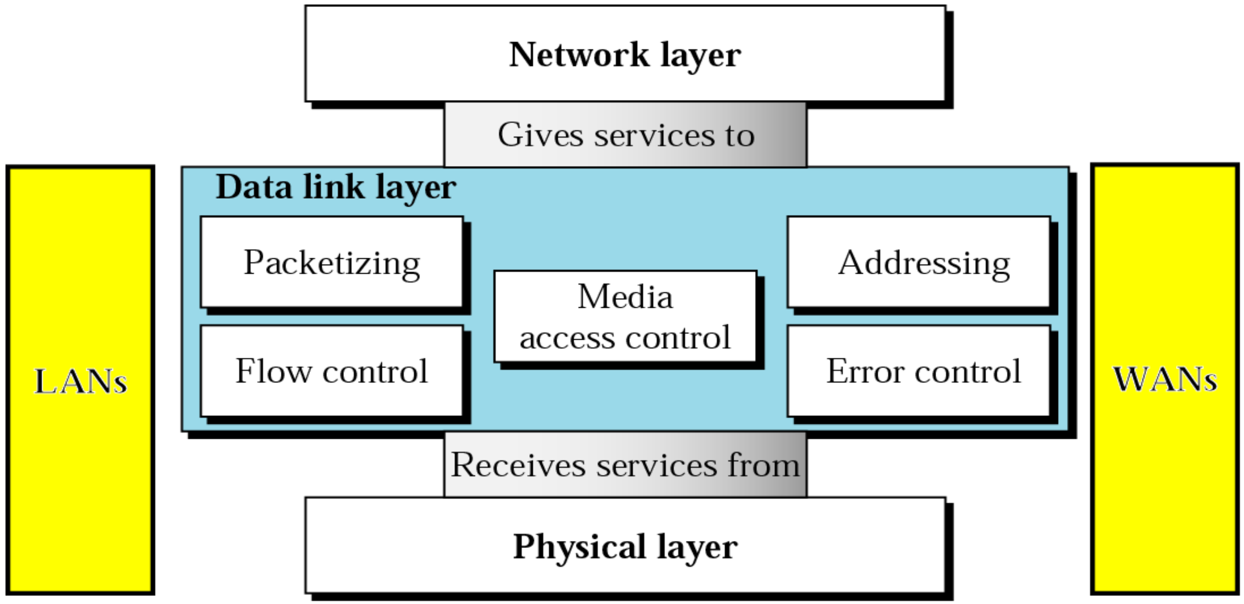
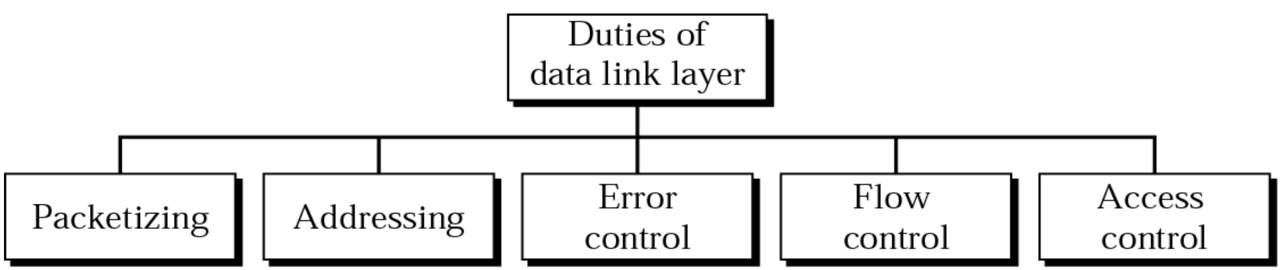
Error Control (1)

Required reading:
Forouzan 10.1, 10.2
Garcia 3.9.1, 3.9.2, 3.9.3

CSE 3213, Fall 2015
Instructor: N. Vlajic

Data Link Layer





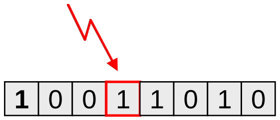
Error Control

Why Error Control? – data sent from one computer to another should be transferred reliably – unfortunately, the physical link cannot guarantee that all bits, in each frame, will be transferred without errors

- error control techniques are aimed at improving the error-rate performance offered to upper layer(s), i.e. end-application

Probability of Single-Bit Error – aka bit error rate (BER):

- wireless medium: $p_b=10^{-3}$
- copper-wire: $p_b=10^{-6}$
- fibre optics: $p_b=10^{-9}$



Approaches to Error Control

(1) Error Detection + Automatic Retrans. Request (ARQ)

- fewer overhead bits ☺
- return channel required ☹
- longer error-correction process and waste of bandwidth when errors are detected ☹

(2) Forward Error Correction (FEC)

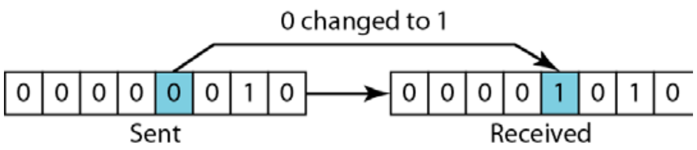
- error detection + error correction

Error Control (cont.)

