Dr SABER Takfarinas
takfarinas.saber@dcu.ie

CA169
Networks & Internet

Link Layer
1-Frames

The Link Layer

- In the physical layer we looked at how digital information is transferred between two machines

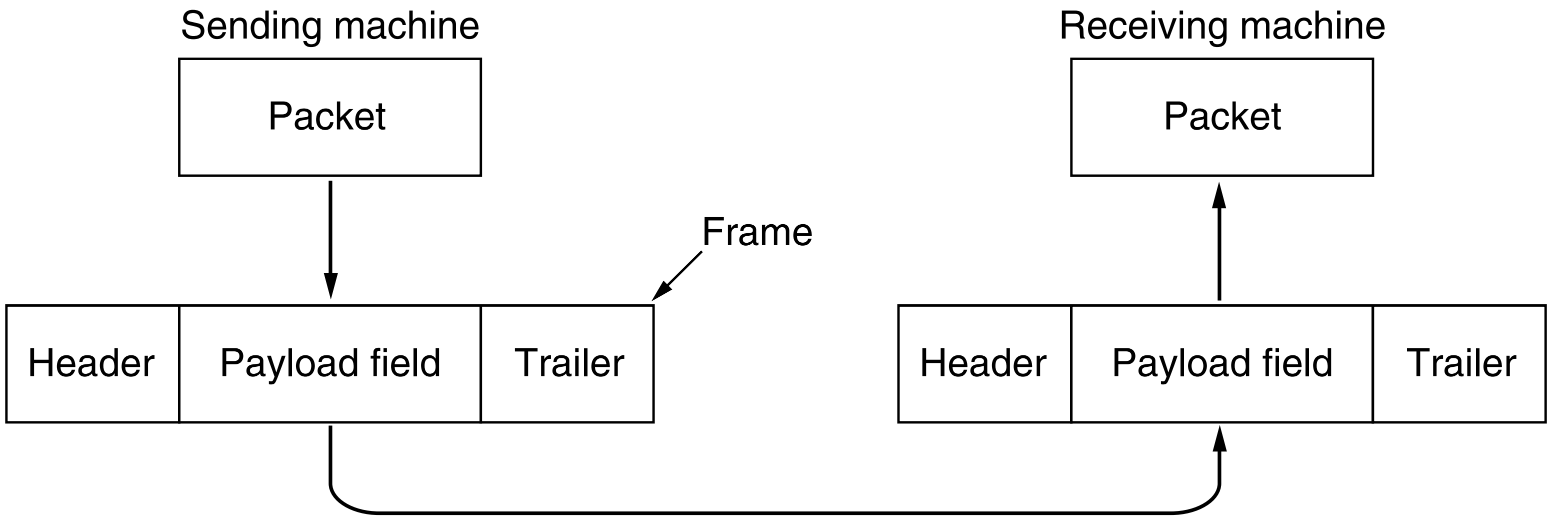
- In the link layer we are concerned with how to send messages between two machines
  - Messages are called **frames**

- The link layer has a number of functions, some important ones are
  - Framing messages
  - Dealing with transmission errors
  - Regulating the flow of data so slow receivers are not swamped

Frames

- So far we have talked about sending packets, in the link layer these are put inside another structure called a frame

- Frames have additional information added in headers (before the message) and trailers (after the message)



Framing

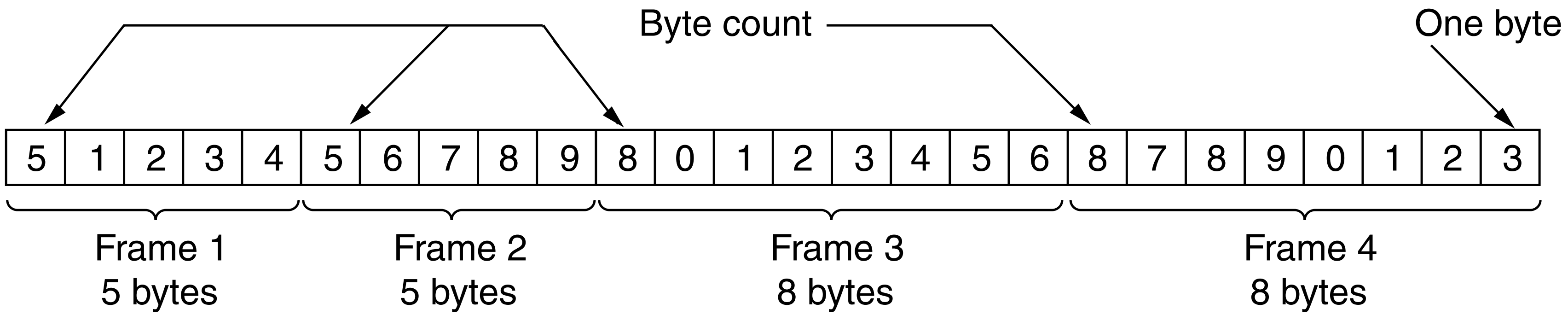
- Breaking up the bit stream into frames is more difficult than it at first appears

