Redes II

Universidade do Algarve

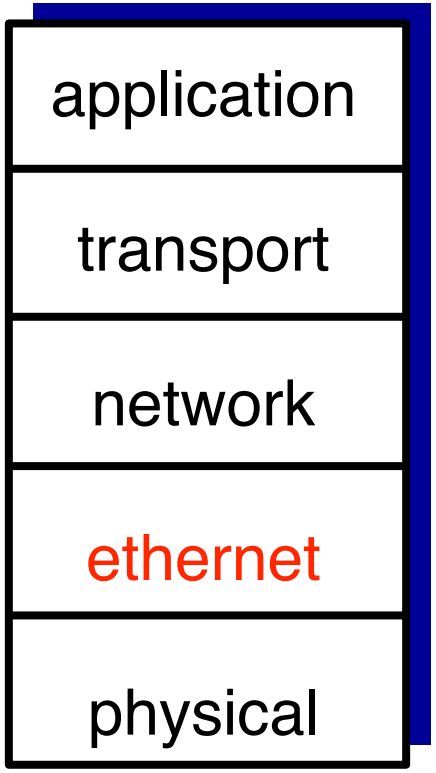
Aula Teórica 4

Néstor Cataño
nestor.catano@gmail.com

The link (ethernet) layer

Goal:

1. To understand the principles behind error detection and error correction.



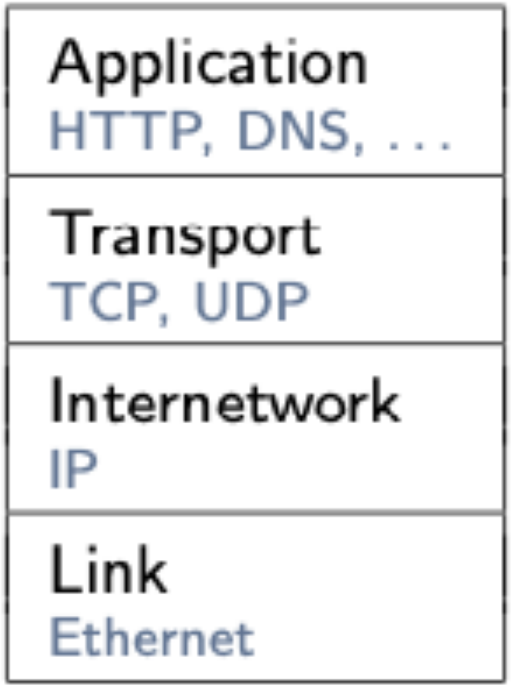
Roadmap

1. Recap: the link layer
2. Error detection and error correction

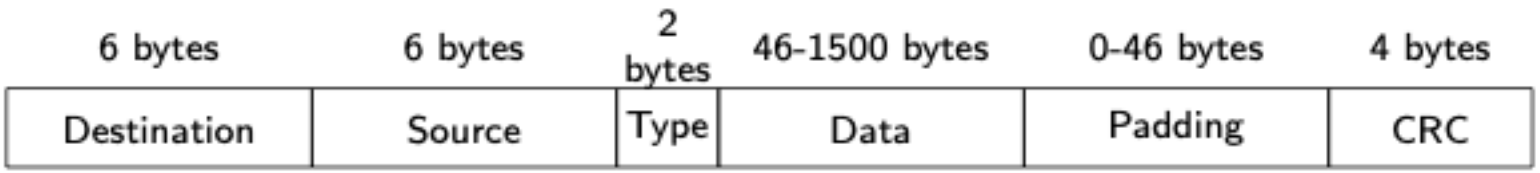
Recap: the link (ethernet) layer

What is ethernet and why do we care?

- Ethernet is a popular approach to solving the problem of transmitting data over a LAN (local area network).
- Immensely successful to this day, it continues to evolve wired, high-speed GigaBytes, wireless, etc.
- Provides link layer support for encapsulating IP datagrams.



