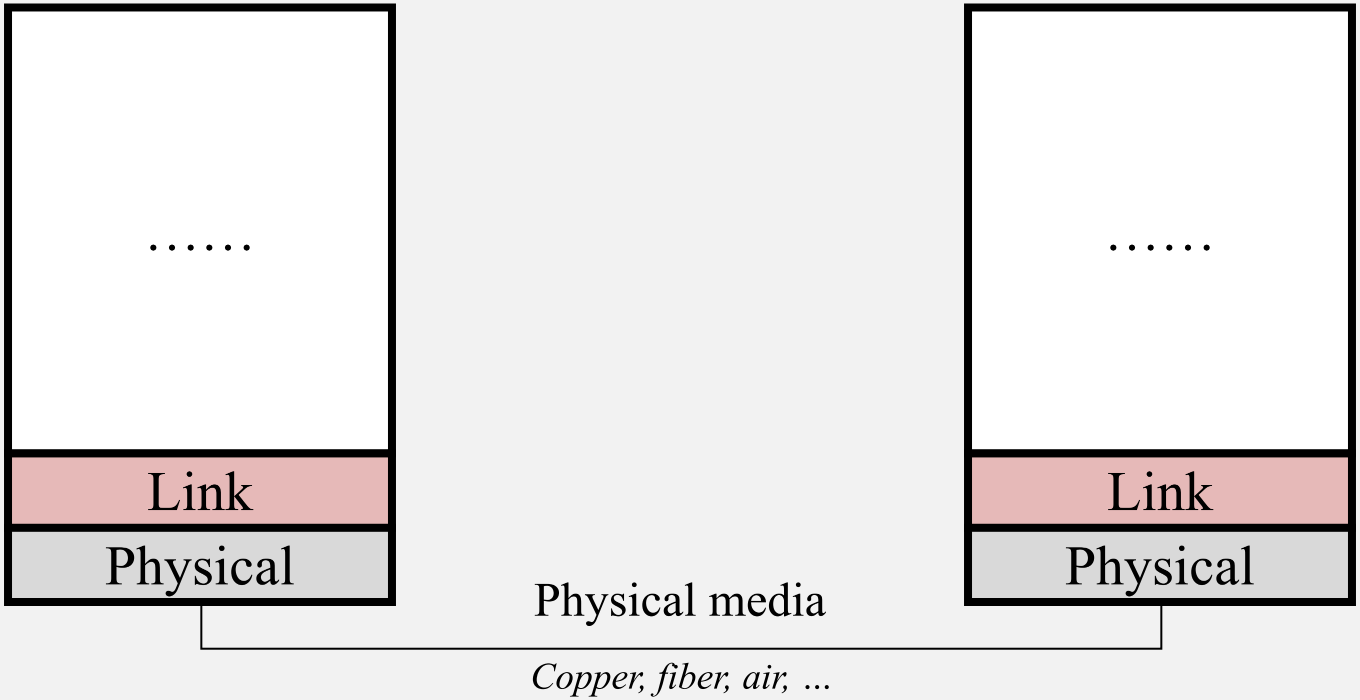
