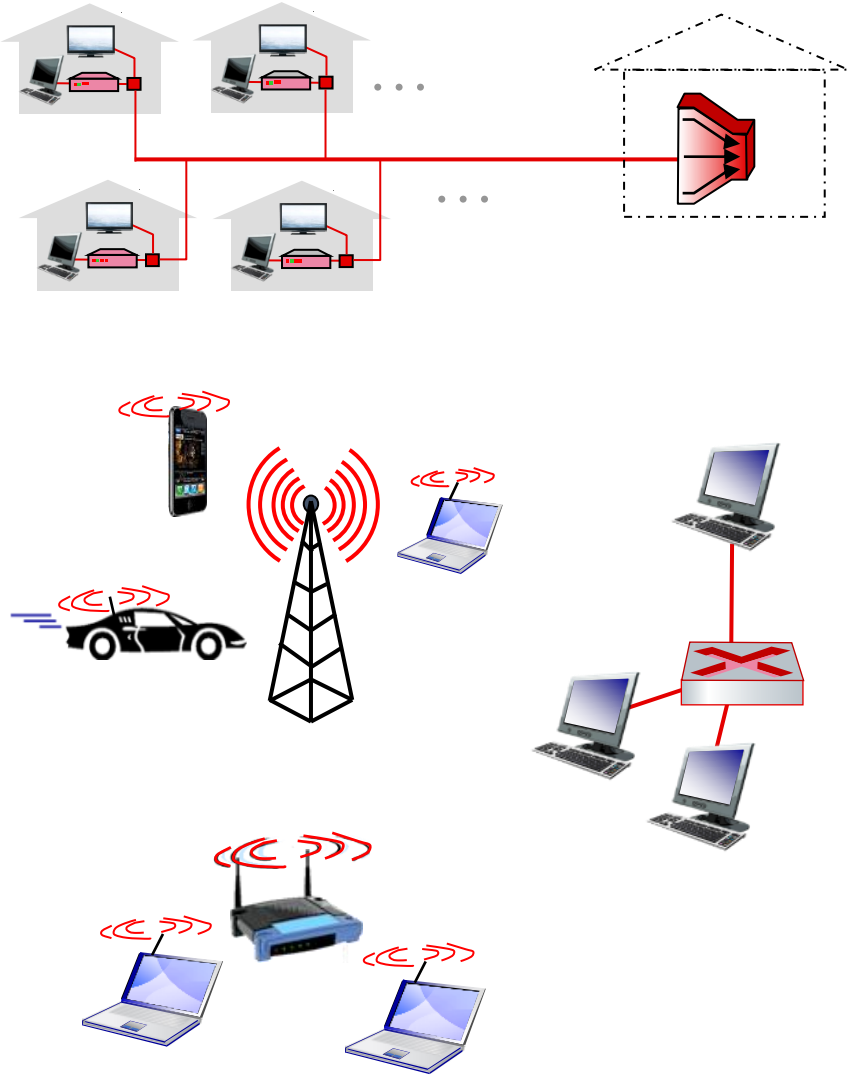
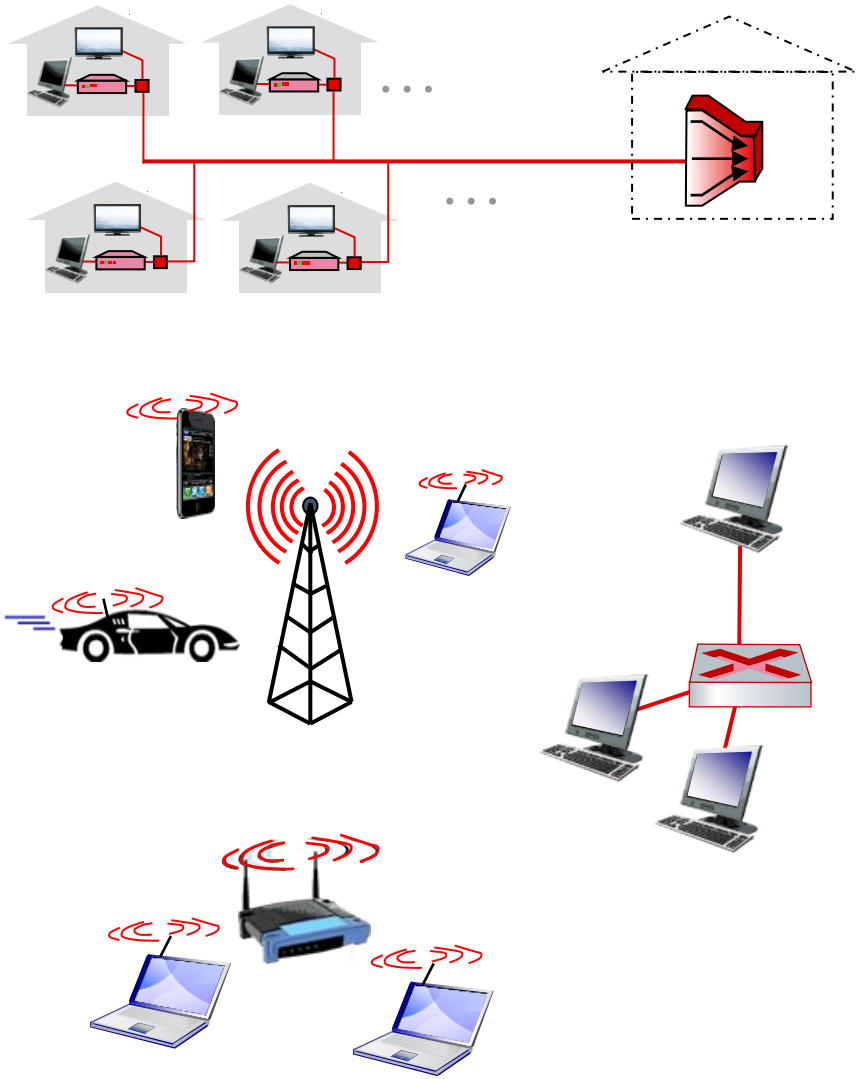
Link layer: services