ethernet frames



Destination - MAC address of the device where the packet is going

Source - MAC address from which the packet came from

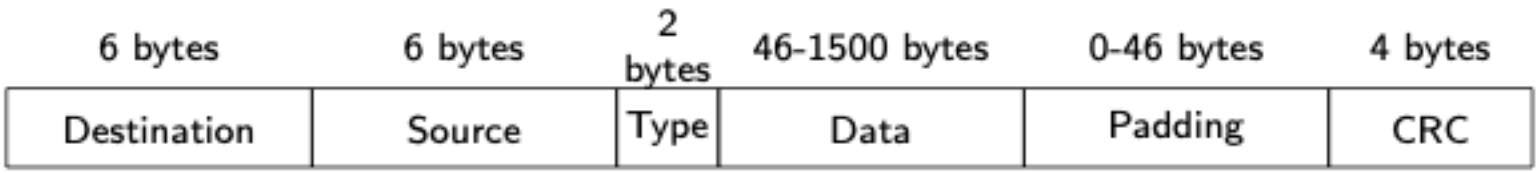
Type - it allows multiplexing (which network protocol will be used)

Data - the datagram that we are sending

Padding - to complete the minimum size of the datagram

CRC - cyclic redundant check, used to handle errors

ethernet frames

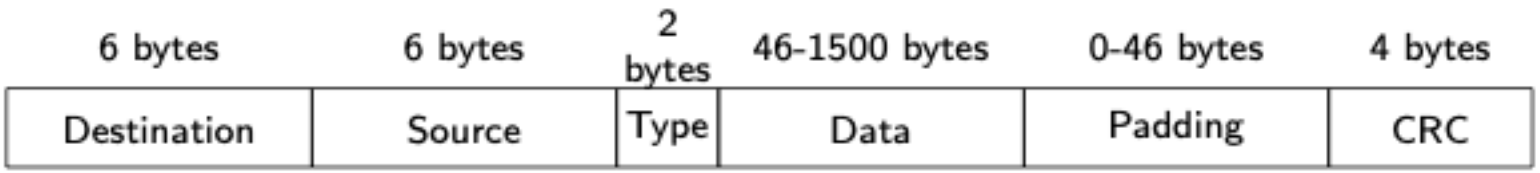


If we were to send 1501 bytes of data, how many frames do we need to send?

Frame 1. the Data field contains 1500 bytes.

Frame 2. the Data field contains 1 data byte plus 45 bytes of padding. Those padding bytes are the Padding field.

ethernet frames - example I



You are sending data over ethernet that is 5400 bytes long?

1. How many ethernet frames will this be?
2. How many bytes (excluding padding) will be in the data field of the last ethernet frame?

Roadmap

1. Recap: the link layer
2. Error detection and error correction

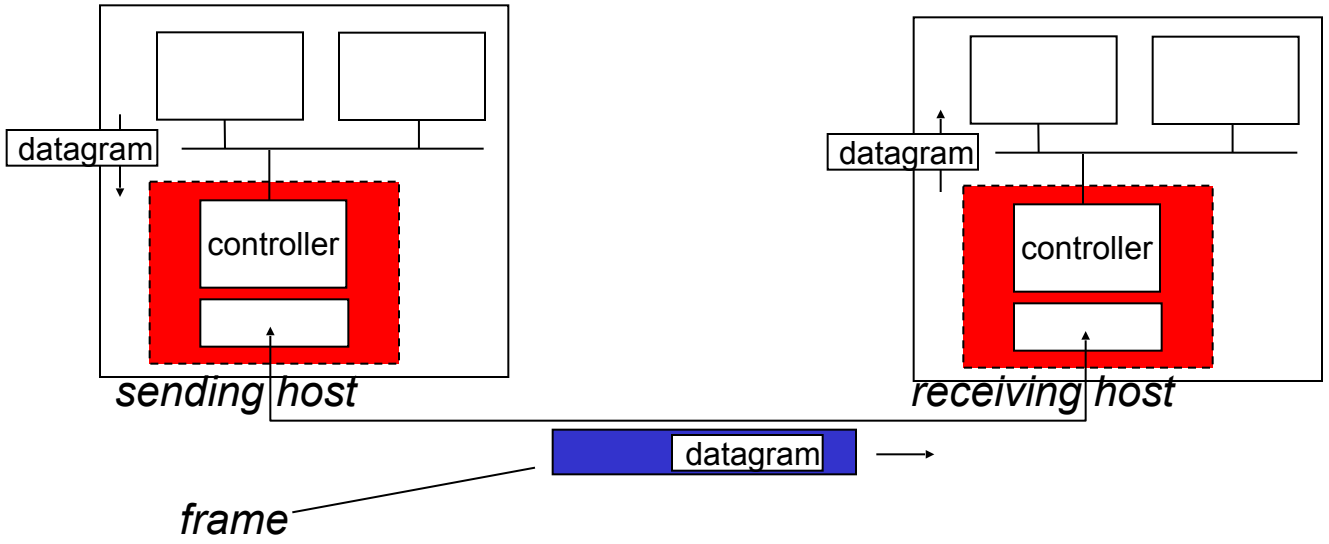
Link layer services

- **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!
- **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 the 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