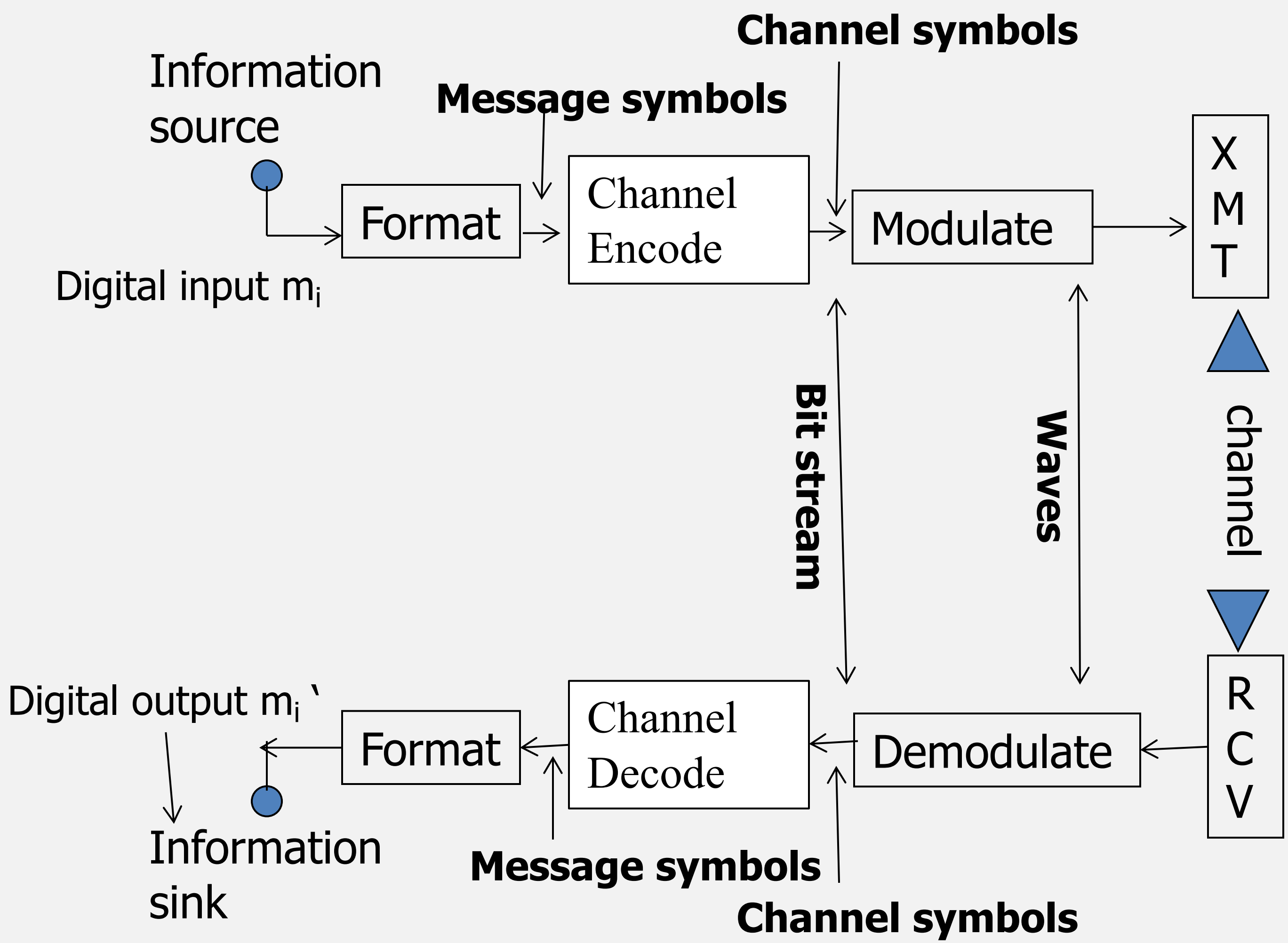
Data Link Control Layer

