COMP 3234B
Computer and Communication Networks

2nd semester 2023-2024

Link Layer (I)

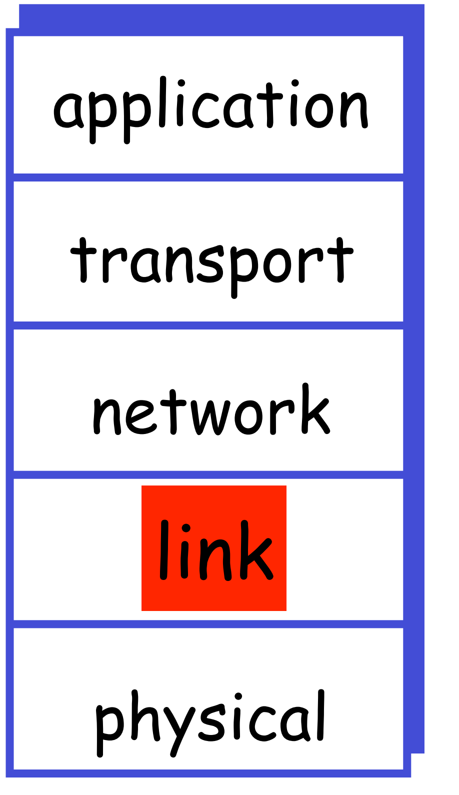
Prof. C Wu

Department of Computer Science
The University of Hong Kong

Roadmap

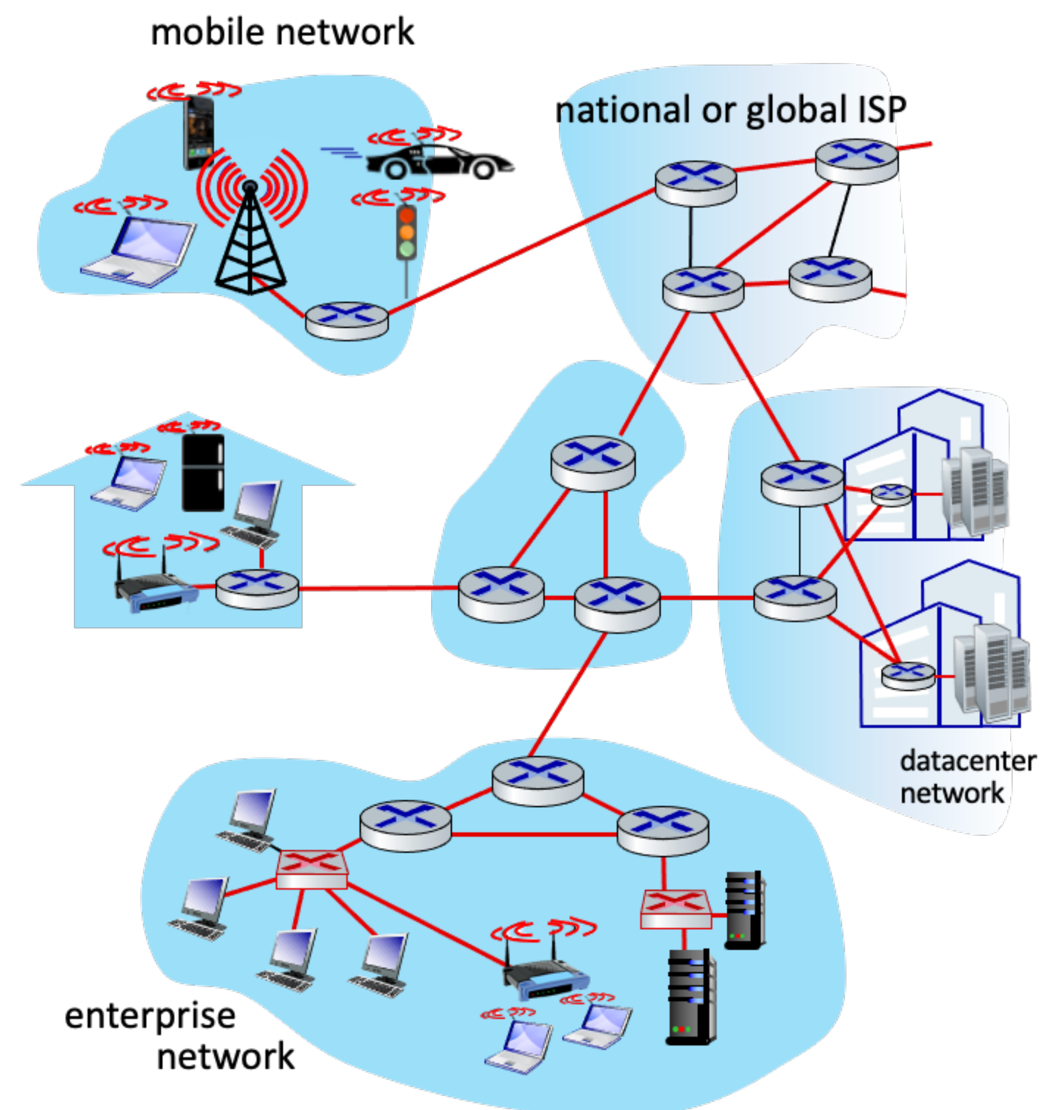
Link layer

- Principles behind link-layer services (ILO1,2,3)
  - error detection and correction
  - multiple access protocols
  - link-layer addressing
- Implementation of various link-layer technologies (ILO2,3)