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



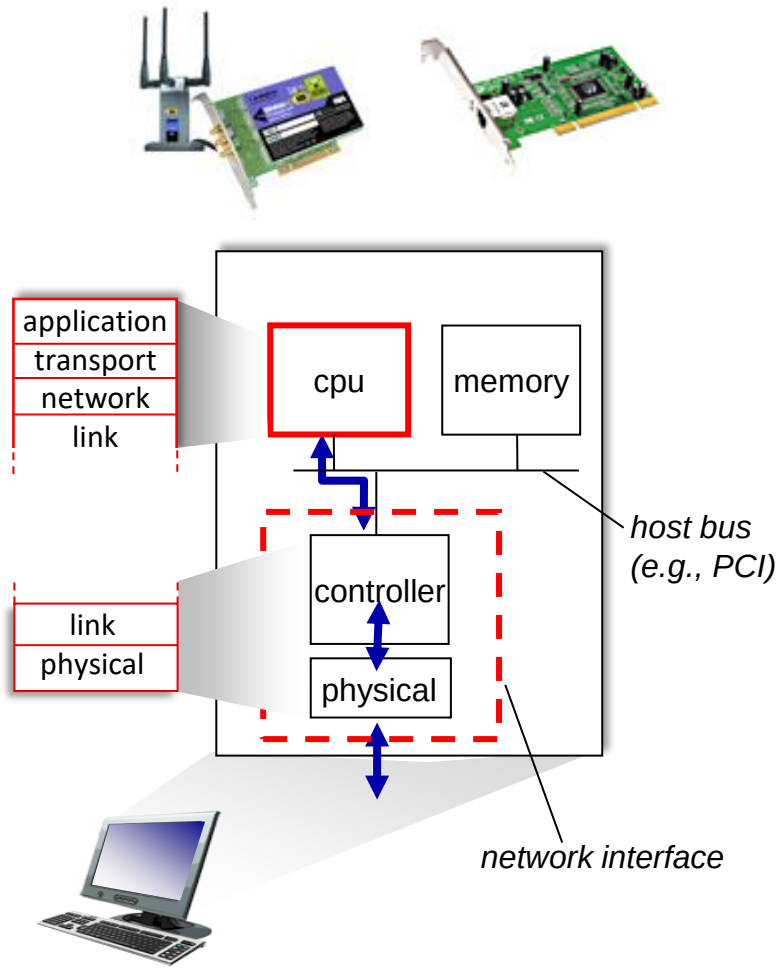
Link layer: services (more)

- 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
- 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

