

# ECE1150 ASSIGNMENT5

Yinhao Qian @ University of Pittsburgh

```
%please ignore this block
answer = @(num,unit) fprintf("<strong> ANSWER: %s [%s]" + ...
    " </strong>\n",mat2str(num),unit);
question = @() eval("clearvars -except answer question");
```

## Q1

Frequency division duplex requires two frequencies, and quadrature amplitude modulation ,although requires mutiple amplitudes and phases, only requires one frequency. **Totalling up to two frequencies.**

## Q2

```
question();
4e3*8+150*(8-1);
answer(ans,"Hz");
```

ANSWER: 33050 [Hz]

## Q3

### ALOHA

- Transmit a frame whenever you want.
- If there is a collision, no acknowledgement will be received.

### Slotted ALOHA

- Synchronizing senders to slots, which reduces collisions.
- Fixing the starting and ending times to specific values. (time slots)
- When a node has frame to send, it waits till the beginning of the next slot and transmits the frame.
- If collision occurs, a node transmits the frame in subsequent slot with certain probability.
- Repeat attempts until frame is successfully received.
- Suitable for low traffic.

## CSMA

- Carriers will sense multiple access.
- If sensed channel is idle, transmit entire frame.
- If sensed channel is busy, defer the transmission.

## CSMA/CD

- CSMA with collusion detection.
- If collision is detected after transmission, devices ceases transmission.
- Each node involved in collision waits for a random delay before retransmitting the frame.

## Polling

- One node acts as master node
- Master node polls other nodes in a round robin fashion to check whether they have data to transmit.
- Node may or may not have data to send.
- Used in Bluetooth.

## Q4

```
question();
```

In slotted ALOHA, only one frame of transmission is allowed at a certain time stamp. The efficiency is directly proportional to the probability of success, which can be calculated using the following:

$$S = G * \exp(-G)$$

It is maximized when only one frame is transmitted during slot time:

```
G = 1;  
S = G*exp(-G);  
answer(S,"");
```

ANSWER: 0.367879441171442 []

## Q5

```
question();
syms M H B Ptx Pe;
t = (M+H)/B;%tranmission time
tp = (Ptx*(1-Pe)*M)/t%throughput
```

$$tp = -\frac{B M P_{tx} (Pe - 1)}{H + M}$$

```
eff = tp/B%efficiency
```

$$eff = -\frac{M P_{tx} (Pe - 1)}{H + M}$$

For Q6, for some reasons the symbolic evaluation algorithms in MATLAB, the negative sign seems to be preferred to be shown like the following:

$$(1 - p) = -(p - 1)$$

The answer might looks ugly, but they should be correct in value.

## Q6A

```
question();
syms p;%probability of tranmission
```

Since it's slotted ALOHA, the tranmission is sucessful if and only if this node A is tranmitting but other three nodes are not.

```
p_succeed = p*(1-p)^3;
```

Since it's the 5th slots, four attempts were failed.

$$p_{a5} = (1-p_{\text{succeed}})^4 \cdot p_{\text{succeed}}$$

$$p_{a5} = -p \left( p(p-1)^3 + 1 \right)^4 (p-1)^3$$

## Q6B

Since only one node can successfully transmit in one slot, the probabilities of either of multiple successful transmissions are mutually exclusive:

$$p_{\text{either4}} = p_{\text{succeed}} + p_{\text{succeed}} + p_{\text{succeed}} + p_{\text{succeed}}$$

$$p_{\text{either4}} = -4p(p-1)^3$$

## Q6C

It means that all transmissions are failed for the first two slots, which means none of nodes get to transmit:

$$p_{3f} = (1-p_{\text{either4}})^2 + p_{\text{either4}}$$

$$p_{3f} = \left( 4p(p-1)^3 + 1 \right)^2 - 4p(p-1)^3$$

## Q6D

A slot is effective as long as either nodes successfully transmit:

$$\text{eff} = p_{\text{either4}}$$

$$\text{eff} = -4p(p-1)^3$$

## Q7A

```

question();
m = 0b1011011u64;%[] message
l = ceil(log2(double(m)));%[] message length
g = 0b1101u64;%[] generator
n = ceil(log2(double(g)))-1;%[] generator length

```

To mimic the mod2 division programatically, I'll shift the generator all the way to the left, XOR it with the messenger, and shift the generator one bit to the right, and XOR it with the messenger again, etc..

```

g = bitsll(g,l-1);
m = bitsll(m,n);
for k = 1:l
    if bitget(m,l+n)
        m = bitxor(m,g);
    end
    m = bitshift(m,1);
end
crc = bitsrl(m,l);
answer(dec2bin(crc,n),"");

```

ANSWER: '001' []

## Q7B

```

m = 0b1011011u64;
l = ceil(log2(double(m)));
g = 0b1101u64;
n = ceil(log2(double(g)))-1;
g = bitsll(g,l-1);
m = bitsll(m,n);

```

I'll manually set the crc to the 3 bits on the right.

```

m = bitor(m,crc);
for k = 1:l
    if bitget(m,l+n)
        m = bitxor(m,g);
    end
    m = bitshift(m,1);

```

```
end
rem = bitsrl(m,1);
```

The remainder should be zero. If `rem==0` evaluates to true, it means no error.

```
answer(rem==0,"");
```

```
ANSWER: true []
```

## Q8A

```
question();
packet = 20e3*8;
delay_prop = 5e3/2.5e8;%[s] propagation delay
delay_tran = packet/1e8;%[s] transmission delay per packet
```

Since it's full-duplex and communications are bi-directional, packet-sending propagation and acknowledgement-receiving propagation can be combined into one.

### Cycle1:

Transmitter => Packet 1 => Receiver (Prop + Tran)

### Cycle2:

Transmitter <= ACK 1 <= Receiver (Prop)

### Cycle3:

Transmitter => Packet 2 => Receiver (Prop + Tran)

### Cycle4:

Transmitter <= ACK 2 <= Receiver (Prop)

```
t = delay_prop*4+delay_tran*2;%[s] times to complete service
tp = packet*2/t;
answer(tp,"bits/s");
```

```
ANSWER: 97560975.6097561 [bits/s]
```

## Q8B

In this case, two packets are transmitted non-stop as the window size is 2.

### Cycle1:

Transmitter => Packet 1 => Receiver (Prop + Tran)

### Cycle2:

Transmitter <= ACK 1 <= Receiver (Shar Prop)

Transmitter => Packet 2 => Receiver (Shar Prop + Tran)

### Cycle3:

Transmitter <= ACK 2 <= Receiver (Prop)

```
t = delay_prop*3+delay_tran*2;
tp = packet*2/t;
answer(tp,"bits/s");
```

ANSWER: 98159509.202454 [bits/s]

The continuous ARQ in this case is more efficient than the stop and wait ARQ.