- A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth

- We will look at three methods:
  - Byte count
  - Flag bytes with byte stuffing
  - Flag bits with bit stuffing

Framing - Byte Count

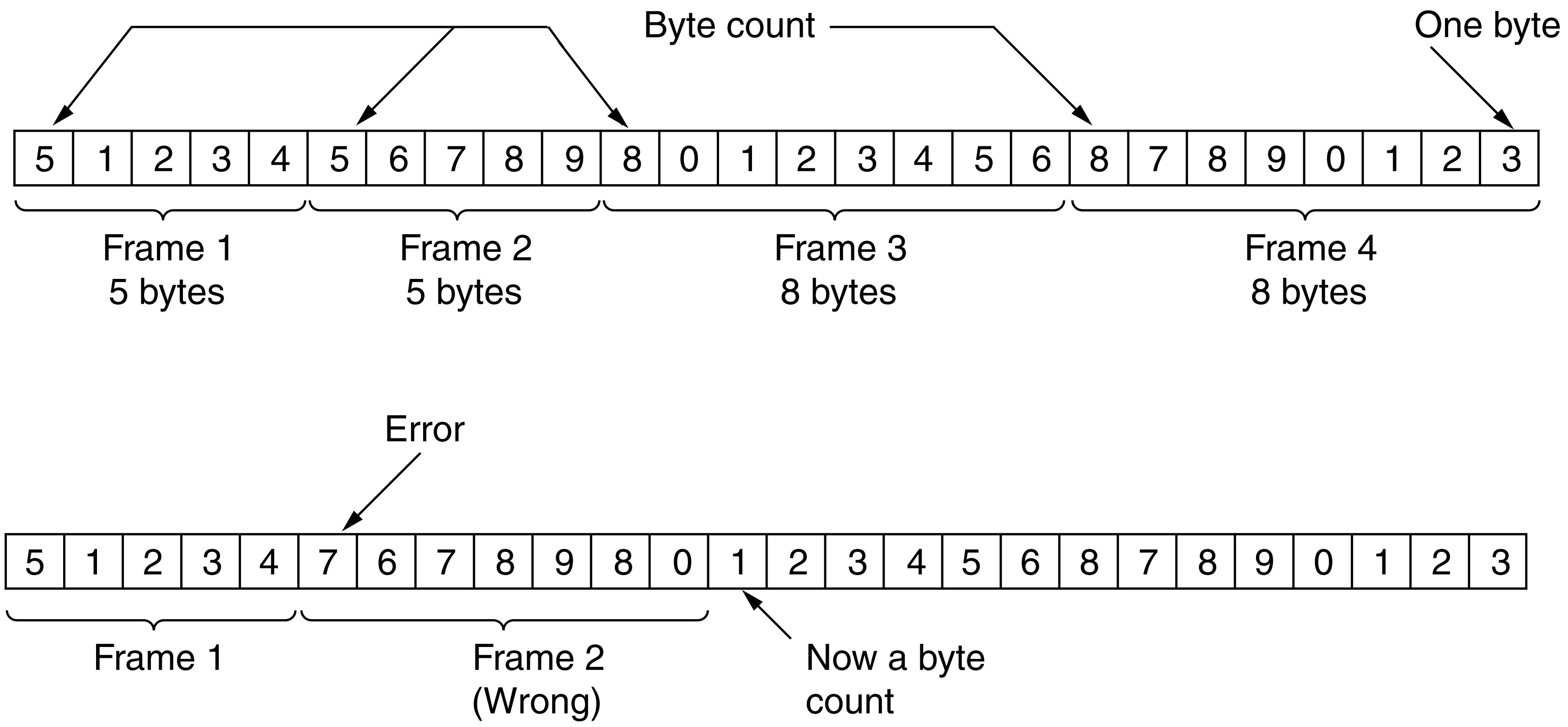
- One of the simplest and most efficient methods
- Simply start the frame by having the first number be the count of how many bytes are in the frame



5	1	2	3	4	5	6	7	8	9	8	0	1	2	3	4	5	6	8	7	8	9	0	1	2	3

Framing - Byte Count

- But what happens if the byte is corrupted in transmission?



