

Project DLC 1: Bit Stuffing

- ❑ Due midnight Feb. 25th.
- ❑ You implement two routines in `stuff.cpp`, `stuff()` and `unstuff()`.
- ❑ To compile, `cwkb dlc1`
- ❑ For submission, upload `stuff.cpp` and `diskout.txt`.
- ❑ All projects must be individual efforts.

CS 656

1

Stuff

```
void stack::stuff(bit_frame* stuffed_frame,
                 bit_frame* unstuffed_frame)
{
    zero_bits(stuffed_frame);
    int bitpos;
    // find length by converting the byte
    // starting at bit 8
    int unstuffed_length =
    8*bittobyte(8,unstuffed_frame->frame_bits);
    // this version only copies the frame;
    // student must add stuffing
    for(bitpos=0; bitpos<unstuffed_length;
        ++bitpos)
        stuffed_frame.frame_bits[bitpos] =
            unstuffed_frame->frame_bits[bitpos];
}
```

CS 656

2

Unstuff

```
void stack::unstuff(
    bit_frame*unstuffed_frame,
    bit_frame* stuffed_bits)
{
    int bitpos;
    // this version only copies the frame;
    // student must add unstuffing
    for(bitpos=0; bitpos<MAX_BIT_FRAME_SIZE;
        ++bitpos)
        unstuffed_frame.frame_bits[bitpos]=
            stuffed_bits->frame_bits[bitpos];
}
```

CS 656

3

Some Notes

- ❑ In the `stuff()` routine, the given `unstuffed_frame` already contains starting and ending flags; be careful not to stuff them.
- ❑ In the `unstuff()` routine, you don't have the information of original frame length, `unstuff_length`; you must look for the end flag.
- ❑ When you read the output of NW, you find, in addition to the body of email messages, many "mysterious bits" in frames. Those are the headers of higher layer protocols.

CS 656

4

Reading NW Output

```
*** starting discrete event simulation ***
```

```
tick 0 host 1
application layer sends to best-effort transport layer message:
031 001 001 002 001 000 Chicken Little was right~
```

```
tick 10 host 1
best-effort transport layer sends to network layer segment:
hex :2A 01 01 02 01 00 00 00 00 00 00 1F 01 01 02 01 00 43 68 69
char: *                               C h i
hex :63 6B 65 6E 20 4C 69 74 74 6C 65 20 77 61 73 20 72 69 67 68
char: c k e n   L i t t l e   w a s   r i g h
hex :74 7E
char: t ~
```

CS 656

5

```
tick 60 host 1
network layer sends to interface 1 packet queue for DLC
hex :32 01 01 02 01 0F 00 00 2A 01 01 02 01 00 00 00 00 00 00 1F
char: 2                               *
hex :01 01 02 01 00 43 68 69 63 6B 65 6E 20 4C 69 74 74 6C 65 20
char:                               C h i c k e n   L i t t l e
hex :77 61 73 20 72 69 67 68 74 7E
char: w a s   r i g h t ~
```

```
tick 100 host 1 interface 1
DLC layer sends to physical layer frame:
01111110 00111101 00000000 00000010 00000000 00000000 00000000 00000000
00110010 00000001 00000001 00000010 00000001 00001110 00000000 00000000
00101010 00000001 00000001 00000010 00000001 00000000 00000000 00000000
00000000 00000000 00000000 00011111 00000000 10000000 10000001 00000000
10000000 00100001 10110100 00110100 10110001 10110101 10110010 10110111
00010000 00100110 00110100 10111010 00111010 00110110 00110010 10010000
00111011 10110000 10111001 10010000 00111001 00110100 10110011 10110100
00111010 00111110 10010000 00011110 01011111 10
```

CS 656

6

```

tick 3389 host 2
interface 1 physical layer receives and passes to DLC:
01111110 00111101 00000000 00000010 00000000 00000000 00000000 00000000
00110010 00000001 00000001 00000010 00000001 00001110 00000000 00000000
00101010 00000001 00000001 00000010 00000001 00000000 00000000 00000000
00000000 00000000 00000000 00011111 00000000 10000000 10000001 00000000
10000000 00100001 10110100 00110100 10110001 10110101 10110010 10110111
00010000 00100110 00110100 10111010 00111010 00110110 00110010 10010000
00111011 10110000 10111001 10010000 00111001 00110100 10110011 10110100
00111010 00111110 10010000 00011110 01011111 10

```

```

tick 3390
host 2 DLC sends to network layer 1 packet:
hex :32 01 01 02 01 0E 00 00 2A 01 01 02 01 00 00 00 00 00 00 1F
char: 2 *
hex :01 01 02 01 00 43 68 69 63 6B 65 6E 20 4C 69 74 74 6C 65 20
char: C h i c k e n L i t t l e
hex :77 61 73 20 72 69 67 68 74 7E
char: w a s r i g h t ~

```

CS 656

7

```

tick 3400
host 2 network layer sends to transport layer segment:
hex :2A 01 01 02 01 00 00 00 00 00 00 1F 01 01 02 01 00 43 68 69
char: * C h i
hex :63 6B 65 6E 20 4C 69 74 74 6C 65 20 77 61 73 20 72 69 67 68
char: c k e n L i t t l e w a s r i g h
hex :74 7E
char: t ~

```

```

tick 3420
host 2 transport layer sends to application layer message:
031 001 001 002 001 000 Chicken Little was right~

*** no more events in DES event list -- simulation terminated

```

CS 656

8

NW Structures

□ Transport-Layer Segment Format:

```
typedef struct {
    byte size;
    byte source_net;
    byte source_host;
    byte dest_net;
    byte dest_host;
    byte seqno;
    byte ackno;
    byte window;
    bit ack;
    bit syn;
    bit fin;
    message msg;
} segment;
```

□ Network-Layer Packet Format

```
typedef struct pkt {
    byte size; // including
                // header
    byte source_net;
    byte source_host;
    byte dest_net;
    byte dest_host;
    byte ttl;
    byte pkt_protocol;
    byte reserved;
    segment seg;
} packet;
```