COMP90007 Internet Technology

Week4

Yiran (Scott) Ruan

Email: yrrua@unimelb.edu.au

Website: https://yiranruan.github.io

Two important theorem

- Nyquist theorem
  - noiseless channel
- Shannon theorem
  - noise-related channel

Bandwidth

- **Frequency**: the number of oscillations per second of a wave is called frequency, measured in Hertz
- The **second concept about bandwidth**: the frequency range in the given medium – continuous signal

Nyquist’s theorem (**Noiseless**)

- Nyquist proved that if an arbitrary signal has been run through a low-pass filter of bandwidth *B*, the filtered signal can be completely reconstructed by making only 2*B* (exact) samples per second.
- Sampling the line faster than 2*B* times per second is pointless because the higher-frequency components that such sampling could recover have already been filtered out.
- If the signal consists of *V* discrete levels, Nyquist’s theorem states:

Maximum data rate = $2B \times \log_2 V \ \ bits/sec$

Nyquist's theorem (**Noiseless**)

- Nyquist theorem (noiseless channel)
  - Maximum data rate = $2B \times \log_2 V \;\; bits/sec$
  - **If a signal has bandwidth B, then the signal can be fully reconstructed by sampling with at least 2B rate.**
  - Increase the bandwidth B can increase the data rate.
  - If signal has $\mathrm{V}\ levels$, each symbol can represent $\log_2 V\ bits$.

Symbol Rate

- One symbol (signal element) can represent multiple bits (data elements)
- Symbol Rate (Baud Rate): number of signal changes per second

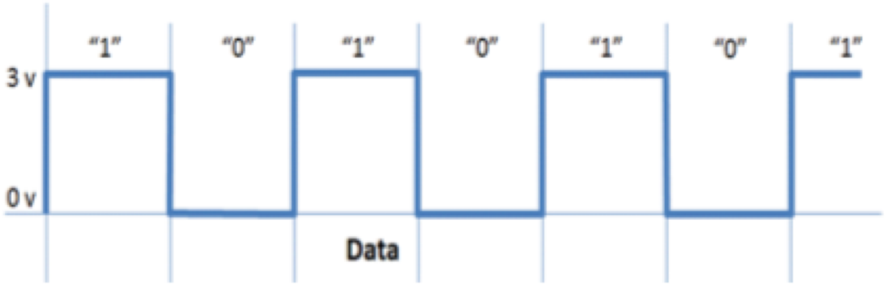


Figure 1. Data bits where logical "0" and "1" are represented by 0 volts and 3 volts respectively

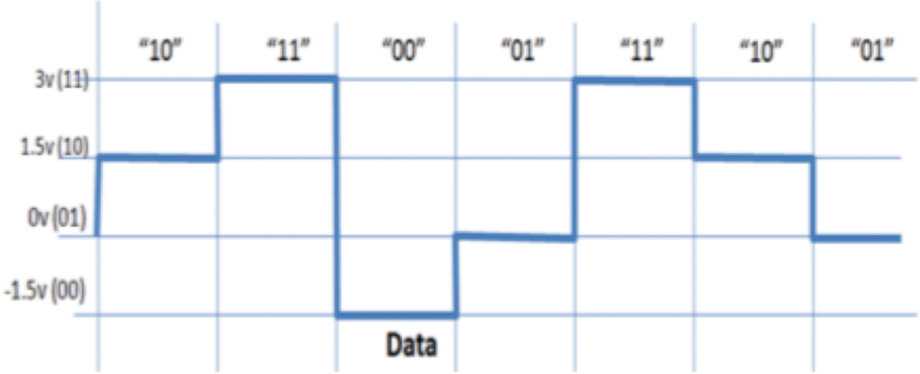
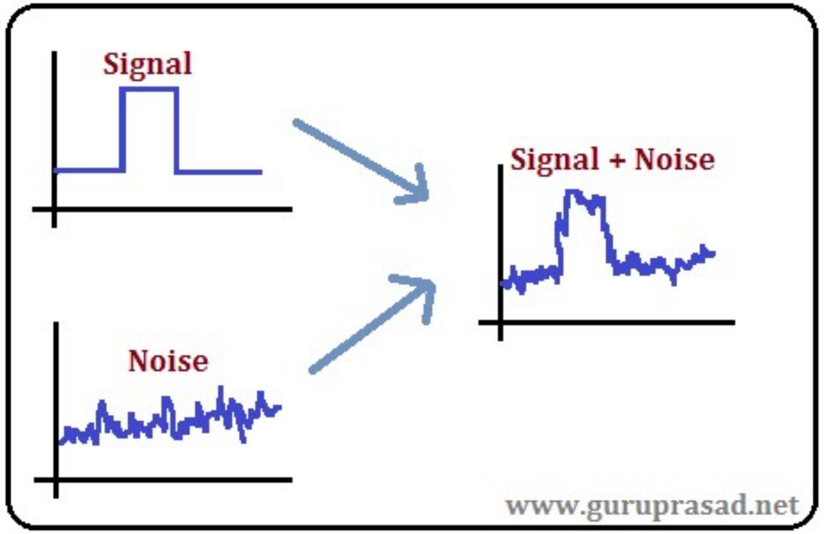


Figure 2. Four signaling levels per clock cycle can represent two data bits.