Byte count

One byte

5 1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Frame 1
5 bytes

Frame 2
5 bytes

Frame 3
8 bytes

Frame 4
8 bytes

Error

5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Frame 1

Frame 2
(Wrong)

Now a byte
count

Framing – 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
- 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

Framing – Flag Byte

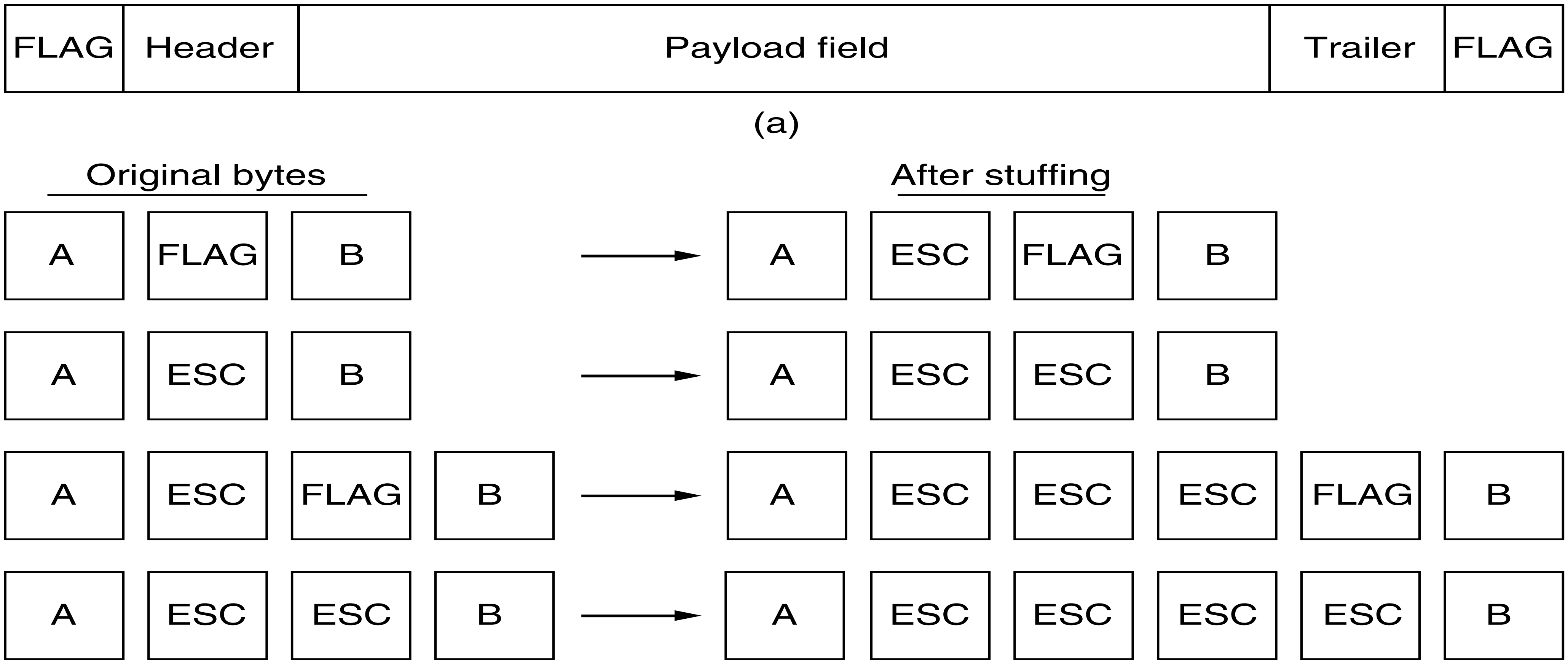
- What happens if the flag byte appears in the data being sent?
- One solution is to insert a special **escape byte** (ESC) before any accidental flag byte
- This way we can tell when a frame ends or the byte is just data
- The receiver then removes these flag and escape bytes from data it receives
- This process is called **byte stuffing**

Framing – Escape Byte

- What happens if the escape byte appears in the data being sent?