  - Ethernet
  - switches
  - 802.11 wireless LAN (WiFi)
  - …



Link layer overview

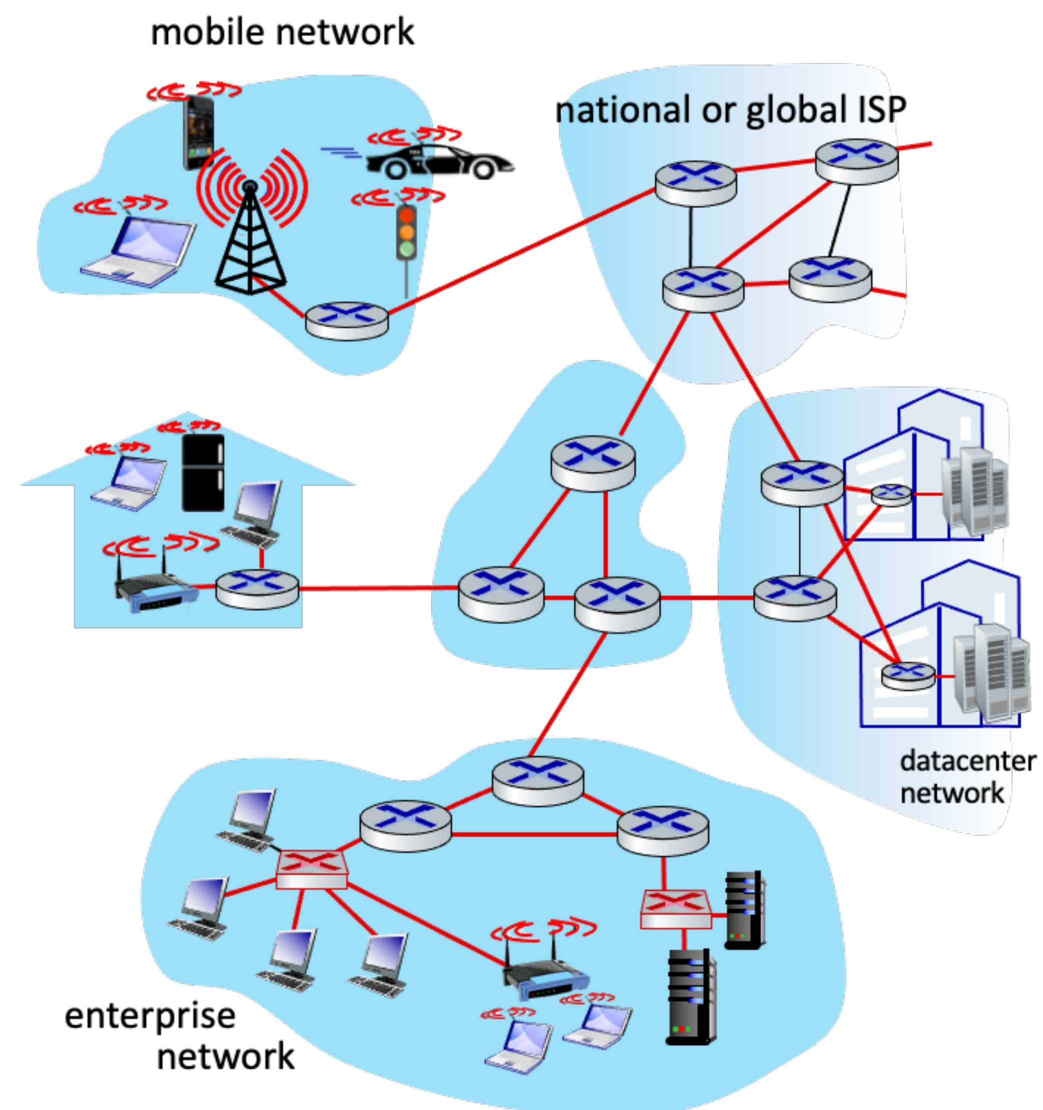
- Transferring data from one node to physically adjacent node over a link
  - node: a host or a router
  - link: communication channel that connects two adjacent nodes along a path
    - wired links
    - wireless links
  - link-layer packet is a frame, encapsulating datagram