Shannon's theorem (**noise-related**)

- This theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the noise (N):

$$\text{Maximum data rate} = \underset{\substack{\uparrow \\ \text{How fast signal} \\ \text{can change}}}{B} \times \log_2(1 + \underset{\substack{\uparrow \\ \text{How many levels} \\ \text{can be seen}}}{S/N})\ bits/sec$$



Question 1 (Sampling)

- Consider a **telephone signal** that is bandwidth limited to 4 kHz.
  - (a) At what rate should you sample the signal so that you can completely reconstruct the signal?
  - (b) If each sample of the signal is to be encoded at 256 levels, how many **bits/symbol** are required for each sample?
  - (c) What is the minimum bit rate **required** to transmit this signal?

Solution 1 (Sampling)

- (a) At what rate should you sample the signal so that you can completely reconstruct the signal?
- By Nyquist's Theorem:

  min. sampling rate = 2×4000 = 8 kHz = 8000 samples/s

Solution 1 (Sampling)

- (b) If each sample of the signal is to be encoded at 256 levels, how many **bits/symbol** are required for each sample?

- 256 possible values per sample requires $\log_2 256 = 8\ bits/sample$

Solution 1 (Sampling)

- (c) What is the **minimum** bit rate required to transmit this signal?
- 8 bits/sample × 8000 samples/s = 64 kbps

Question 2 (Sampling)

- Is the Sampling theorem true for optical fibre or only for copper wire?

Solution 2 (Sampling)

- The Sampling theorem is a property of mathematics and has nothing to do with technology.
- It states that if you have a function whose Fourier spectrum (frequency domain representation) **does not contain** any frequency components (sines or) **above *f*, then by sampling at a frequency of *2f*,** you capture all the information there is. The Nyquist theorem is **independent of the transmission medium.**

- **Note**: You do not need to know the Fourier transform, law, etc.

Question 3 (Max Data Rate)

- Given a **noiseless 4 kHz channel**, What is the maximum data rate of the communications channel?

Solution 3 (Max Data Rate)

- The key word here is “noiseless” (related to Nyquist Theorem).
- A noiseless channel can carry an <u>arbitrarily large amount of information</u>, no matter how many levels of signals to use (i.e. there can be an infinite number of signaling levels due to noise-free.)
- In other word, in an noise-free channel, we could distinguish arbitrarily amount of levels of signals.

Solution 3 (Max Data Rate)

- Just send a lot of data per sample. Use many levels of signals, so that each symbol can carry many bits.
- Assume a **4 kHz channel, sampled at 8 kHz,** i.e. 8k symbols/sec**.** If each symbol is **16 bits**, the channel can send **128 kbps**. If each symbol is 1024 bits, the channel can send 8.2 Mbps.
- *(16/T -> 16*f -> 16bits * 8kHz -> 128kbps)*

Solution 3 (Max Data Rate)

- However, with a normal noisy 4 kHz channel, Shannon specifies a limit on the information rate on the channel known as its *capacity.*

Question 4.1 (Max Data Rate)

- The **bandwidth** of a television video stream is 6 MHz. How many bits/sec are sent if **four-level digital signals are used**? Assume a noiseless channel.

Solution 4.1 (Sampling)

- Using the Nyquist theorem, we should sample at 12 MHz or 12 million symbols/s.
- Four levels of signalling provide:
  - $\log_2 4 = 2\ bits/symbol\ (V = 4)$
- Hence, the total data rate is:
  - $12\ million\ symbols/s \times 2\ bits/symbol = 24\ Mbps$

Question 4.2 (Max Data Rate)

- Let’s consider about what happened in a noisy channel
- The **bandwidth** of a television video stream is 6 MHz. How many bits/sec are sent if **four-level digital signals are used**? <u>Now assume a S/N of 20db.</u>

Solution 4.2 (Sampling)

- Using the Shannon's theorem, we have:
  - $B \times \log(1 + \frac{S}{N}) = 6MHz \times \log(1 + 100)$
    $= 6MHz \times 6.65$
    $= 39.9MHz$

Question 5

- The following character encoding is used in a data link protocol:

A:	`01000111`	B:	`11100011`
FLAG:	`01111110`	ESC:	`11100000`

- Show the bit sequence transmitted (in binary) for the four-character frame payload *A B ESC FLAG,* when each of the following framing methods are used:

  (a) Character count

  (b) Flag bytes with byte stuffing

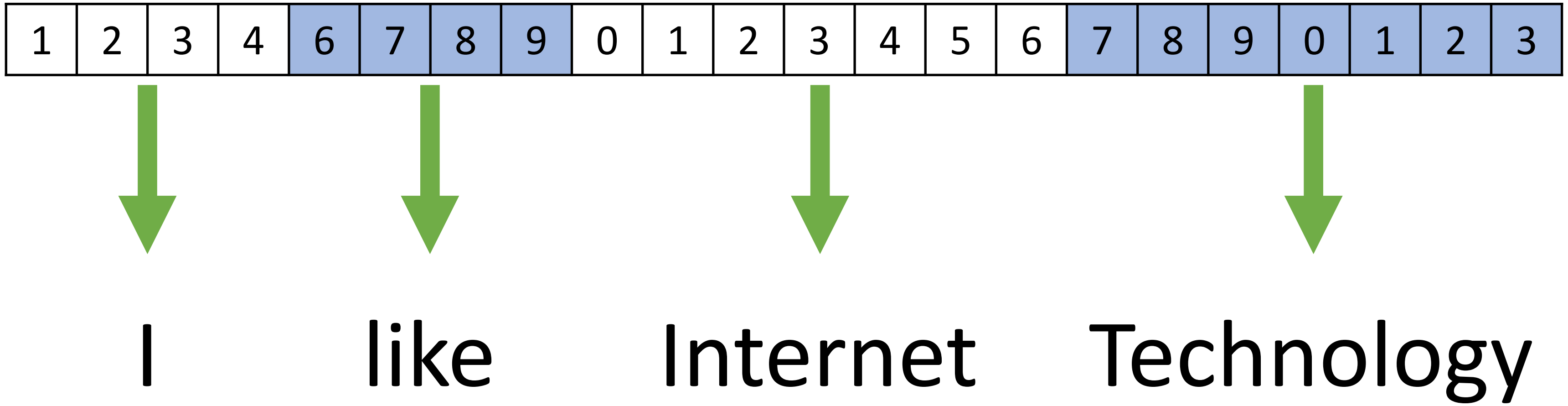
  (c) Starting and ending flag bytes, with bit stuffing

Character count

- Use a field in the header to specify the number of bytes in the frame.
- When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is.

