Q:* why both link-level and end-end reliability?

Link layer services (more)

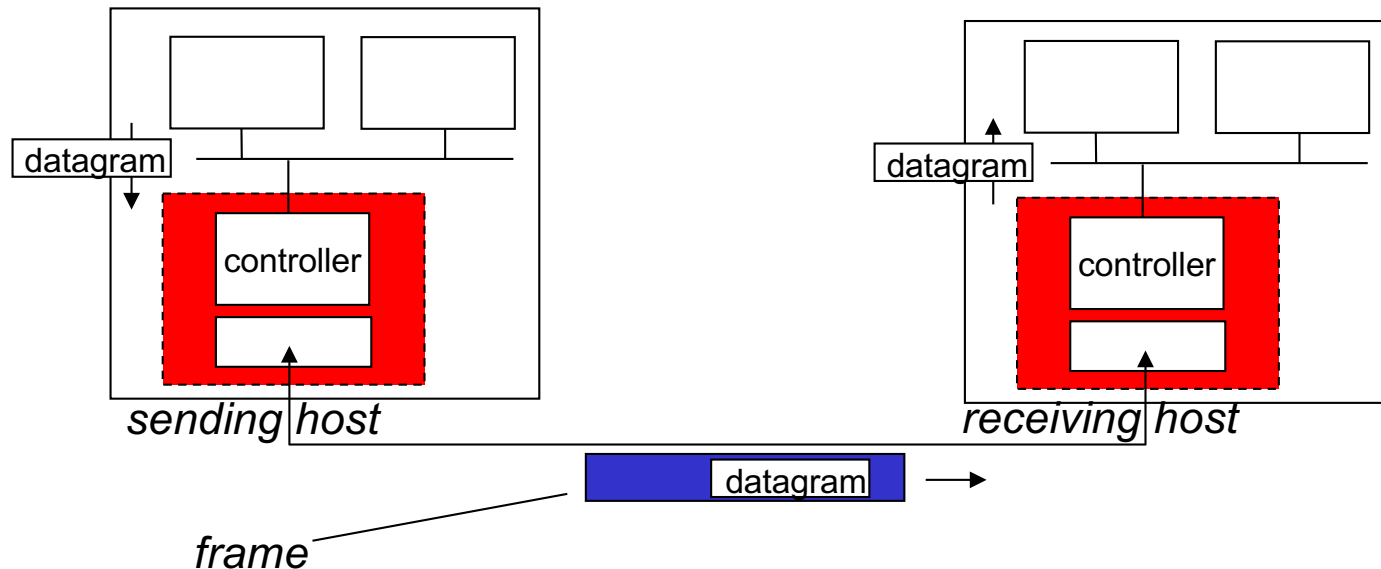
- *flow control:*
 - pacing between adjacent sending and receiving nodes
- *error detection:*
 - errors caused by signal attenuation, noise.
 - receiver detects presence of errors:
 - signals sender for retransmission or drops frame
- *error correction:*
 - receiver identifies *and corrects* bit error(s) 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
 - Controller (single chip) implements link, physical layer
- attaches into host's system buses (looks like any I/O)
- 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