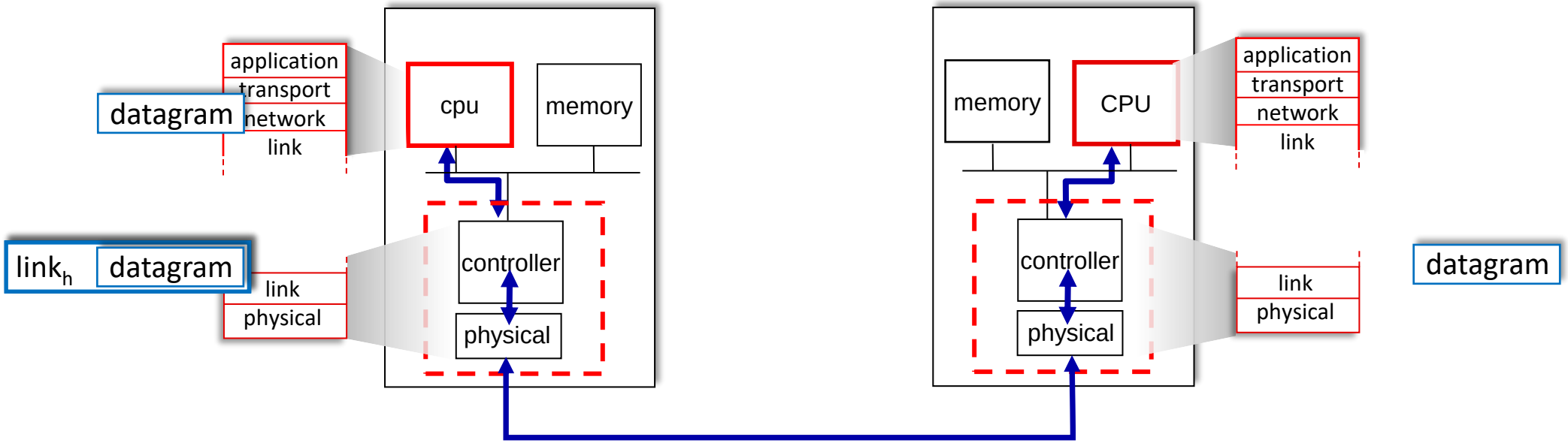


Where is the link layer implemented?

- 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:

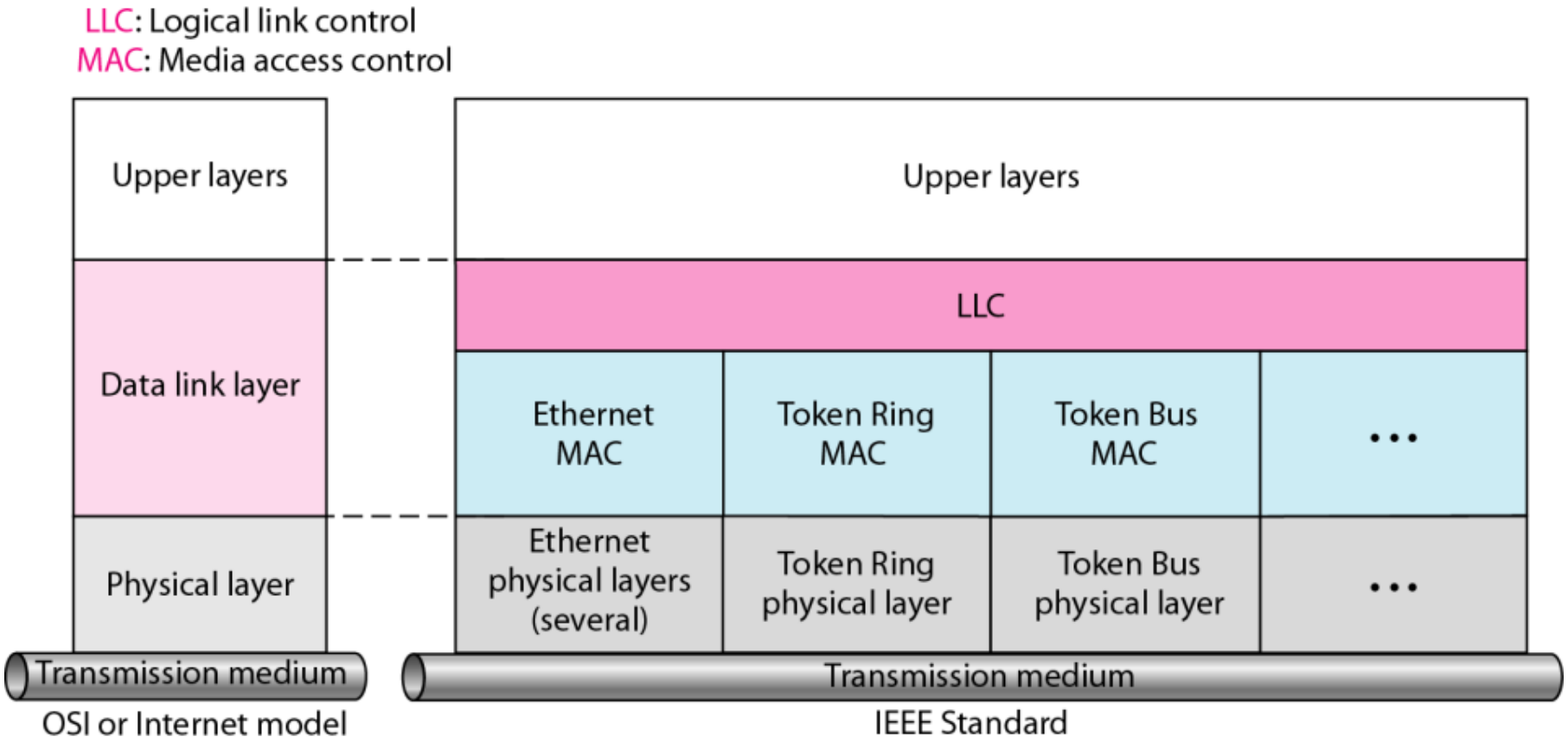
- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

IEEE standard for LAN

IEEE divided the Data link layer into two sublayer:

- upper layer **:** logical link control (**LLC**); flow and error control.
- Lower sublayer : Multiple access (**MAC**); media access control.
  - Multiple access **(MAC)** :for resolving access to the shared media.
  - If channel is dedicated (point to point) we do not need the **(MAC);** sublayer.