CSC/ECE 570 - Fall 2024, Section 002

Where is PHY?


……
Link
Physical
……
Link
Physical
Physical media
Copper, fiber, air, …

Block Diagram of a Digital Communication System



Bandwidth limited Bit rate

- Nyquist's theorem
  - Maximum bit rate = $2H \log_2 V$ bits/sec
  - H = bandwidth
  - V = number of discrete states
- Shannon's theorem
  - Maximum bit rate = $H \log_2 (1 + S/N)$ bits/sec
  - Note on application
    - SNR in Shannon's theorem - ratio of power content (P_S/P_N)
    - Usual unit of SNR - dB, a logarithmic unit
    - dB = $10 \log_{10} (P_S/P_N)$

Question

Shannon Capacity

Show Correct Answer

Based on the Shannon capacity formula, a client device sending application data to a server via a direct wireless link can always achieve H log2 (1 + S/N) bits/sec of application throughput, where H is the bandwidth, S is the signal strength, and N is the background noise.

A	TRUE
B	FALSE

Question (consider Shannon Capacity)

Which of the following affect the theoretical limit of information that can be transmitted via a channel?

A Bandwidth

B Modulation

C Noise

D Interference

E Overhead

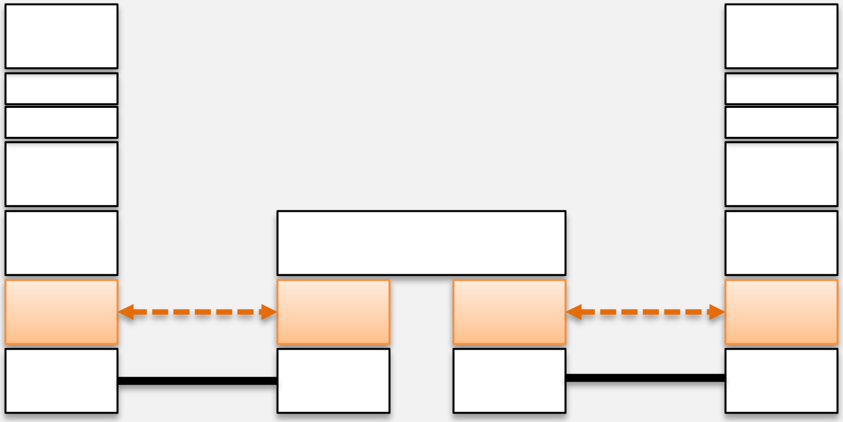
F Protocol

G Distance

DLC - General Concepts

Data Link Layer Perspective

- Second of the OSI Model
- Utilizes (unreliable) bitpipe from PHY layer
- Provides service to Networking layer
- Equipment: switch
- PDUs: “Frames”



Data Link Layer Services

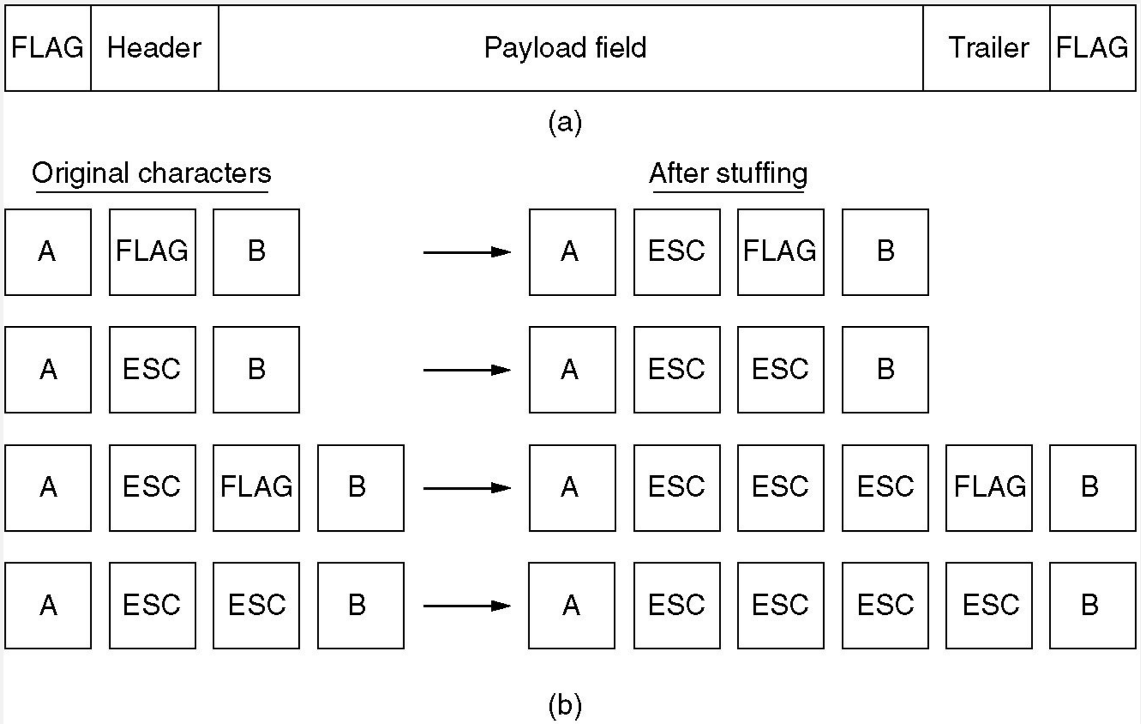
- Services Provided to the Network Layer
  - **Framing**
    - Logical bit groupings (header/trailers) – more use at higher layers
  - **Error Control**
    - Error control overlaps with physical layer (bit errors 0->1)
    - DLC – retransmission strategies
  - **Flow Control**
    - Matches dissimilar endpoint processing speeds (overwhelm)
    - Slow receiver should not be swamped by fast sender
    - Also a backward error correction mechanism
  - Optionally, mediate contention for shared medium
    - Such as Ethernet