Link layer overview (cont'd)

- Possibly different link-layer protocols on different links along an end-to-end path
  - e.g., Ethernet on the first link, frame relay on intermediate links, 802.11 WiFi on the last link
  - each link protocol defines different frame formats and provides different services

    e.g., may or may not provide rdt



- A transportation analogy
  - a trip from Hong Kong U. to Tsinghua U. :

    bus: HKU —> HK airport

    plane: HK airport—> Beijing airport

    subway: Beijing airport —>Tsinghua U.

  - 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
  - multiple access problem: how multiple nodes can share a single broadcast link
  - Medium Access Control (MAC) protocols, or multiple access protocols: to solve the multiple access problems
- Reliable delivery between adjacent nodes
  - done by acknowledgement and retransmission (ARQ)
  - Q: why need both link-level and end-end rdt?

    to correct loss and error locally
  - used on links prone to high error rates, e.g., wireless links; seldom used on low bit-error link, e.g., fiber, coax, twisted pair copper wire

Link-layer services (cont'd)

- Flow control
  - pacing between adjacent sending and receiving nodes

    nodes have limited frame buffering capacity

    prevent sending node from overwhelming receiving node

- Error detection
  - errors caused by signal attenuation, electromagnetic noise, etc.
  - receiving node detects presence of errors

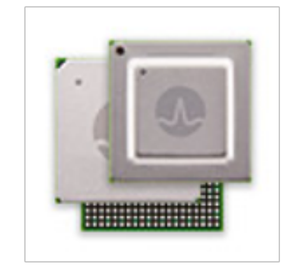
    drops frame or signals sender for retransmission