IEEE standard for LAN



Data Link Layer Services

- Flow Control
- Error Control
- Access Control
- Addressing and Framing

Flow Control

- Stop and Wait
- Go back N
- Selective repeat

- Same as discussed in transport layer difference is in data link layer it works between to directly connected nodes whereas in transport layer it works between source and destination.

Error Control

- Error can be single bit or burst error

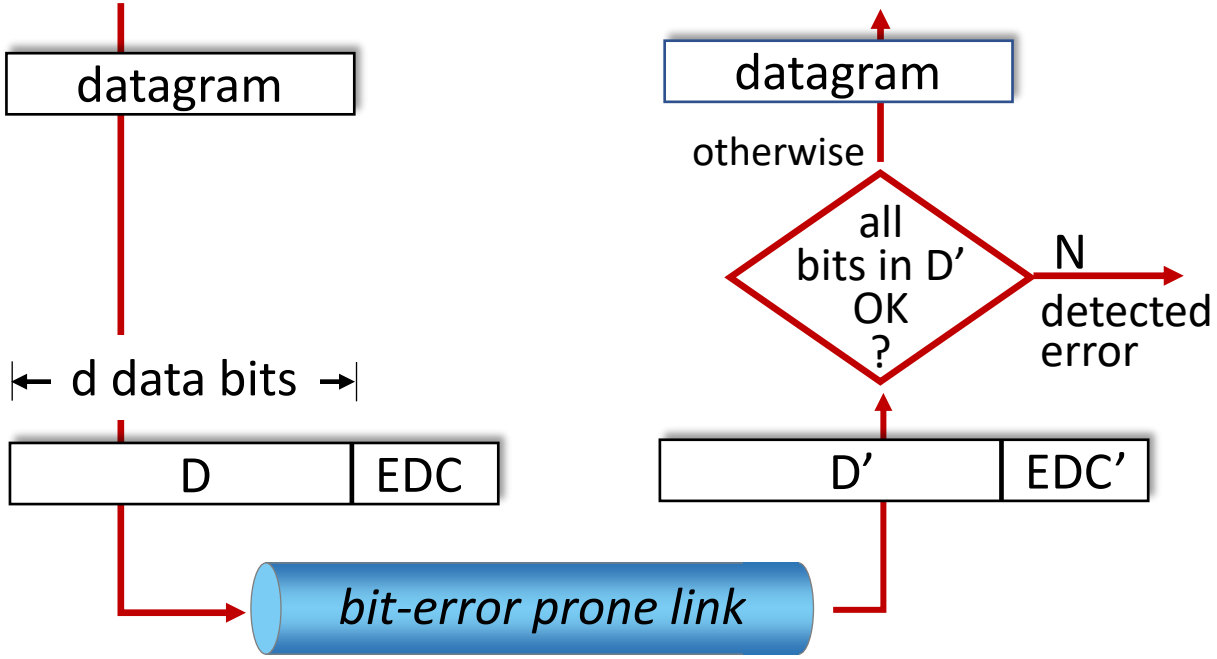
- Error detection:
  - Parity check – Simple parity and 2D parity check
  - Checksum – used in higher layers (already discussed)
  - Cyclic Redundancy Check - used in data link layer

Error detection and correction

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

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

Total bits = D + EDC



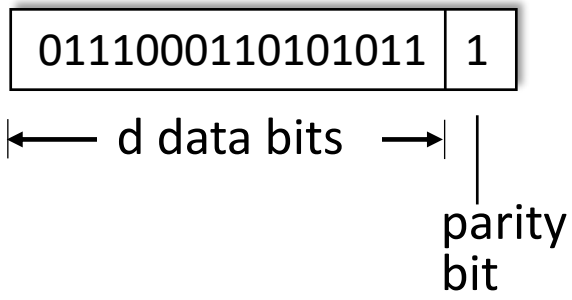
Error detection not 100% reliable!

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

Parity checking

single bit parity:

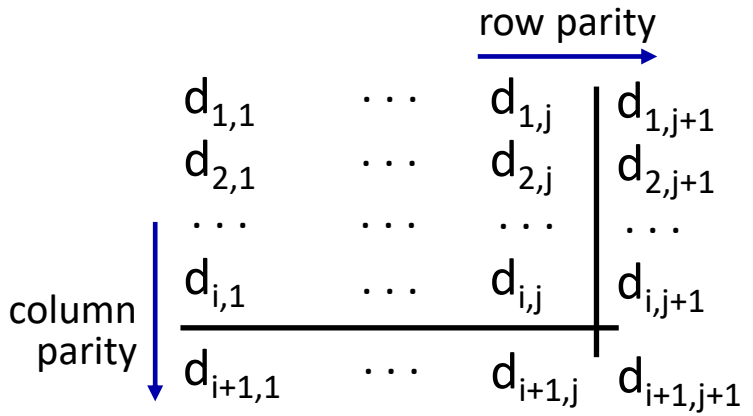
- detect single bit errors



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

two-dimensional bit parity:

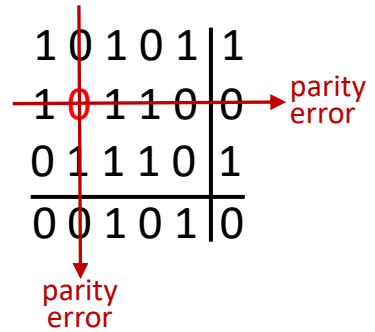
- detect *and correct* single bit errors



no errors:

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

detected and correctable single-bit 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:

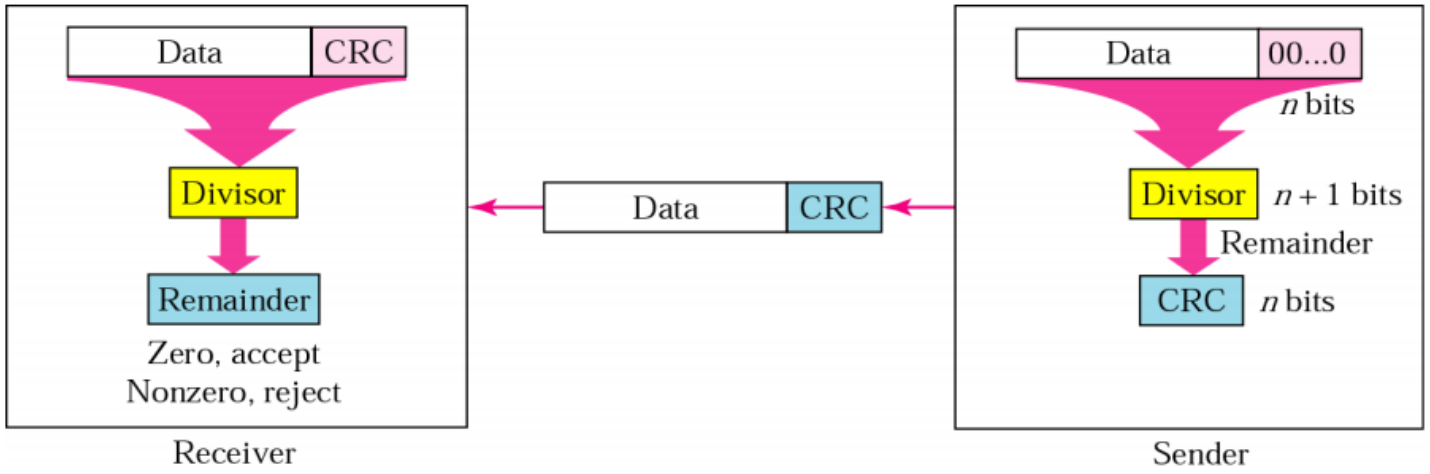
- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment content
- checksum value put into UDP checksum field

receiver:

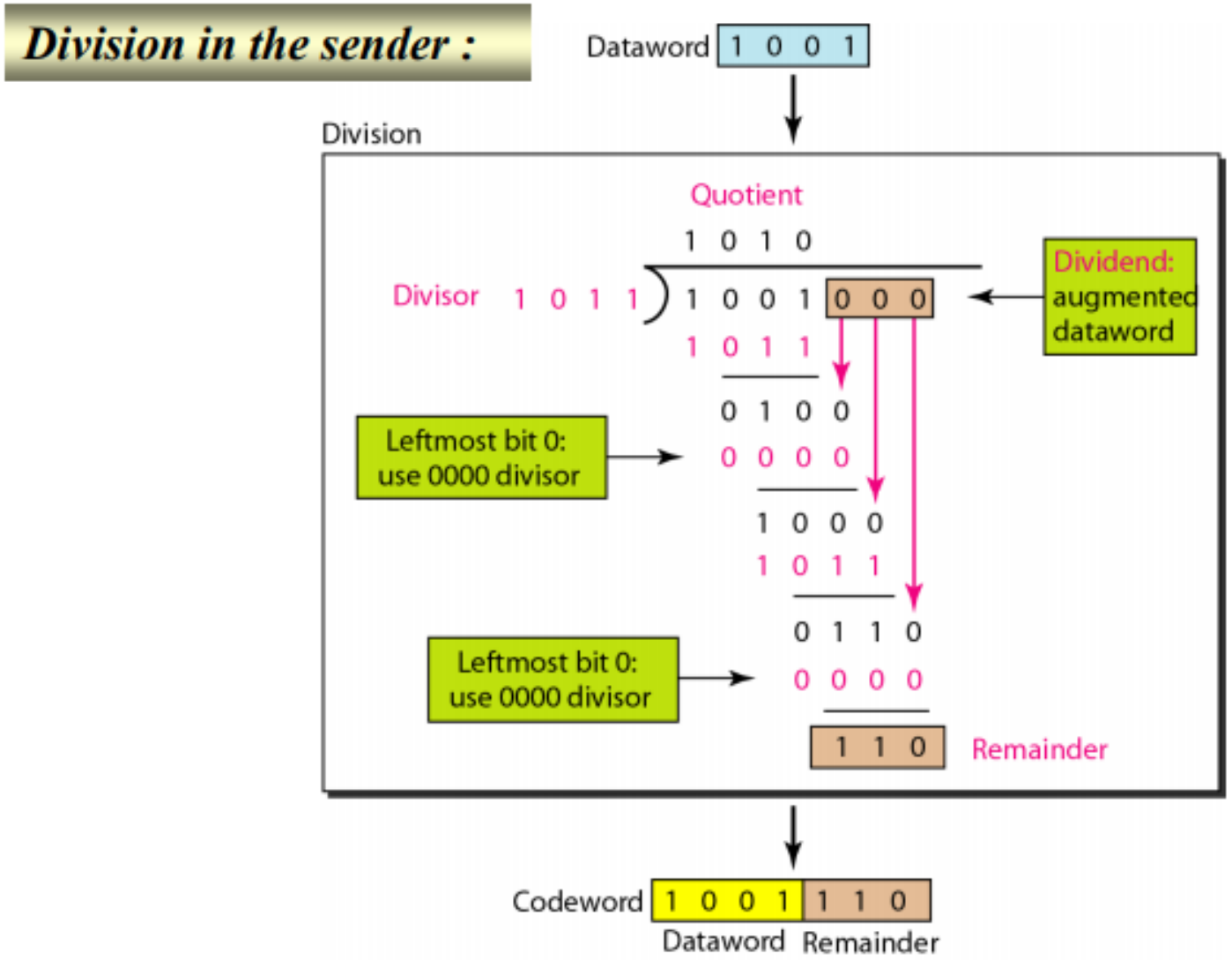
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - not equal - error detected
  - equal - no error detected.