- The simple solution is that another escape byte is stuffed before it

Framing - Flag Byte



FLAG | Header | Payload field | Trailer | FLAG

(a)

Original bytes → After stuffing

A FLAG B → A ESC FLAG B

A ESC B → A ESC ESC B

A ESC FLAG B → A ESC ESC ESC FLAG B

A ESC ESC B → A ESC ESC ESC ESC B

Framing – Flag Bits

- Byte stuffing has a drawback, adding these bytes increases the size of the message a lot
    - Because each flag adds another byte

- The third method of framing uses single bits for stuffing so the increase in size is not as much

Framing – Flag Bits

- Each frame begins with the bit pattern $0\underbrace{111111}_{\text{six 1s}}0$

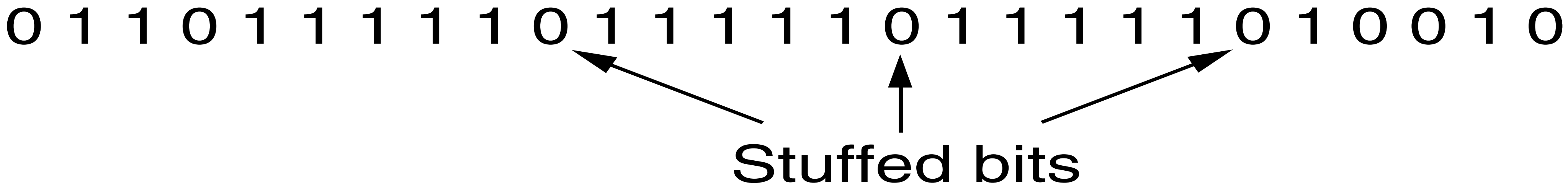
- Bit Stuffing
  - **Sender:** Whenever we see five 1s in our data we **insert** a 0
  - **Receiver:** Whenever we see five 1s we **remove** the following 0

- It is the same idea as byte stuffing but with less overhead

- The only problem is that the final frame could be of any **number of bits** but byte stuffing it is always a **number of bytes**

Framing – Flag Bits

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

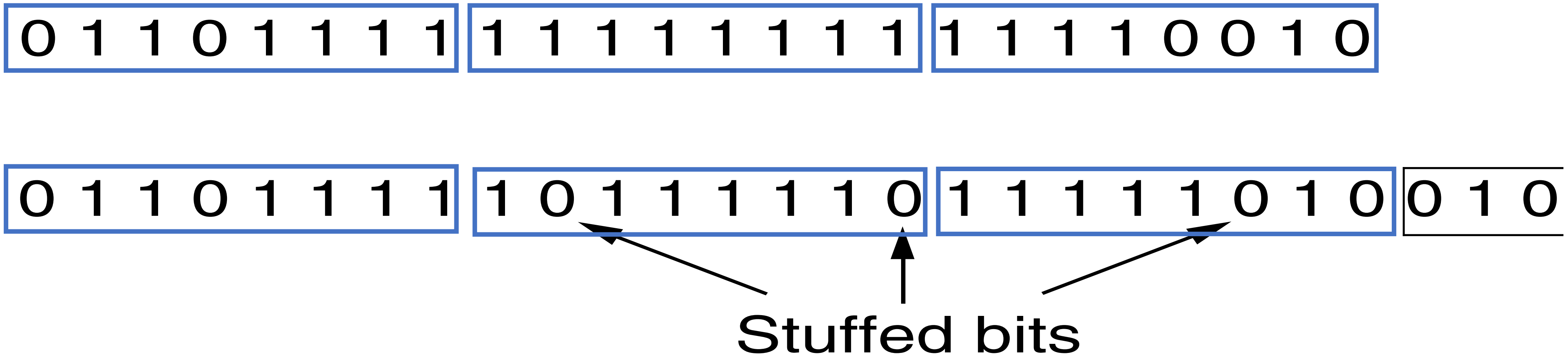


0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

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

Framing – Flag Bits



0 1 1 0 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 0 0 1 0

0 1 1 0 1 1 1 1 | 1 0 1 1 1 1 1 0 | 1 1 1 1 1 0 1 0 | 0 1 0

Stuffed bits

Physical Layer Framing

- There exists other methods of framing that we are not studying closely

- Some of them exploit knowledge of what is happening in the physical layer
  - If the physical layer is using a protocol such as 4B/5B
  - It means that we know that **certain byte sequences** cannot appear in the data
  - Therefore, we can use one of these Byte sequences to signal the start and end of frames