Adaptors communicating

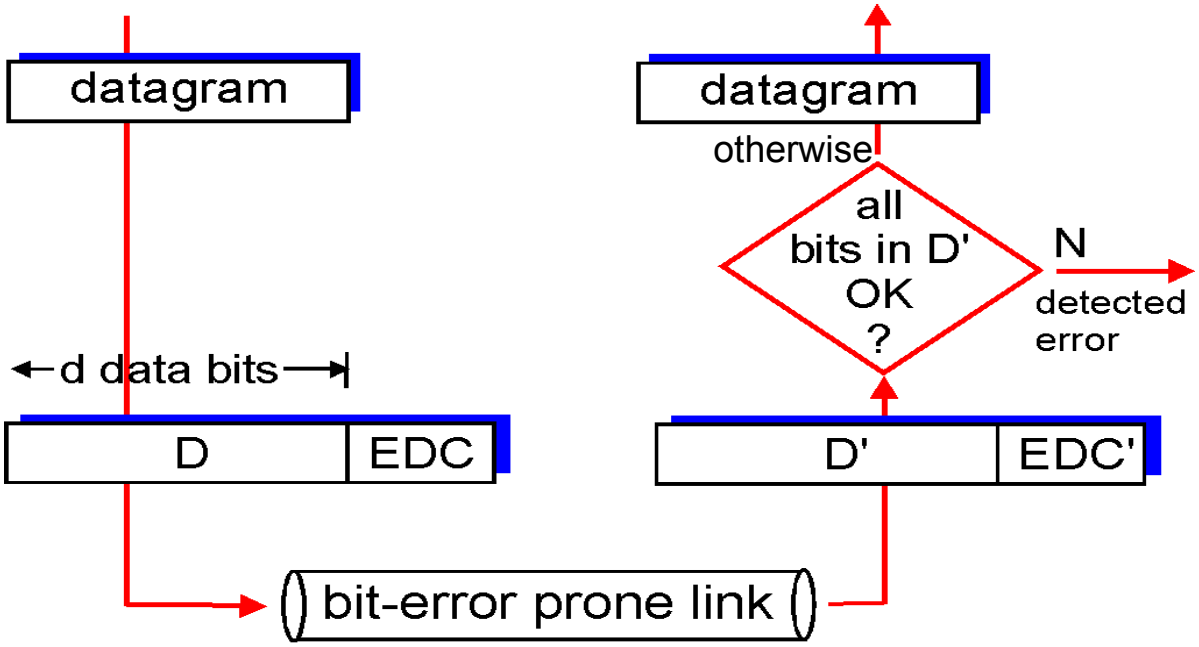


- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc.
  - extracts datagram, passes to upper layer at receiving side

Error detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, it may include header fields

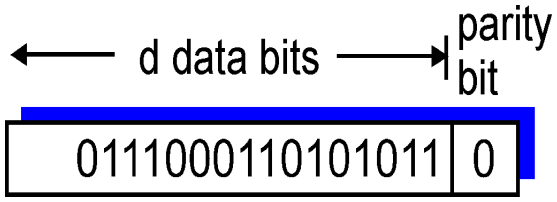
- Error detection is not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity checking

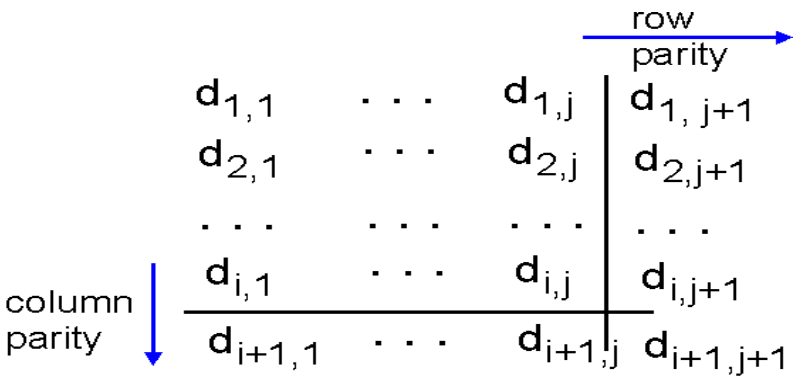
single bit parity:

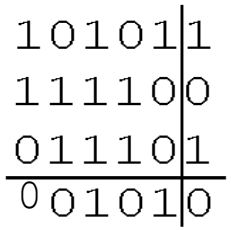
- detect single-bit errors



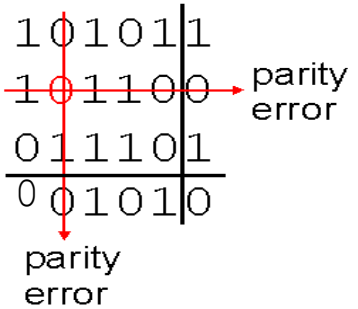
two-dimensional bit parity:

- detect and correct single-bit errors





no errors



correctable single bit error

Internet checksum (review)

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

sender:

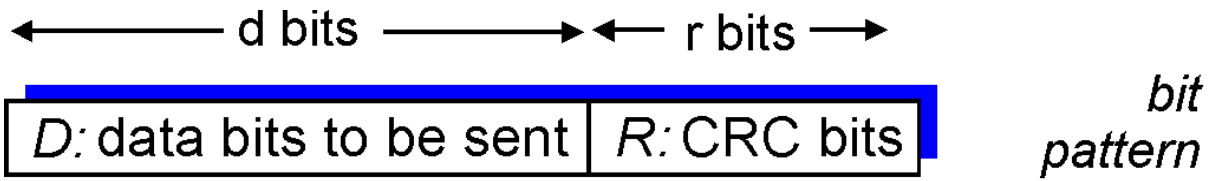
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

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?

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
  - <D,R> exactly divisible by G (modulo 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, ATM)



$D * 2^r$ XOR R *mathematical formula*

CRC example

want:

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

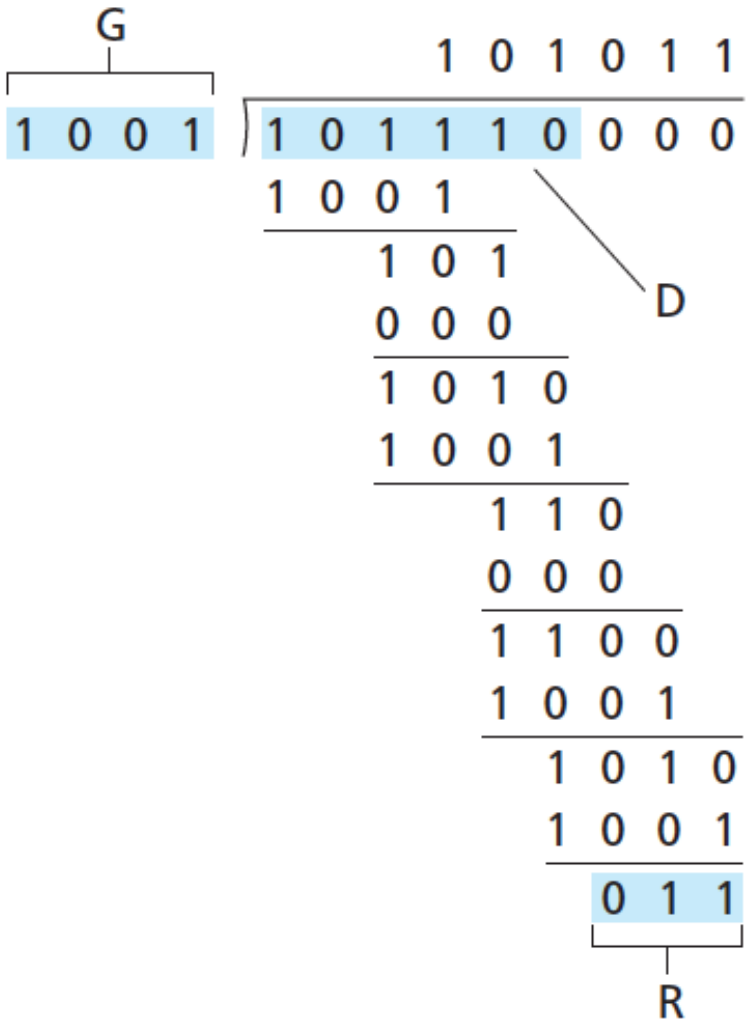
equivalently:

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

equivalently:

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

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



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

Summary

- Error detection
- Error correction