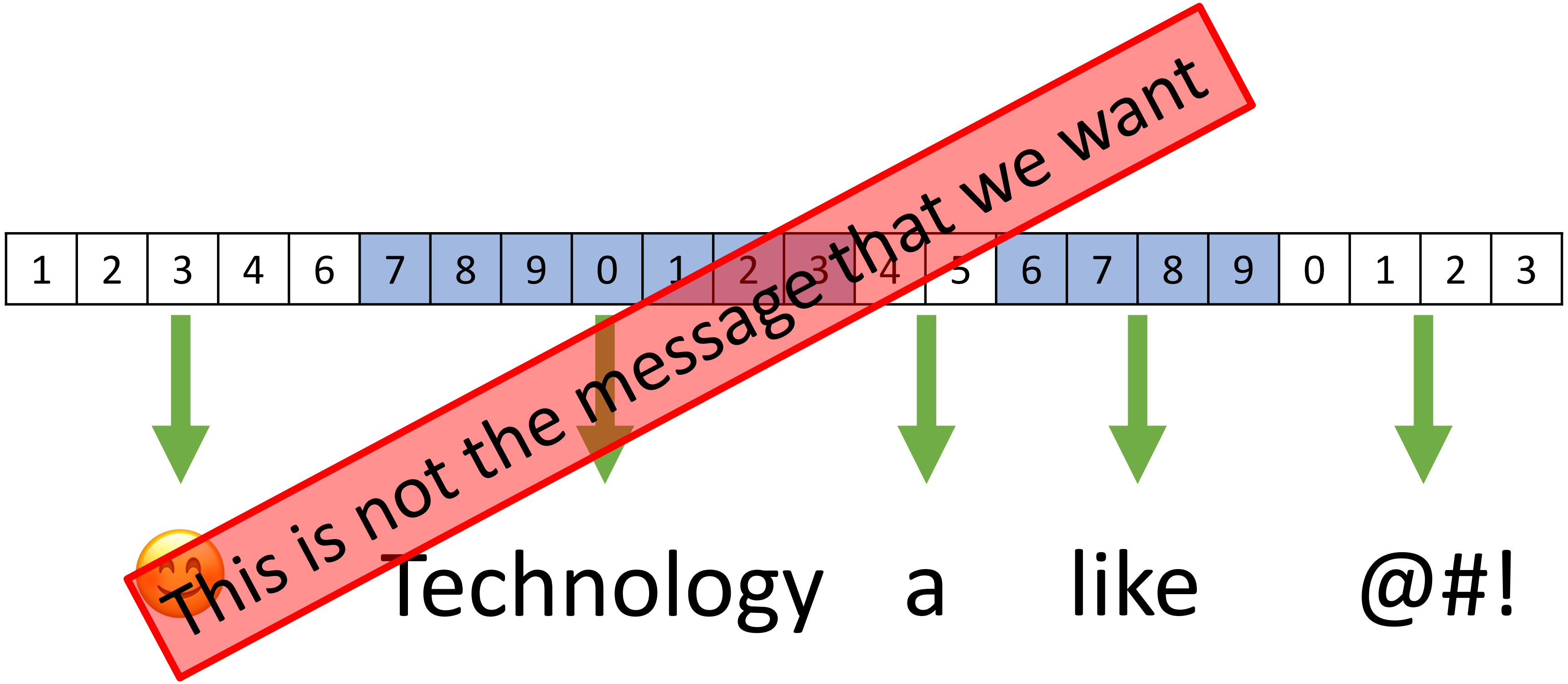
Receiver Issue

- Data



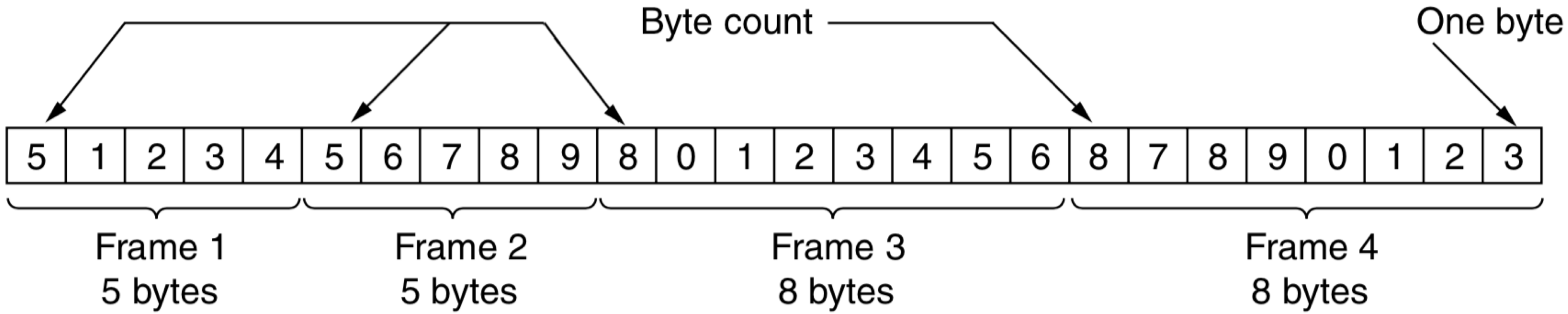
Receiver Issue

- Data



* Computer Network Page 197

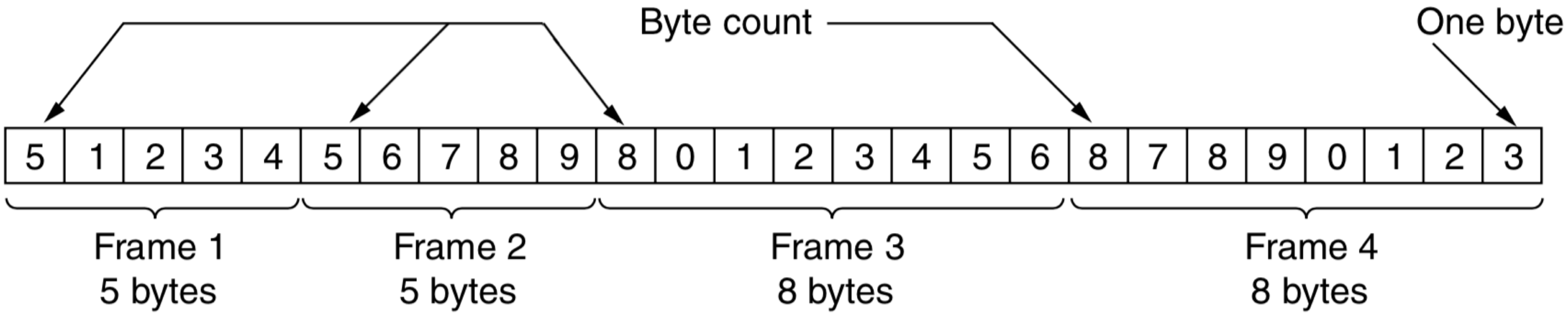
Character count



* Computer Network Page 197

Character count

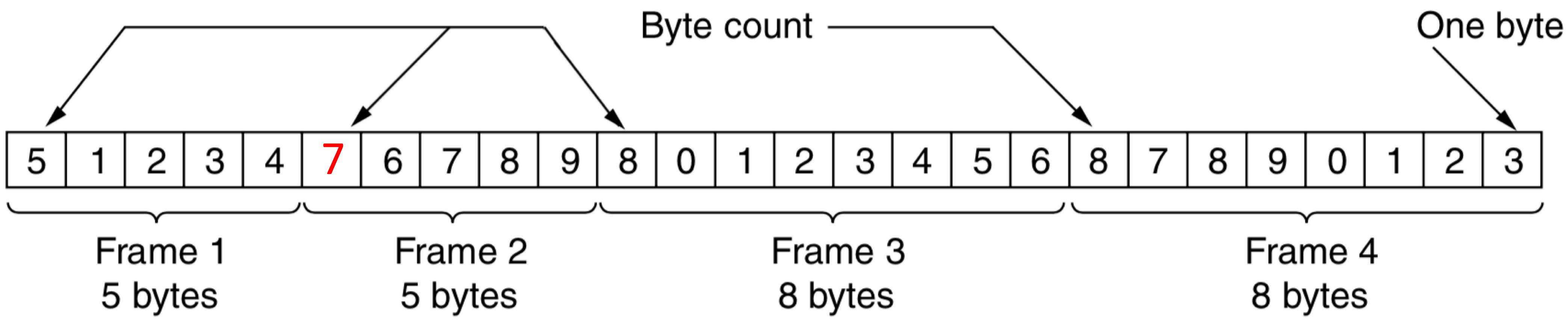
- Example:
  - The byte count of 5 in the second frame becomes a 7 due to a single bit flip.



* Computer Network Page 197

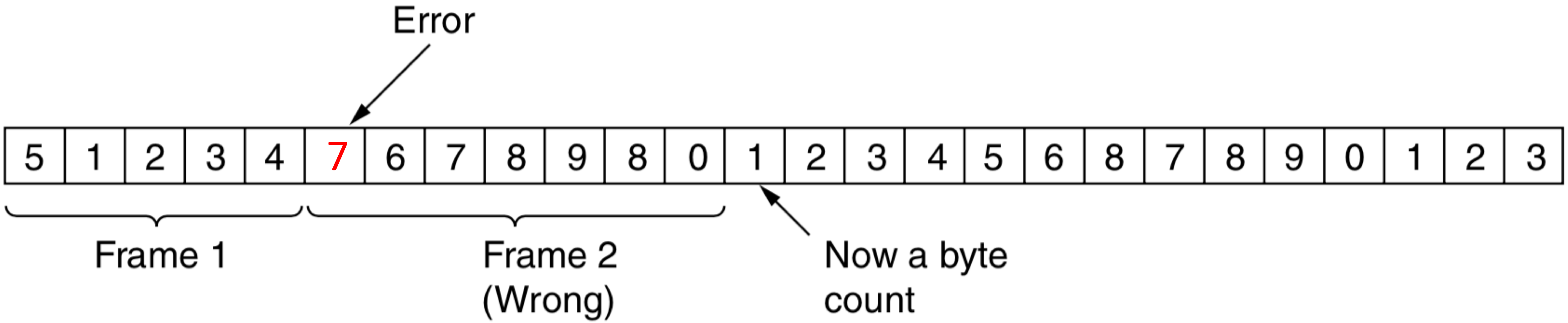
Character count

- Example:
  - The byte count of 5 in the second frame becomes a 7 due to a single bit flip.



Character count

- Example:
  - The destination will get out of synchronization. It will then be unable to locate the correct start of the next frame.



* Computer Network Page 197

Flag Byte

- The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. Often the same byte, called a **flag byte**, is used as both the starting and ending delimiter.

