Homework 1: Link Layer

Due: 11:59 PM, Oct 1, 2019

Contents

1 Problem 1

1.1

Use Hartley's Law:

```
C = 2B log_2(M) bits/s
C = 168Mbps = 168 * 10^6 bits/s
B = 21MHz = 21 * 10^6 Hz
M = ?

log_2(M) = C/2B
M = 2^(C/2B)
M = 2^(168/42) = 16
```

```
168 * 10^6 = 2(21*10^6 Hz)* log_2(M)
168 =  2(21)* log_2(M)
4 = log_2(M)
16 = M = 16-QAM
```

1.2

```
-70dBm - -90dBm = 10log(S/N)
20dBm = 10log(S/N)
10 = log(S/N)
S/N = 100
```

Now use Shannon's Law (to get maximum bitrate):

```
C = B log_2(1 + S/N) bits/s
C = ?
B = 21MHz = 21 * 10^6 Hz
S/N = 100 (not the dB measure)
C = 21*10^6 * log_2(101) = 139.82 Mbps
```

We need to get the maximum rate <= 139.83 Mbps Use $2(21 * 2^6 Hz) * \log_2(M)$ (see part 1)

```
BPSK:    2(21*10^6)* log_2(2) = 42 Mbps
QPSK:    2(21*2^6)* log_2(4) = 84 Mbps
8PSK:    2(21*2^6)* log_2(8) = 126 Mbps
16-QAM:  2(21*2^6)* log_2(16) = 168 Mbps
256-QAM: 2(21*2^6)* log_2(256) = 336 Mbps
```

8PSK is the highest rate below the maximum so that is the correct answer.

1.3

```
C = B log2(1+S'/N)
log2(1+S'/N) = C/B
1+S'/N = 2^(C/B)
S'/N = 2^(C/B) - 1
       = 2^(168/21) - 1
       = 2^(8) - 1
       = 255
Original S/N was 100, so we need to multiply the original power by 2.55 at least.
```

2 Problem 2

2.1

```
P(not make it) = 1/10*1/10*1/10 = 1 / 1000
P(make it) = 1 - P(not make it) = 1 - 1 / 1000 = 999/1000 = 99.9\%
```

2.2

```
E(time) = E(1st try)*P(1st try) + E(2nd)*P(2nd) + E(3rd)*P(3rd)
    0.5RTT*9/10 + 1.5RTT*(1/10*9/10) + 2.5RTT*(1/10*1/10*9/10)
 = 607.5/1000 RTT = 0.6075 RTT
```

2.3

Example in favor of retransmission:

- You want to do less work in an upper layer

Examples opposed to retransmission:

- If you care about delay then the added delay will be bad.
- More overhead in sending

3 Problem 3

3.1

The key for this is to realize what the learning switch is doing: when it sees a frame from a node, it remembers which port this came from. Further messages to that node will be sent only to that port, instead of being broadcast. If the switch has no knowledge of a node, it will broadcast a message. Periodically the switch forgets mappings, which allows nodes to move.

1. No correctness issues. Performance is bad, because every packet has to be broadcast.
2. No correctness issues. Performance is better, but only messages to 2 nodes are not broadcast, all others are still broadcast.
3. No correctness issues. Performance is best, after a while no (unicast) messages to nodes that transmitted at least once will be broadcast.
4. Performance will be good. However, there can be correctness issues: if a node moves from one port to another, the old mapping will be in effect, and messages might not be delivered.

3.2

The attack consists of the attacker sending frames with different source MAC addresses, so that the switch evicts the mapping of the victim V, and is forced to broadcast frames that are destined V.

V is sending 10 frames/s, which means that the switch renews this entry in its table 10 times/s. The attacker has to force the eviction of this between each message. There are 32 entries, which means the attacker has to send 32 messages with different source MAC addresses 10 times/s, or 320 frames/s.

4 Problem 4

4.1

This means each packet on the Ethernet network can get to every destination on the network. Broadcast domain ==> all devices in the network can broadcast to one another.

4.2

Exponential backoff ==> more fairness and less waste. Mainly: You don't waste cycles if the assigned node is idling.

The main difference is that in the randomized scheme you can accommodate any* number of nodes transmitting, without having to know in advance which or how many nodes will participate. This is good when you don't know, or when the number of nodes communicating, and the frequency with which they communicate, will change. The fixed scheme is good when you do know who and how much, a priori, and you can come up with an optimal scheme.

4.3

False! MAC learning doesn't completely eliminate broadcasts: any message to a node you have not heard from will still be broadcast. If the network has loops, and you didn't run STP, then you will have broadcast packets going around forever, in what we call a broadcast storm.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS168 document by filling out the anonymous feedback form:

`https://piazza.com/brown/fall2019/csci1680`.