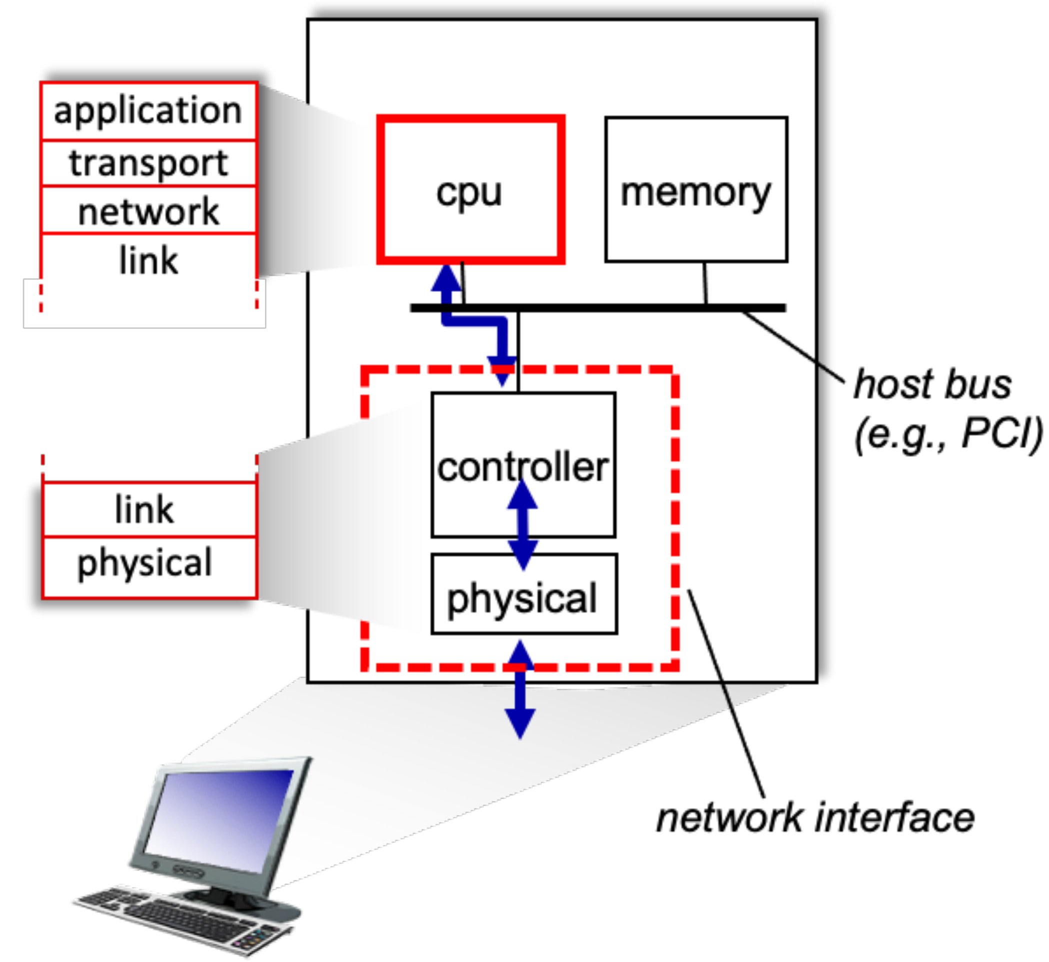
- Error correction
  - receiving node may identify and correct bit error(s) without resorting to retransmission

- Half-duplex and full-duplex
  - full duplex: nodes at both ends of link can transmit at same time
  - half duplex: nodes at both ends of link can transmit, but not at same time