Utilizing Unreliable Bitpipe - Framing

- *“Sometimes sender sends bits, sometime not”*
- How does the receiver (receiving DLC) know when bits are being received, and when not?
  - Must create own (DLC layer) convention (protocol) about transmission of bits
  - Start with **preamble**, end with **conclusion**
  - Creates logical groups at DLC layer - “frames”
  - Also serves as logical groups to encapsulate higher layer PDUs
    - Higher layers want service in multiple-byte chunks
    - Logical units determined by the logic of the higher layers
- Frame delineation – 1) must start with (and end with) an easily recognizable set of bits 2) that are unlikely to arise randomly due to noise

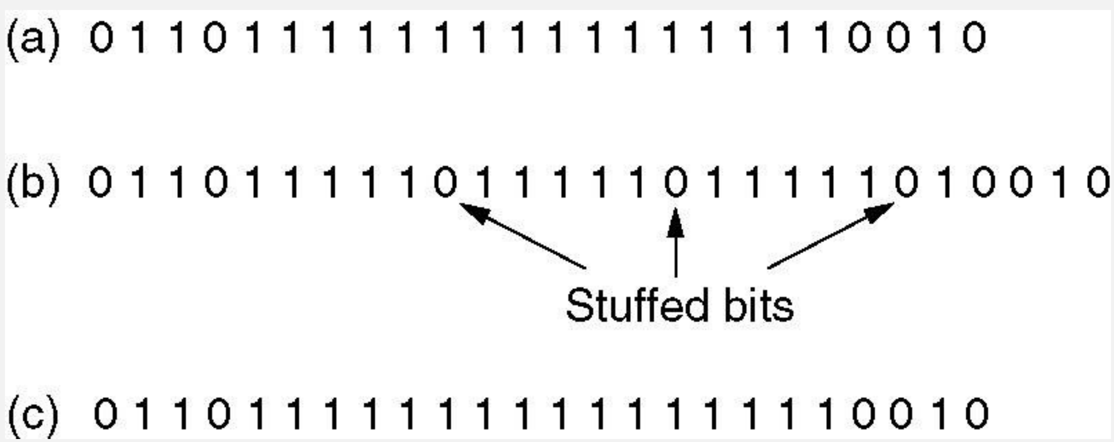
Framing with FLAGs (1st way)

- FLAG is a distinctive bit pattern
  - Such as 01111110 (01^60)
  - Must not appear inside data

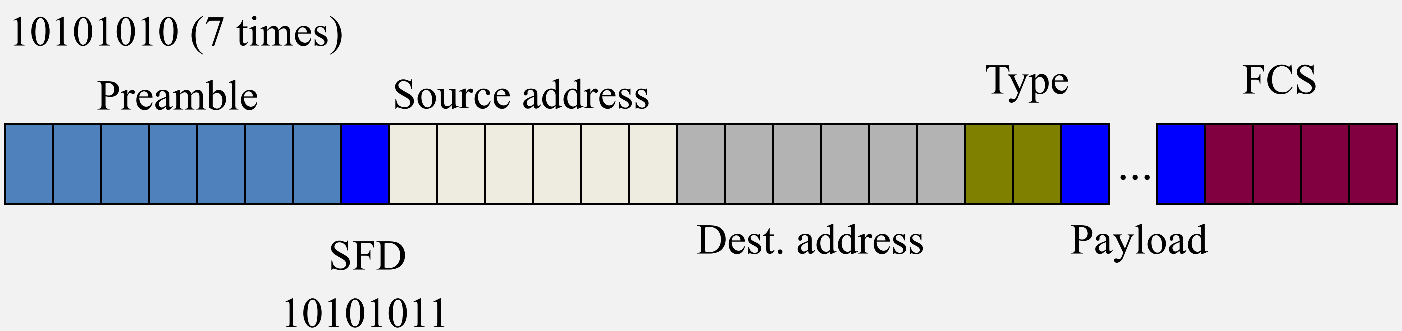


Bit Stuffing (2^{nd} way)

- Bit stuffing method
  - Delineate by pattern of many bits
  - Prevent pattern from occurring in data by few bits
  - Must be completely reversible, *i.e.* destuffable