Flag Byte

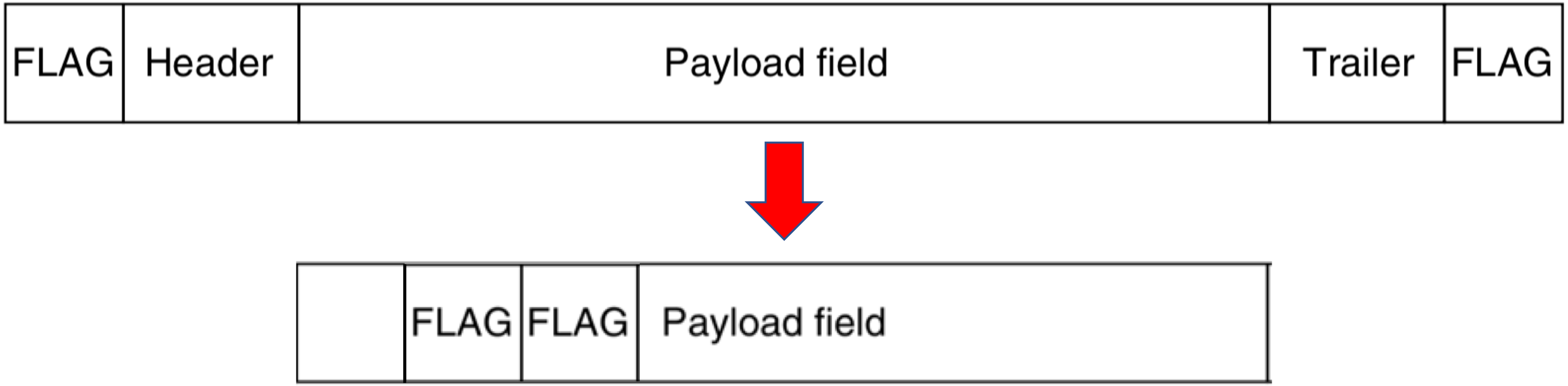
- **Two consecutive flag** bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for **two flag bytes** to find the end of the current frame and the start of the next frame.