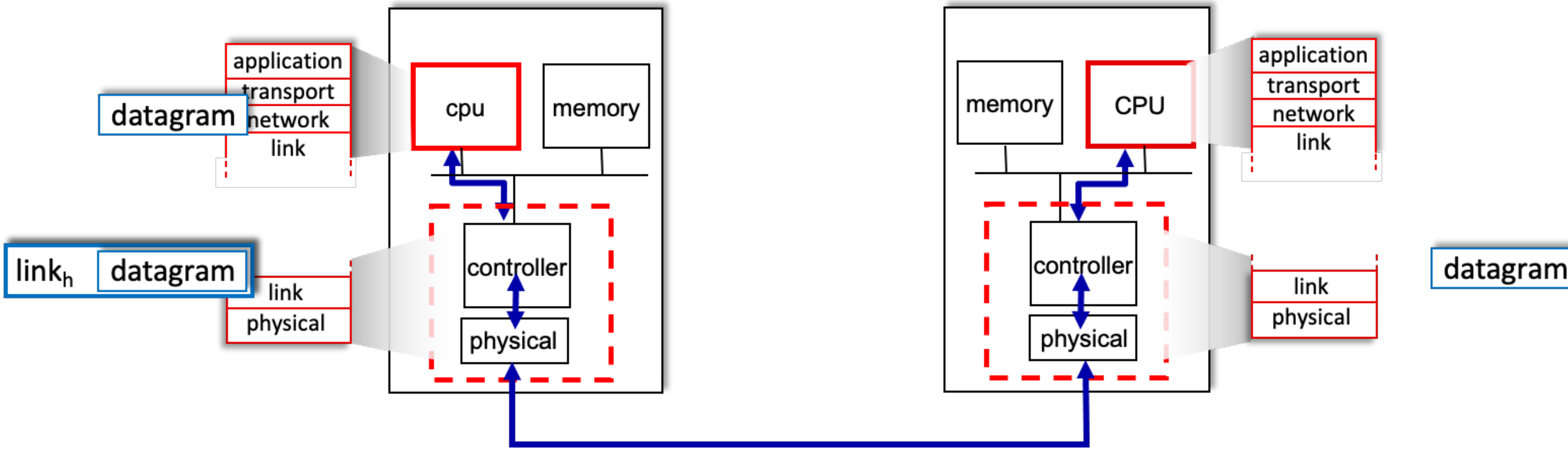
Where is link layer implemented?

- In every host, link layer is implemented in “network adaptor”, i.e., network interface card (NIC)
  - e.g., Ethernet card, 802.11 card
  - the link-layer controller implements many of the link-layer services
  - link layer is a combination of
    hardware: controller implements framing, link access, error detection, etc.
    software: runs on CPU, e.g., assembling link-layer addressing information, activating controller hardware



