Computer Networks
COL 334/672

Error Checks and Access Rules

Slides adapted from KR

Sem 1, 2025-26

Link Layer: Services

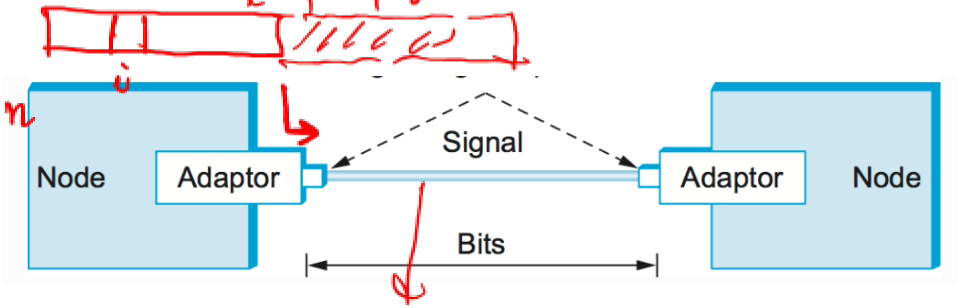
- Encoding
- Framing
- Error detection
- Addressing
- Link access

Error Detection

- There can be bit errors as a frame is transmitted
- **Question: (how)** can we detect errors at the receiver?
- Simplest approach: append a copy to the frame
  - If bit(i) != bit(n+i) → bit error! Drop the frame

n bits copy

$\forall i < n$



bit errors on the channel

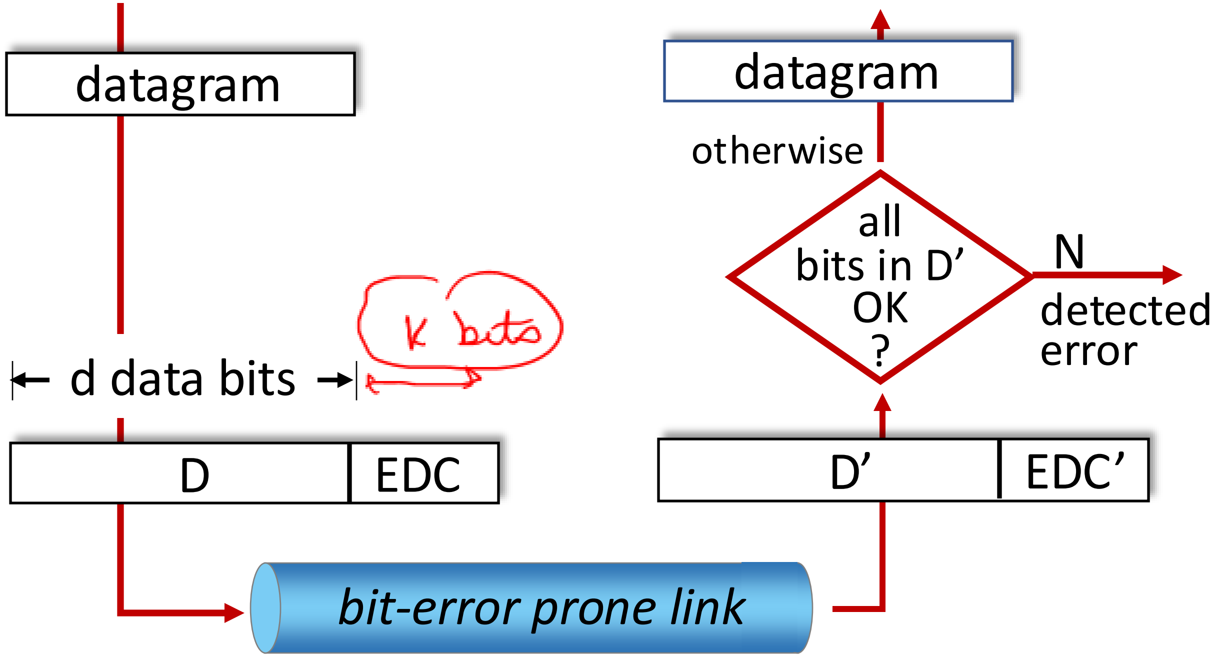
1. does not detect errors if bits flipped at same index in the two copies
2. efficiency = 50%

Error detection → Coding Theory

① Parity bit
② hash (D), Δ
③ Cyclic Redundancy Check (CRC)

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

Goal: maximize probability of detecting errors using only a small number of redundant bits

Methods for error detection

- Parity bit
- Checksum
- Cyclic redundancy check

Parity bit

Even parity:

$\underbrace{0\ 1\ 0\ 0\ 1\ 1}_{3\ 1\text{s},\ \text{Parity } 1}\ \boxed{1}$

odd parity:

Add zero at the end

Claim: Parity bit can detect single-bit errors

Parity bit can detect odd number of bit errors

Issue: does not detect even number of bit errors

↳ $\boxed{\text{Bit errors typically happen in bursts}}$

Checksum

Hash(D): cryptographic hash → ① [Processing is an overhead]

② Bit errors are random

↓ As simple as possible

checksum(D)

D'

H(D) = H(D')

Sender

D | checksum

4-bit checksum

| 1101 | 0110 | 1011 |

13 + 6 = 19

12 + 7 =

| 1100 | 0111 | 1011 |

42 = 10 1010 (lower)

1010
+ 10
= 1100

10011

0011
+ 1
= 0100 → 1s complement

| 0010 | 0100 | 1001 |

0010
+ 0100
= 0110

Receiver

1101
0110
1011
= 10001

0010
+ 1101
= 1111

claim: if error in only 1 block, checksum will detect it

Error correction code

→ Hamming codes

where the error is

- Also known as Forward Error Correction

(1) Requires more bits than Error detection

16 bits

- Using 2D parity

4x4

  Can detect *and* correct errors (without retransmission!)

  - detect *and correct* single bit errors

row parity →

column parity ↓

$d_{1,1}$	. . .	$d_{1,j}$		$d_{1,j+1}$
$d_{2,1}$	. . .	$d_{2,j}$		$d_{2,j+1}$
. . .	. . .	. . .		. . .
$d_{i,1}$	. . .	$d_{i,j}$		$d_{i,j+1}$
$d_{i+1,1}$	. . .	$d_{i+1,j}$		$d_{i+1,j+1}$

no errors:

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

detected and correctable single-bit error:

1 0 1 0 1	1	
1 0 1 1 0	0	parity error
0 1 1 1 0	1	
1 0 1 0 1	0	

- Always useful?
  - When cost of retransmissions are high
  - When there are frequent bit errors

(1) satellite communication / latency is very high

(2) video calling / voice calling

Cyclic Redundancy Check

- Based on *finite fields*, gives a more systematic framework for error detection

M: 1 0 1 0 1 → n bits

→ $M(x)$ ↳ $x^4 + x^2 + 1$

→ $P(x)$

↳ $G(x)$

(Both sender & receiver know this polynomial)

such that $G(x) \mid P(x)$

Transmit $P(x)$

→ Receiver: $P'(x)$ → Check if $G(x) \mid P'(x)$

How do you transform $M(x) \to P(x)$

Some facts [for this course!]

- Any polynomial $B(x)$ can be divided by a divisor polynomial $C(x)$ if $B(x)$ is of higher degree than $C(x)$
- Any polynomial $B(x)$ can be divided once by a divisor polynomial $C(x)$ if $B(x)$ is of the same degree as $C(x)$
- The remainder obtained when $B(x)$ is divided by $C(x)$ is obtained by performing the exclusive OR (XOR) operation on each pair of matching coefficients

$B(x) = x^3$

$C(x) = x^3 + 1$ 1001

$x^3+1 \overline{)x^3}$

1 → quotient

$$\begin{array}{r} 1000 \\ 1001 \\ \hline 0001 \end{array}$$ → Remainder

$B(x) = C(x) \times Q(x) + R(x)$

$1001 \times 1 \oplus 0001$

$1001 \oplus 0001 = 1000$

CRC Algorithm

1. Multiply $M(x)$ by x^k; that is, add k zeros at the end of the message. Call this zero-extended message $\boxed{T(x)}$. →
2. Divide $T(x)$ by G(x) and find the remainder. → $R(x)$
3. Subtract the remainder from $T(x)$

$$T(x) = G(x) \times Q(x) + R(x)$$

$$G(x) \mid \underbrace{T(x) - R(x)}$$

$$G(x) \mid P(x)$$

CRC Example

- M(x) = 101110
- G(x) = 1001 → 3
- What is P(x)?

```
          10101
1001 ) 101110000
       1001
       ----
       001010
         1001
         ----
          1100
          1001
          ----
           1010
           1001
           ----
             11
```

```
101110000 - 11
         11
-----------
101110011
```

$x^k \cdot M(x)$

k bits

R(x)

Replace them with

Cyclic Redundancy Check (CRC)

$P'(x) = P(x) + E(x)$

- How to pick G(x)?
  - Transmitted message: P(x) + E(x)
  - For errors to go undetected, E(x) should be divisible by G(x)
  - Pick G(x) such that above is unlikely to happen for common errors
  - **Claim**: If G(x) has non-zero coefficients at x^k and x^0, all single-bit errors can be detected

$G(x) \mid P'(x) \Rightarrow G(x) \mid P(x) + E(x)$

$\Downarrow$

$G(x) \mid E(x)$

$x^k + \dots x^0$

$E(x) = x^i$

$x^k + \dots x^0 \quad \cancel{*} \quad x^i$