Cyclic Redundancy Check (CRC)

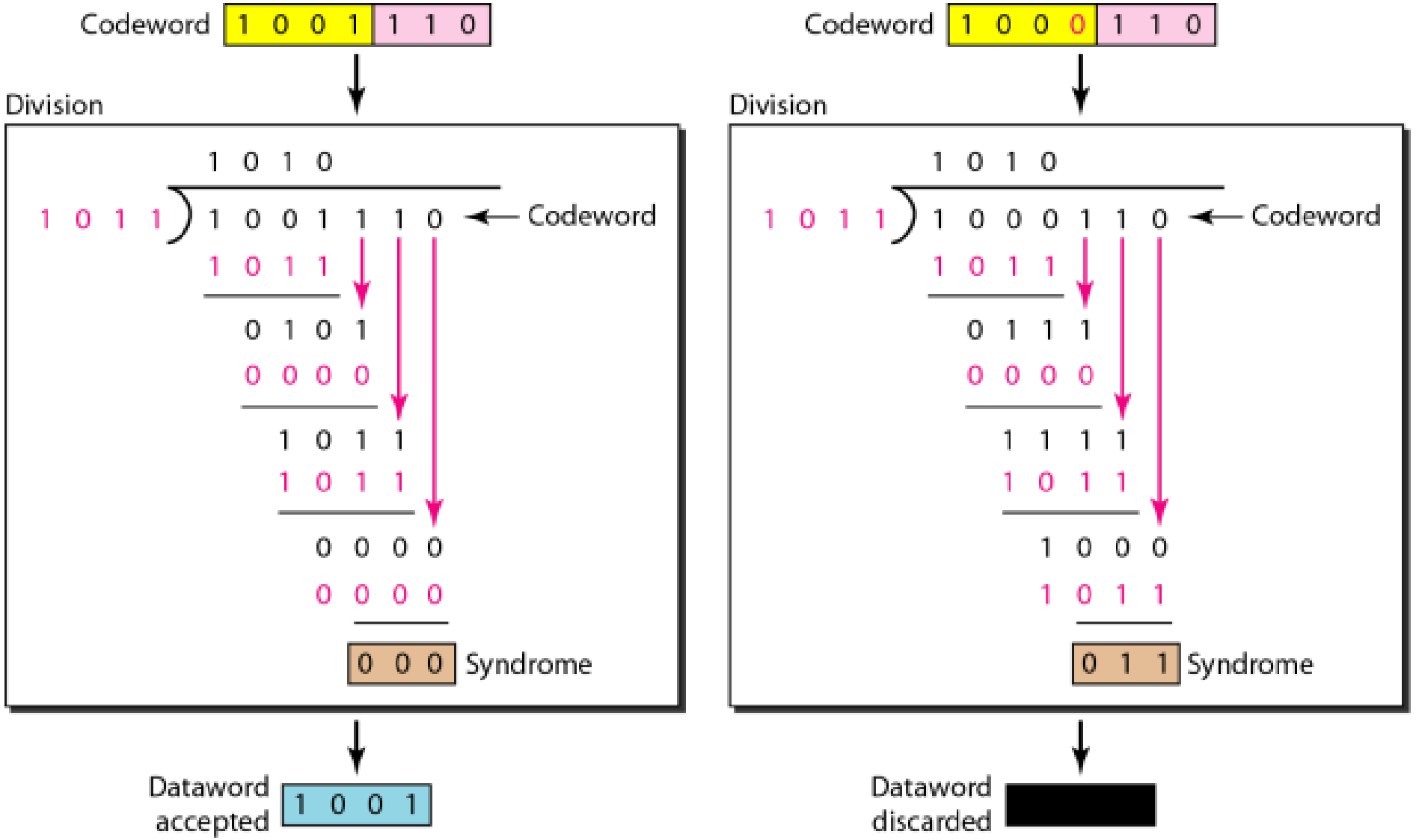
- more powerful error-detection coding
  - a group of error control bits (which is the remainder of a polynomial division of a message polynomial by a generator polynomial) is appended to the end of the message block • with considerable burst-error detection capability.
- widely used in practice (Ethernet, 802.11 WiFi)