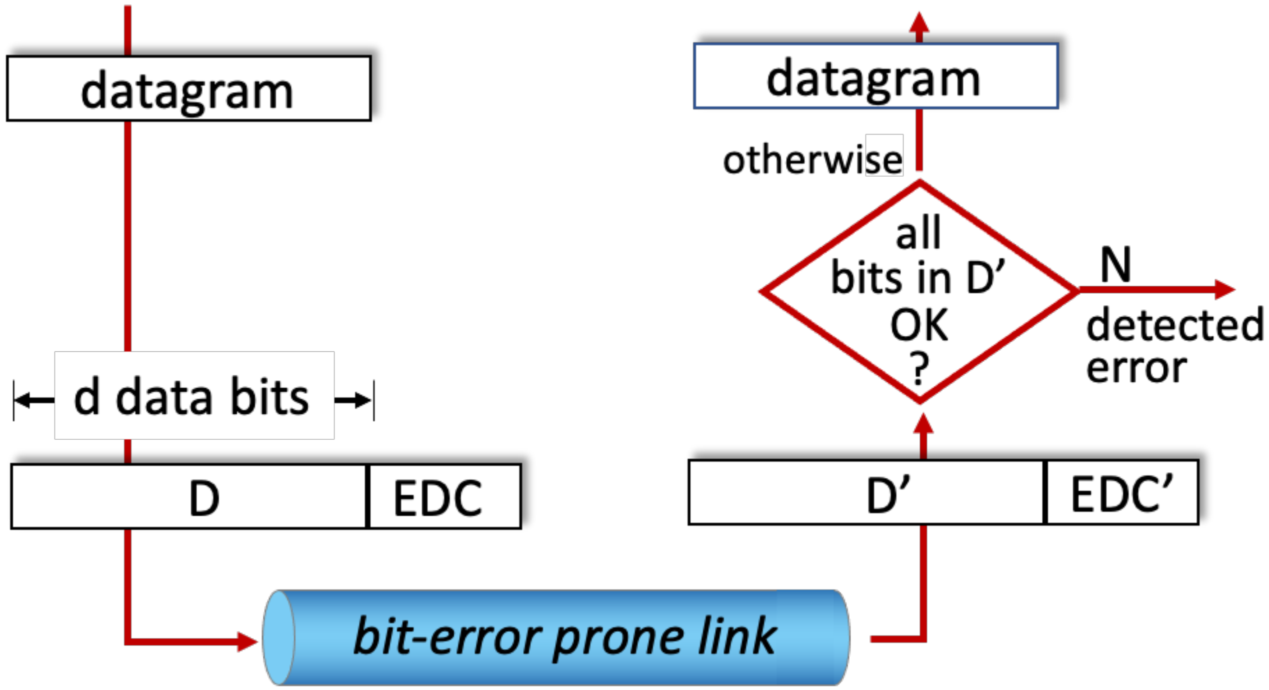


Interfaces communicating



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

Error detection and correction



- D = Data protected by error checking
  - not only datagram, but also link-layer header fields
- EDC= Error Detection and Correction bits
- Error detection: receiver decides whether D' is the same as D, using D' and EDC' it received
  - error detection techniques not 100% reliable!

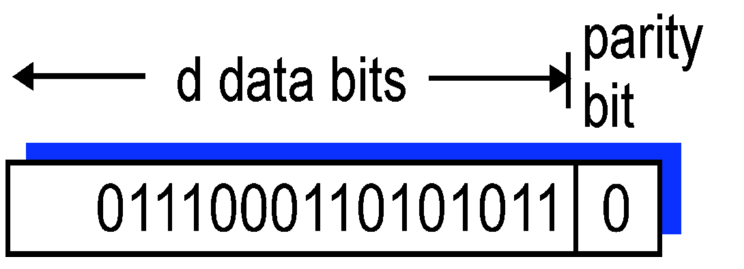
    may miss some errors (undetected bit errors), but rarely
  - larger EDC field (i.e., larger overhead) yields better detection and correction

Parity checks

- Simplest form of error detection
  - EDC = a single parity bit

    even parity scheme: the total number of 1s in the d+1 bits are even

    odd parity scheme: the total number of 1s in the d+1 bits are odd



  - Receiver: if the parity of received bits is wrong, knows some odd number of bit errors have occurred

Parity checks (cont'd)

- Two-dimensional parity
  - d bit data divided into i rows and j columns
  - a parity bit for each row, each column
  - EDC = i+j+1 parity bits
  - detect and correct single-bit error in data or parity bits
  - detect (but may not correct) two-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

Checksum (revision)

- Internet checksum
  - typically employed in transport layer: TCP, UDP

Sender:

- treat data as sequence of 16-bit integers
- checksum: addition (1's complement of the sum) of data
- sender puts checksum value into checksum field of the packet

Receiver:

- compute checksum of received packet
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected (But maybe errors nonetheless)

- relatively simple and fast, but may not detect all errors

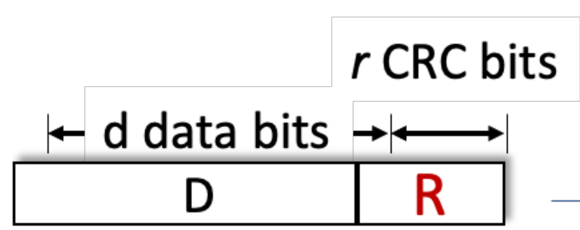
CRC

- Cyclic Redundancy Check codes
  - more powerful error detection coding
  - view d-bit data, D, as a binary number
  - given r+1 bit pattern (*generator*), G