Types of Errors

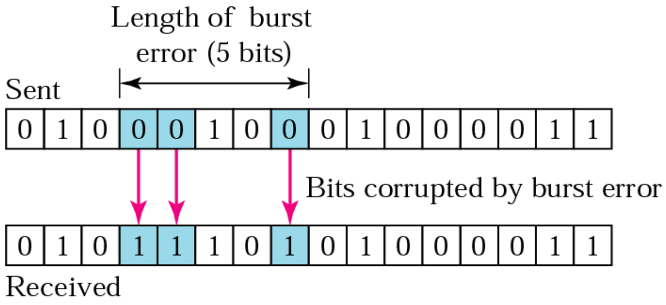
(1) Single Bit Errors

- **only one bit in a given data unit (byte, packet, etc.) gets corrupted**



(2) Burst Errors

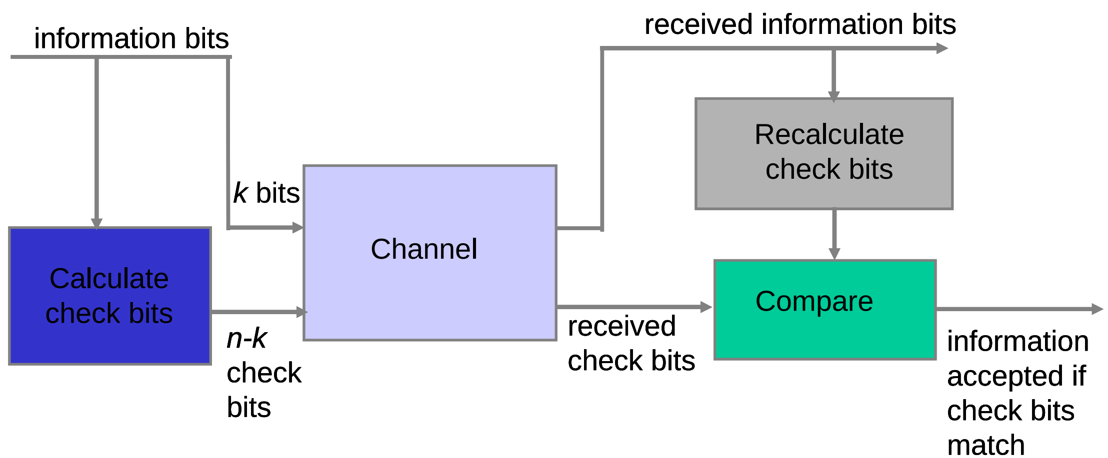
- **two or more bits in the data unit have been corrupted**
- **errors do not have to occur in consecutive bits**
- **burst errors are typically caused by external noise (environmental noise)**
- **burst errors are more difficult to detect / correct**

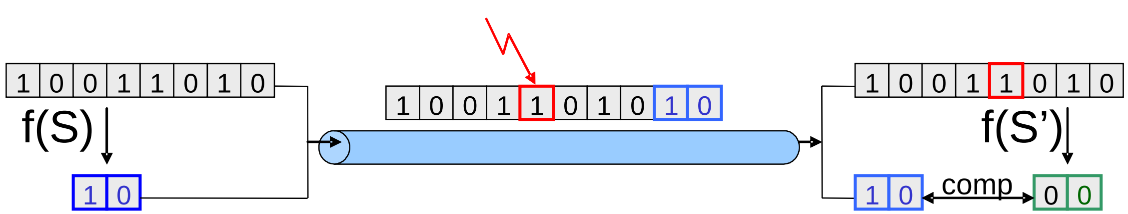


Error Control (cont.)

Key Idea of Error Control – **redundancy!!! – add enough extra information (bits) for detection / correction of errorors at the destination**

- **redundant bits = 'compressed' version of original data bits**
- **error correction requires more redundant bits than error detection**
- **more redundancy bits ⇒ better error control ☺ ⇒ more overhead ☹**





Modulo-2 Arithmetic

Modulo 2 arithmetic is performed digit by digit on binary numbers.
Each digit is considered independently from its neighbours.
Numbers are not carried or borrowed.

$$0 \oplus 0 = 0 \qquad 1 \oplus 1 = 0$$

a. Two bits are the same, the result is 0.

$$0 \oplus 1 = 1 \qquad 1 \oplus 0 = 1$$

b. Two bits are different, the result is 1.

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

Hamming Distance

Hamming Distance Between 2 Codewords – number of differences between corresponding bits

- can be found by applying XOR on two codewords and counting number of 1s in the result

Minimum Hamming Distance (d_{min}) in a Code – minimum Hamming distance between all possible pairs in a set of codewords

- d_{min} bit errors will make one codeword look like another
- larger d_{min} – better robustness to errors

Example [k=2, n=5 code]

Code that adds 3 redundant bits to every 2 information bits, thus resulting in 5-bit long codewords.

Dataword	*Codeword*
00	00000
01	~~01011~~ **11110**
10	10101
11	11110

Any 1-bit error will not be detected.

Any 1-bit and 2-bit errors will be detected.