Ethernet Frame



Data Link Layer Services

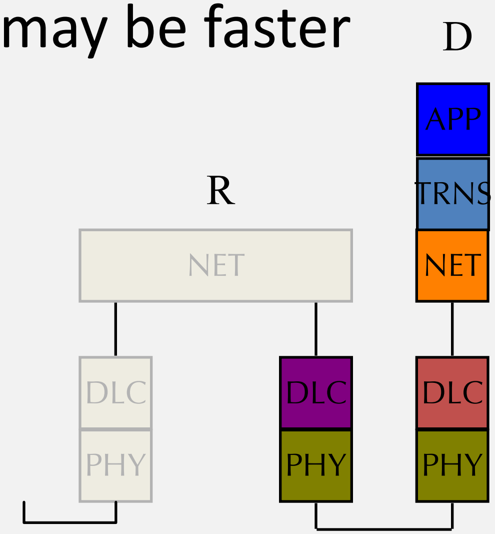
- Services Provided to the Network Layer
  - **Framing**
    - Logical bit groupings – more use at higher layers
  - **Error Control**
    - Error control overlaps with physical layer (bit errors 0->1)
    - DLC – retransmission strategies
  - **Flow Control**
    - Matches dissimilar endpoint processing speeds (overwhelm)
    - Slow receiver should not be swamped by fast sender
    - Also a backward error correction mechanism
  - Optionally, mediate contention for shared medium
    - Such as Ethernet

Error Control

- Errors happen
- Detection
  - Can at least catch errors
  - ARQ strategies may come in
- Correction (Forward Error Correction - FEC)
  - Better if we can
  - Sometimes essential due to link characteristics
- Involves error codes
  - Coding, not encryption

Flow Control

- Potential data generation and transmission may be faster than consumption
  - Receiver might be "overwhelmed"
- Fast sender, slow receiver (root cause)
  - Also at E2E layer (transport, or app)
- Realistically, must stop and wait during transmission
  - Channel idle for some fraction of time
  - Possible to proactively use some of this time
  - Windowed flow control



Flow Control vs. Error Control

- Flow control → if sender "gets ahead" of receiver, must retransmit
- Error control → may need to retransmit because of error in channel as well
  - Error must be detected
  - Only non-erroneous copy at the sender
  - Receiver must request a repeat
  - *A*utomatic *R*epeat Re*Q*uest (ARQ)
- But – sender needs to know
  - Requests are sometimes implicit

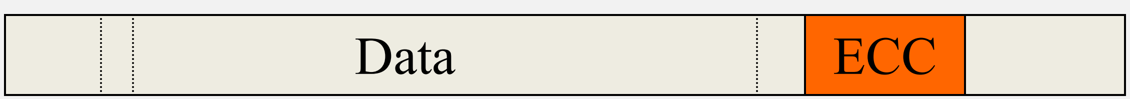
Error Control

Error Control

- Errors happen
- Detection
  - Can at least catch errors
  - ARQ strategies may come in
- Correction (FEC)
  - Better if we can
  - Sometimes essential due to link characteristics
- Involves error codes
  - Coding, not encryption

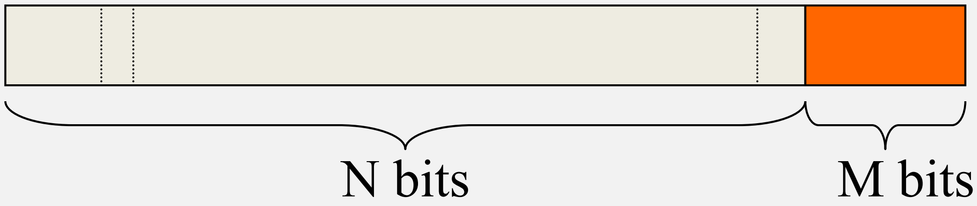
Fundamental Concepts - Redundancy

- Some data is actual data for error control
  - Transmitted data: Could be data + metadata
  - More data introduced expressly for the purpose of error control
  - "Metadata" for error control
  - Data and metadata must be distinguished
    - Usually a framing issue
  - Metadata introduces overhead
  - Only data is "useful" outside error control



Fundamental Concepts - Rate

- Overhead → "rate" of a coding scheme
  - Rate of useful data per encoded data, < 1.0
- Higher rate is more efficient
- Unfortunately, less likely to be good coding
- Expressed as a ratio of bits
  - Over long periods, if not constant



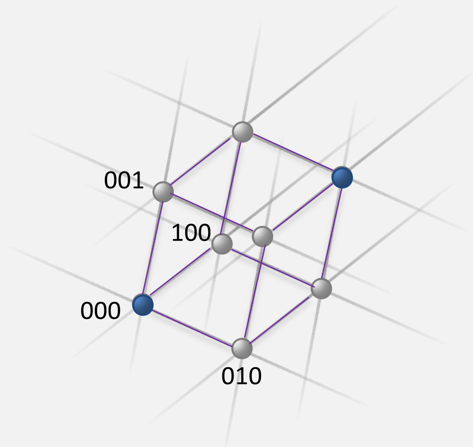
Fundamental Concepts – Block

- If a coding scheme always codes *n* bits to *n+m* bits,
  - We speak of *n* bit blocks that are independently coded
  - The code is called a block code