Sender side operation

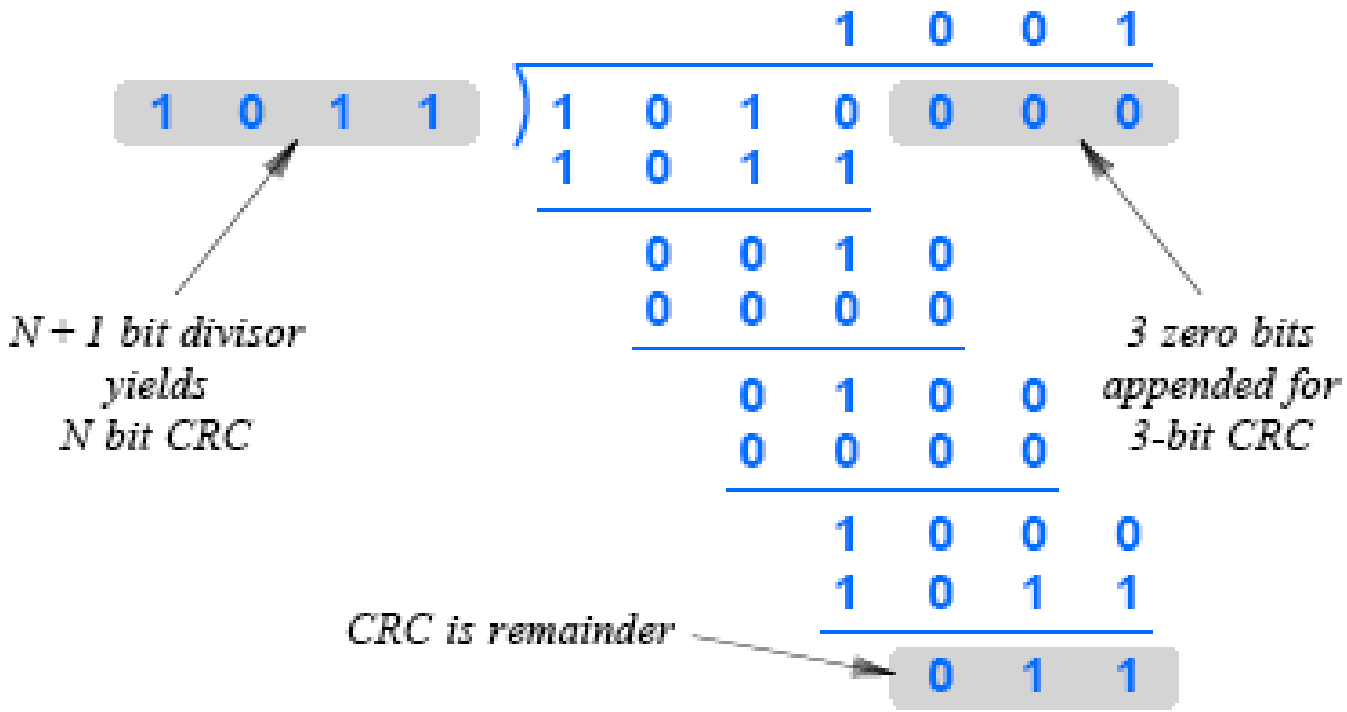


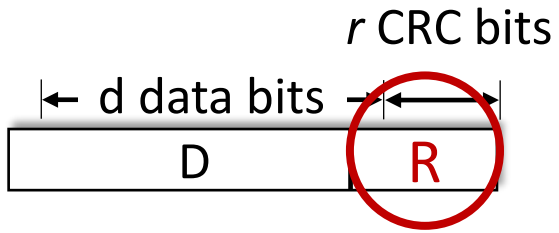
Receiver side operation



Cyclic Redundancy Check (CRC)

- Data: 1010
- Divisor: 1011
- CRC=???