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,
correction

6.3 multiple access
protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANs

6.5 link virtualization:
MPLS

6.6 data center
networking

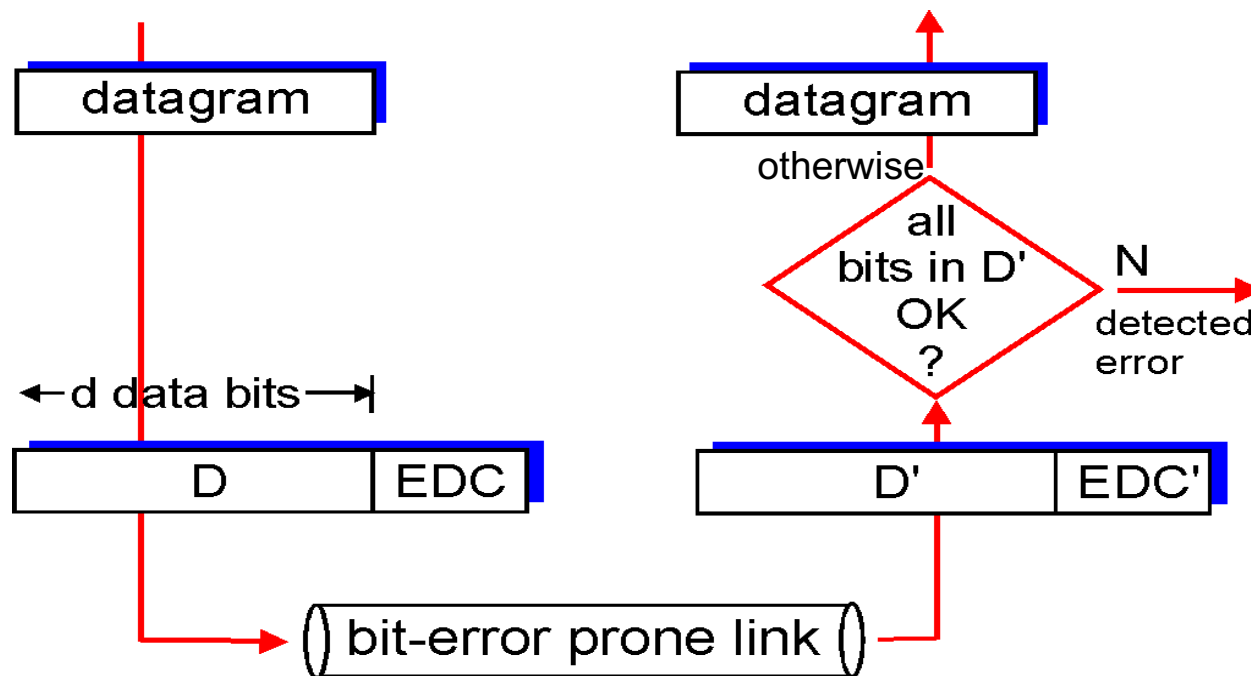
6.7 a day in the life of a
web request

Error detection

EDC= Error Detection and Correction bits (redundancy)

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

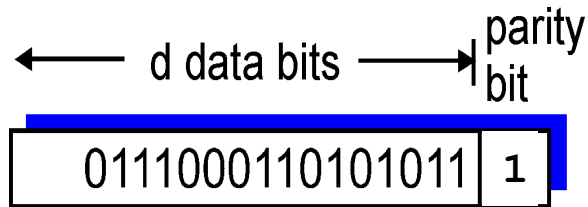
- Error detection not 100% reliable!
 - protocol may miss some errors, but rarely
 - larger EDC field yields better detection and correction



Parity checking

single bit parity:

- ❖ detect single bit errors
- ❖ example of even parity:

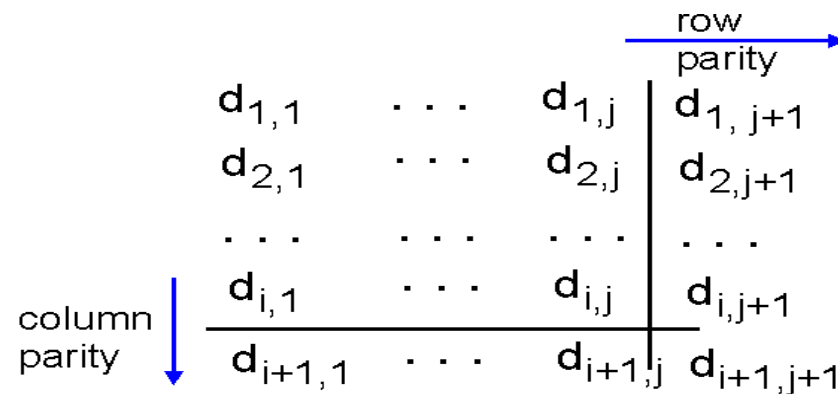


Questions:

- Can errors go undetected?
- How to deal with burst errors?
- Detection vs correction

two-dimensional bit parity:

- ❖ detect and correct single bit errors
- ❖ can detect 2 errors as well



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
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
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 (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

receiver:

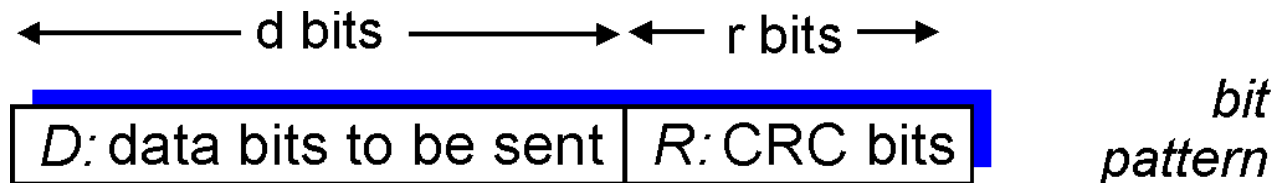
- compute checksum of received segment
- check if computed checksum equals checksum field value:
 - NO - error detected
 - YES - no error detected.
But maybe errors nonetheless?

Comments:

- ❖ Simple/fast thus implemented in software (vs. dedicated hardware for CRC)
- ❖ Low overhead but weak protection
- ❖ Checksum used in transport/IP layer, CRC implemented in link layer

Cyclic redundancy check

- more powerful error-detection coding
- **view data bits, D , as a binary number or polynomial**
- choose $r+1$ bit pattern (generator), G , starting with 1
- 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

CRC example

want:

$$D \cdot 2^r \text{ XOR } R = nG$$

equivalently:

$$D \cdot 2^r = nG \text{ XOR } R$$

equivalently:

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

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

*All operations are modulo 2 arithmetic (GF(2)) c,
without carries in addition or borrows in subtraction.*

$r = 3$

D

101110000

1001

101

000

1010

1001

110

000

1100

1001

1010

1001

011

G

1001

101000

R

CRC

- ❖ Can detect up to r burst errors, for sure.
- ❖ Can detect any odd number of bit errors.
- ❖ (Under assumptions) can detect burst of length $>r+1$ w.p. $1-0.5^r$
- ❖ The longer G , the stronger the protection, the higher the overhead and computation
- ❖ Typically
 - ❖ CRC: 32 bits, generators standardized
 - ❖ needs dedicated hardware
 - ❖ used at the link layer

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