- Alternative – "stream" type coding
  - Convolution

Fundamental Concepts - Distance

- Data given to error control can be anything
  - 2^n "datawords"
  - For each, a unique *m* bit code
  - "Codeword" (encoded data) is *n+m* bits
- Hamming distance is a measure of strength
- d_{min} = minimum Hamming distance between any two codewords
- To guarantee detection of t errors
  - $d_{min} > t$
- To guarantee correction of t errors
  - $d_{min} > 2t$



Goal: maximize d_{min}

Efficient Use of Rate

- Lower rate means more overhead bits
  - But does not necessarily translate into higher distance
- Choosing appropriate codewords "smartly" can increase distance

$0 \rightarrow 00,\ \ 1 \rightarrow 10$
(does not increase robustness)

$0 \rightarrow 00,\ \ 1 \rightarrow 11$
(does)

$n = 1, m = 1$

$n = 2, m = 1$

$00 \rightarrow$
$01 \rightarrow$
$10 \rightarrow$
$11 \rightarrow$

?

Simple Repeat (coding example)

- Correction is possible, but rate suffers
- "*k* out of 2*k*-1" kind of voting process

- E.g., if each bit is sent seven times,
  - 1 bit blocks
  - 2^1 datawords ('0' and '1')
  - Only 2 valid codewords ('0000000' and '1111111')
  - Hamming distance $d_{min} = 7$
  - $t < 7$ if only detection is attempted (up to 6)
  - Up to 3 if correction is attempted from any seven-bit combination

Parity

- XOR all bits of the message
- Append the resultant bit to the word (m = 1)
- Detects *all* single bit errors
  - In fact, can detect all odd number of bits errors
  - But can *not* correct
- Unfortunately, chances of errors increases

$$s_1\ s_2\ \ldots\ s_n \longrightarrow \boxed{\text{Coder}} \longrightarrow s_1\ s_2\ \ldots\ s_n\ c$$

$c = s_1 \oplus s_2 \oplus \ldots \oplus s_n$, the modulo-2 sum

Horizontal and Vertical Parity

- Arrange message bits in a grid
- Parity bits for each row, each column
- Detects all single, double, triple errors
- Some but not all quadruple errors
  - Which ones?
- **Can it correct errors? (up to 1)**

Message = [s_{ij}]

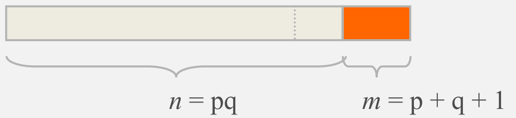
$c_i = s_{i1} \oplus s_{i2} \oplus \ldots \oplus s_{ip}$

$r_j = s_{1j} \oplus s_{2j} \oplus \ldots \oplus s_{qj}$

p

q

s_{11}	s_{12}	s_{13}	s_{14}	c_1
s_{21}	s_{22}	s_{23}	s_{24}	c_2
s_{31}	s_{32}	s_{33}	s_{34}	c_3
s_{41}	s_{42}	s_{43}	s_{44}	c_4
r_1	r_2	r_3	r_4	c



Trade-off? Rate is decreased … (efficiency)

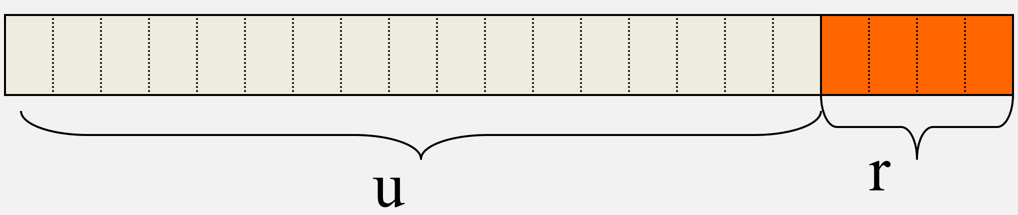
CRC – In Plain Words

- Let dataword bits be *U* (viewed as a binary number)
- Define a generator bit sequence *G* of *t*+1 bits