FLAG	Header	Payload field	Trailer	FLAG

* Computer Network Page 198

Flag Byte

- Drawbacks: The flag byte occurs in the data

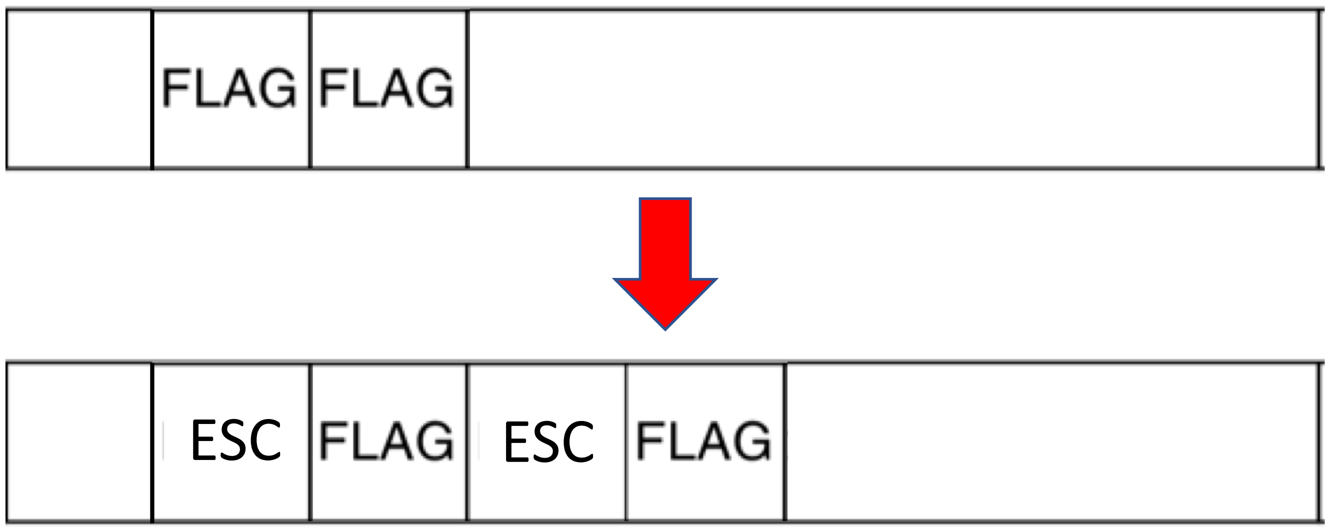


Data

* Computer Network Page 198

Byte Stuffing

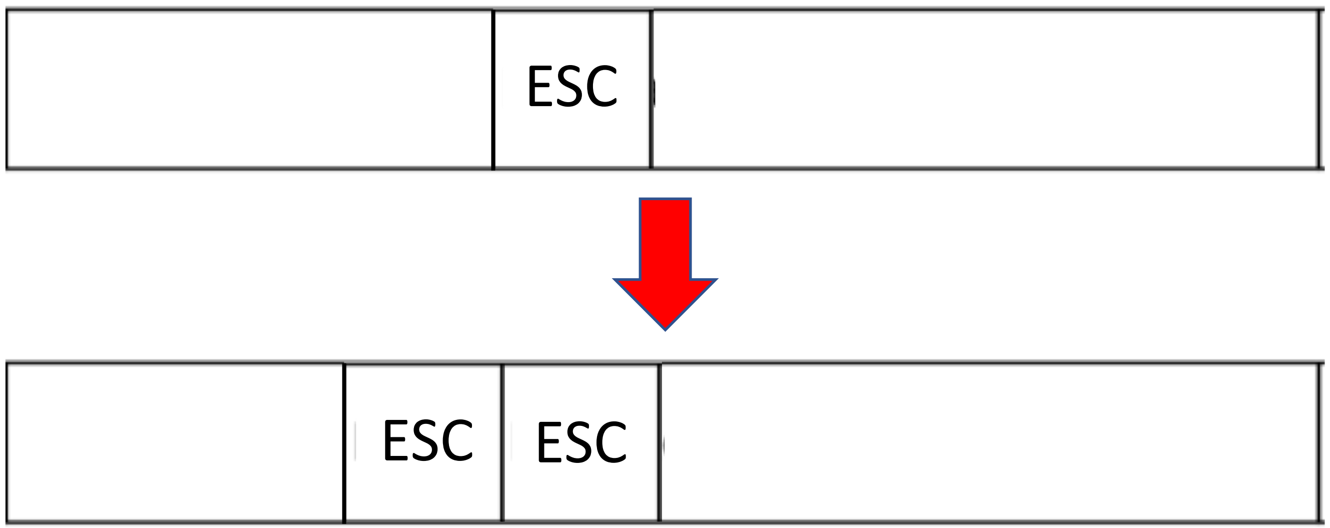
- Solution: Insert a special escape byte (ESC) just before each ‘‘accidental’’ flag byte in the data.



* Computer Network Page 198

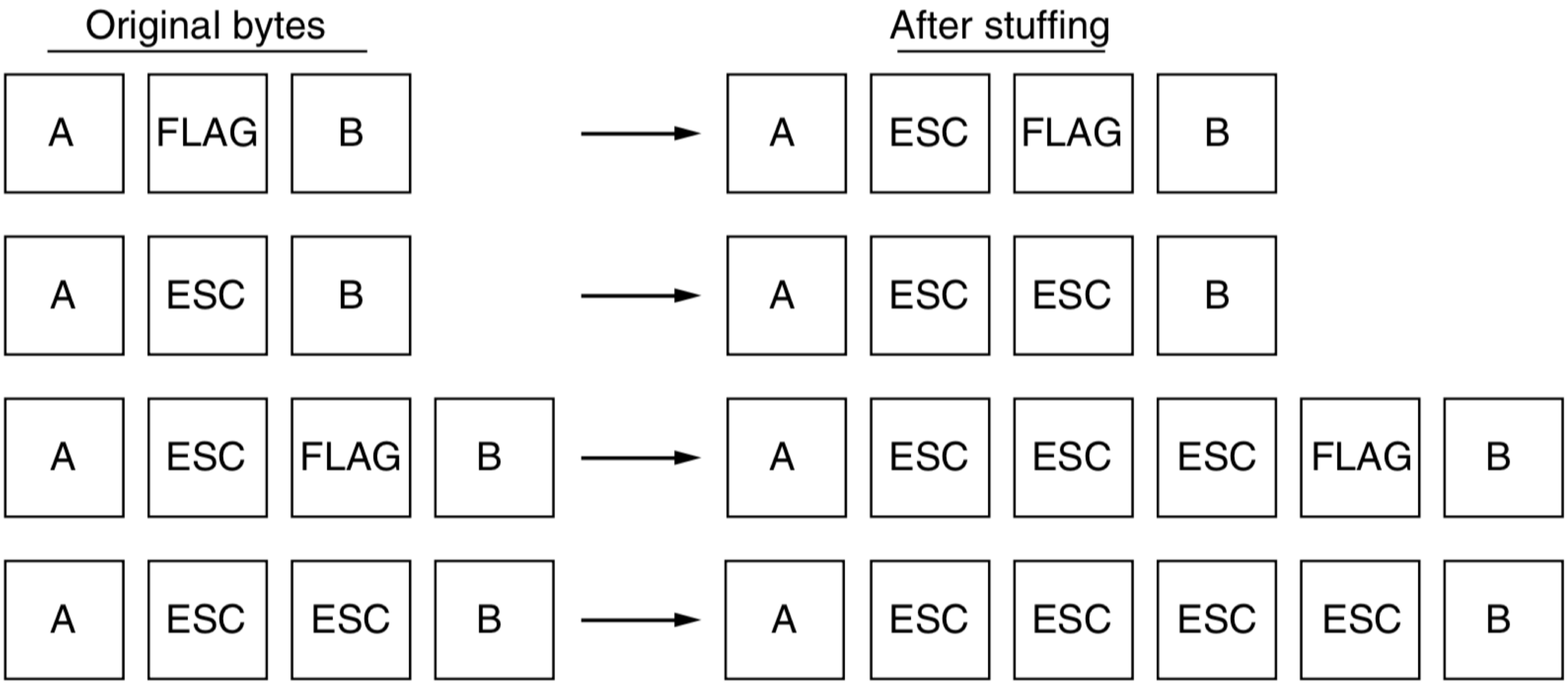
Byte Stuffing

- Solution: Insert a special escape byte (ESC) just before each ''accidental'' flag byte in the data.