    the first bit is always 1
  - decide r CRC bits, R, such that <D,R> exactly divisible by G (modulo 2)
  - receiver knows G,

    divides received <D,R> by G, error detected if non-zero remainder

    <=> divides received $D \cdot 2^r$ by G, error detected if remainder is not R



bits to send

<D,R> = $D * 2^r$ XOR R — formula for these bits

e.g., D=101110, R=011, <D,R>=101110011

G=1001

CRC

multiplication/division: same as in base-2 arithmetic
but all without carries in addition and borrows in subtraction
<=> addition = subtraction = bitwise XOR

How to derive R

- want:

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

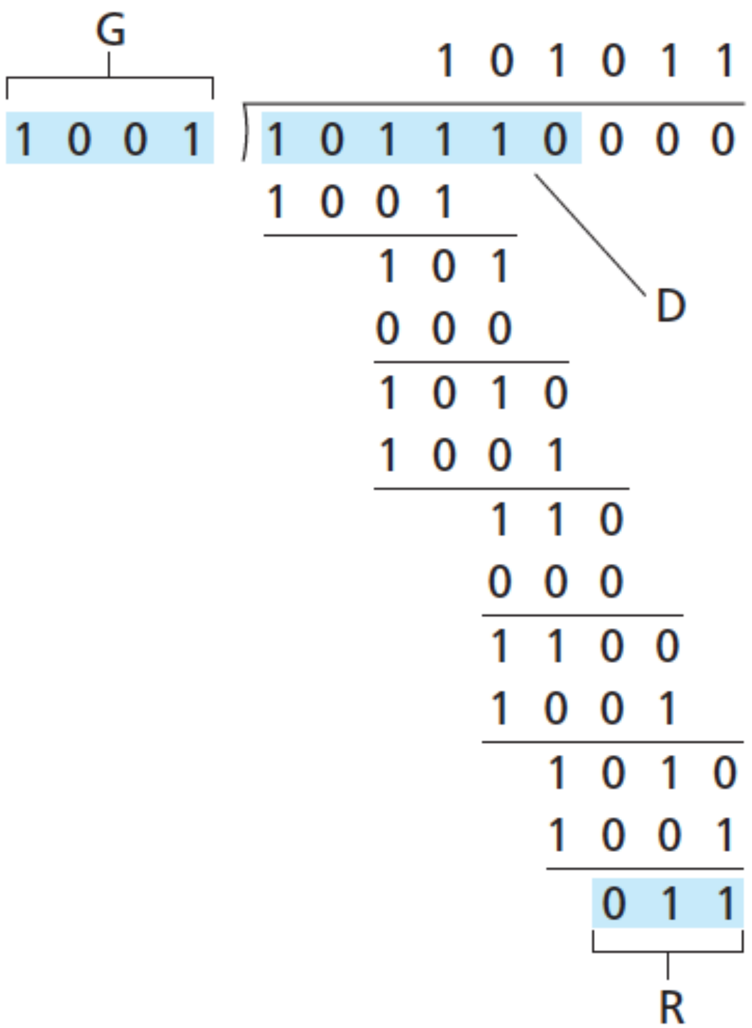
- equivalently (by XOR R to both sides of th
  above equation):

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

- equivalently:

  if we divide $D \cdot 2^r$ by G, remainder is R

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



- CRC can detect all burst errors less than r+1 bits (consecutive bit errors of r bits or fewer) and any odd number of bit errors
- widely used in practical link-layer protocols (e.g., Ethernet, 802.11 WiFi)
- generators with r=8, 12, 16, 32 defined

- Required reading:
  - *Computer Networking: A Top Down Approach* (8th Edition)

    Ch 6.1, 6.2

- Acknowledgement:
  - Some materials are extracted from the slides created by Prof. Jim F. Kurose and Prof. Keith W. Ross for the textbook.