  Both sender and receiver knows *G*
- Compute a trailer *R* of *t* bits based on *U* and *G*, s.t. **<*U*, *R*>** is exactly divisible by *G* (using modulo-2 arithmetic)

- Can detect all up to *t* bits **bursty errors…**
  … but not the generator sequence itself

  => **Bursty:** consecutive bit flippings (frequent on Internet)



Calculating CRC Example

want:

$U \cdot 2^t \oplus R = nG$

equivalently:

$U \cdot 2^t = nG \oplus R$

thus:

if we divide $U \cdot 2^t$ by G, the remainder is R

$$\boxed{R = remainder\left[\frac{U \cdot 2^t}{G}\right]}$$

Long Division – but addition & subtraction are XOR

$t = 3$

G U

```
              101011
     1001 )101110000
           1001
            101
            000
            1010
            1001
             110
             000
             1100
             1001
              1010
              1001
               011
```

R

XOR

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

G = 101, t = 2, U = 111011

CRC – Polynomial View

Message polynomial

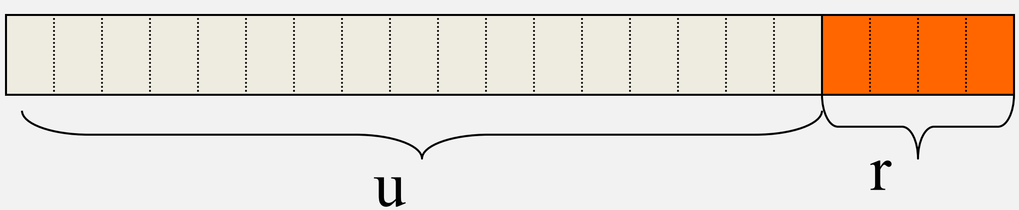
$$\mathrm{u}(x) = u_0 + u_1 x + u_2 x^2 + \ldots + u_{n\text{-}1}\, x^{n\text{-}1}$$

CRC polynomial

$$\mathrm{r}(x) = r_0 + r_1 x + r_2 x^2 + \ldots + r_{m\text{-}1}\, x^{m\text{-}1}$$

Code polynomial

$$\mathrm{c}(x) = \mathrm{u}(x)\, x^m + \mathrm{r}(x)$$



CRC as Remainder of a Division

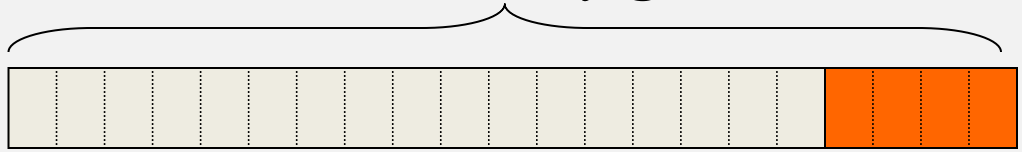
Generator polynomial :

$$g(x) = 1 + g_1 x + \ldots + g_{m-1} x^{m-1} + x^m$$

Encoding Rule:

$$r(x) = \text{Rem}\left\{ \frac{u(x)\, x^m}{g(x)} \right\}$$

Divisible by **g**



Long Division Example

n=3 and m=3

g(x) = 1 + x^2 + x^3

u(x) = 1 + x^2

c(x) = ?

$$
\begin{array}{r|l}
x^3 + x^2 + 1 & x^5 + \qquad x^3 \\
 & x^5 + x^4 + \qquad x^2 \\
\hline
 & \qquad x^4 + x^3 + x^2 \\
 & \qquad x^4 + x^3 + \qquad x \\
\hline
 & \qquad\qquad x^2 + x
\end{array}
$$

Remainder $\rightarrow$ $x^2 + x$

Review: DLC Error Control

Error Control in DLC Layer Show Correct Answer

TRUE or FALSE: With error control in the DLC layer, there is no need for further reliability mechanisms in the upper layers, since errors can be fully handled in the DLC layer.

A	True
B	False

Review: CRC Bits

CRC Bits

TRUE or FALSE: A CRC code with a 10-bit generator can detect all 10-bit errors.

A	True
B	False

Running Summary

- DLC - framing, flow control, error control
  - Optionally medium access
- Error control
  - Metadata must be added to enable error detection/correction
  - May be integrated with encoding of data – overlap with physical layer
  - Block codes
- Encapsulates error detection/correction largely within DLC and physical layers