* Computer Network Page 198

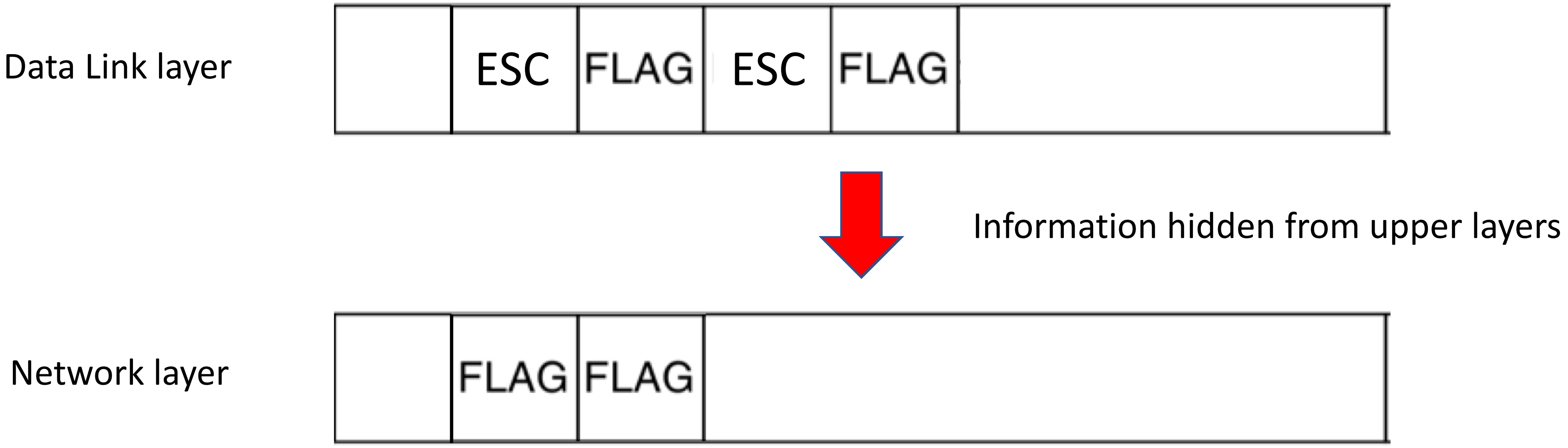
Byte Stuffing



* Computer Network Page 198

Byte Stuffing

- a framing flag byte can be distinguished from one in the data by the **absence** or **presence** of an escape byte before it. The data link layer on the **receiving end removes the escape bytes** before giving the data to the network layer.



* Computer Network Page 198

Byte Stuffing

- In Byte stuffing, we use a byte consisted by 8 bits to indicate FLAG or ESC existing in frame data, which cost expensive on space consumption.

* Computer Network Page 198

Bit Stuffing

- Each frame begins and ends with a special bit pattern, **01111110** or **0x7E** in hexadecimal. This pattern is a **flag byte**.
- Whenever the sender's data link layer encounters **five consecutive 1s** in the data, it automatically **stuffs a 0 bit** into the outgoing bit stream.

`01111110 ➔ 011111010`

- This **bit stuffing** is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

* Computer Network Page 199

Bit Stuffing

- Original data: 011011111111111111110010
- Bit stuffing: 011011111011111011111010010
- Receiver: 011011111111111111110010

* Computer Network Page 199

Question 5

- The following character encoding is used in a data link protocol:

A:	`01000111`	B:	`11100011`
FLAG:	`01111110`	ESC:	`11100000`

- Show the bit sequence transmitted (in binary) for the four-character frame payload *A B ESC FLAG,* when each of the following framing methods are used:
  (a) Character count
  (b) Flag bytes with byte stuffing
  (c) Starting and ending flag bytes, with bit stuffing

Solution 5

1. 00000101 01000111 11100011 11100000 01111110

   5 A B ‘ESC’ ‘FLAG’

2. 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110

   FLAG A B ESC ‘ESC’ ESC ‘FLAG’ FLAG

3. 01111110 01000111 110100011 111000000 011111010 01111110

   FLAG A B ‘ESC’ ‘FLAG’ FLAG

Question 6

- The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm as described in the lecture is used:

A B ESC C ESC FLAG FLAG D.

- What is the output after stuffing?

Solution 6

After stuffing we get:

A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D.

Assignment 1

- The assignment 1 has been released yet.
- The due date of this assignment is ***September 2nd Monday 11:30am***