CRCs and Polynomial Representation

We can view the above process as a **polynomial division**:

Think of each bit in a binary number as the coefficient of a term in a polynomial

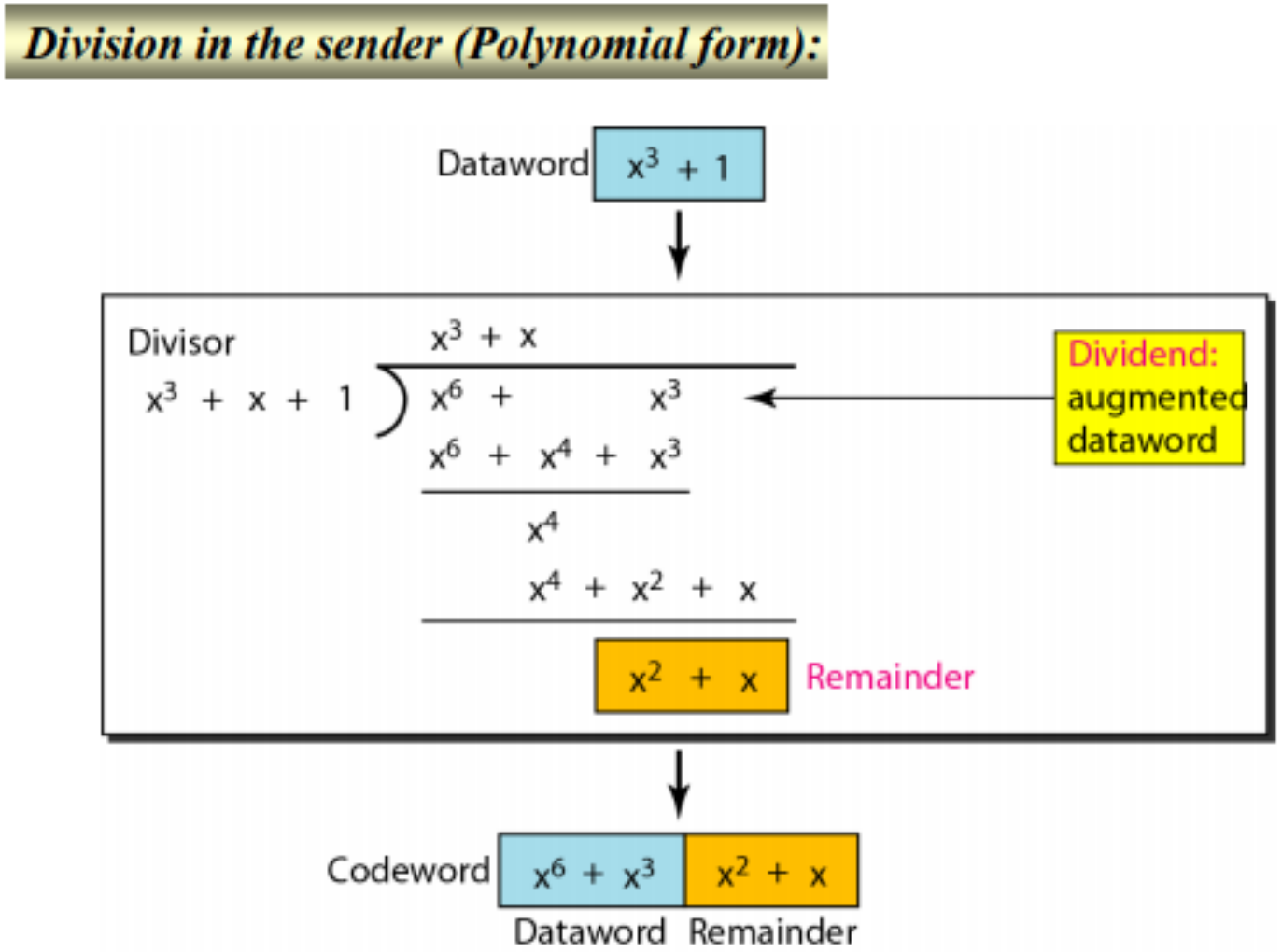
For example, we can think of the divisor in Figure, 1011, as coefficients in the following polynomial:

$$1 \times x^3 + 0 \times x^2 + 1 \times x^1 + 1 \times x^0 = x^3 + x + 1$$

Similarly, the dividend, 1010000, represents the polynomial:

$$x^6 + x^4$$

Polynomial Form



Common CRC Codes

Code	Generator Polynomial g(X)	Appended Bits	Applications
CRC-4	$X^4 + X + 1$	4	ITU G.704
CRC-8	$X^8 + X^2 + X + 1$	8	ATM header
CRC-10	$X^{10} + X^9 + X^5 + X^4 + X + 1$	10	ATM AAL
CRC-16-CCITT	$X^{16} + X^{12} + X^5 + 1$	16	Bluetooth
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$	32	LANs

CRC	$C(x)$
CRC-8	$x^8 + x^2 + x^1 + 1$
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^1 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Both Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11) use CRC-32.

Multiple access links, protocols (Access Control)

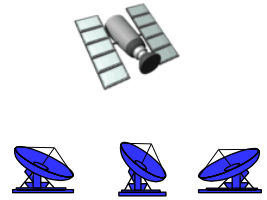
two types of “links”:

- point-to-point
  - point-to-point link between Ethernet switch, host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - 802.11 wireless LAN, 4G/5G, satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: satellite

humans at a cocktail party (shared air, acoustical)

Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

An ideal multiple access protocol

given: multiple access channel (MAC) of rate *R* bps

Requirement:

1. when one node wants to transmit, it can send at rate *R*.
2. when M nodes want to transmit, each can send at average rate *R/M*
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple