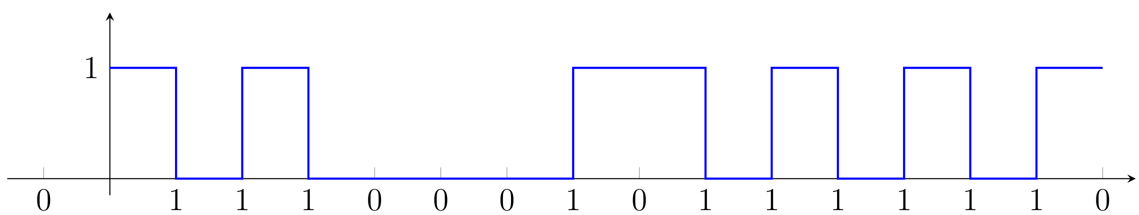
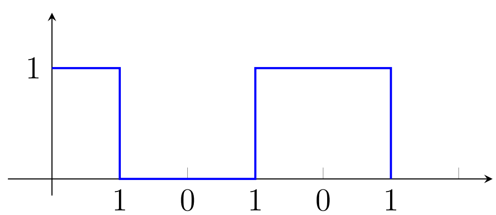
CSCI 4273 Fall 2014	Homework #3	**Vince Coghlan**

1: In the 4B/5B encoding (see Table 2.4), only two of the 5-bit codes used end in two 0s. How many possible 5-bit sequences are there (used by the existing code or not) that meet the stronger restriction of having at most one leading and at most one trailing 0? Could all 4-bit sequences be mapped to such 5-bit sequences?
You have all the ones that start and end with 1, or $3^2 = 9$. You have numbers that begin with 01 and end in 1, or $2^2 = 4$. Numbers that begin with 1 and end in 10, or $2^2 = 4$. and numbers that start with 01 and end with 10, which is two. That is 19, more than enough to handle 4 bits. Any other number will have at least 2 leading and/or following zeros.

2: Show the 4B/5B encoding, and the resulting NRZI signal, for the following bit sequence:
1110 0101 0000 0011
11100 01011 11110 10101

NRZI encoding





3: Assuming a framing protocol that uses bit stuffing, show the bit sequence transmitted over the link when the frame contains the following bit sequence: 110101111101011111101011111110
Mark the stuffed bits.
Stuffed bits will have an asterick before them:
1101011111*001011111*0101011111*0110

4: Suppose that one byte in a buffer covered by the Internet checksum algorithm needs to be decremented (e.g., a header hop count field). Give an algorithm to compute the revised checksum without rescanning the entire buffer. Your algorithm should consider whether the byte in question is low order or high order

take the old checksum one's compliment it, subtract 1 (pad it with 1, 2, or 3 bytes of zeros depending on where the byte occurs in its word.). The one's compliment will be the new checksum. For example if the checksum is 0xabcd, and the byte in question occurs second in the word 0x1234, then our new checksum would be:

$$0x5432\text{-}0x0100 \Rightarrow 0xACCD$$

5: Suppose we want to transmit the message 11100011 and protect it from errors using the CRC polynomial $x^3 + 1$.

(a) Use polynomial long division to determine the message that should be transmitted

```
       doesnt matter
1001 )11100011000
      1001
       1110
       1001
        1110
        1001
         1111
         1001
          1101
          1001
           1000
           1001
              100
```

This means we will send 11100011100

(b) Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What is the result of the receivers CRC calculation? How does the receiver know that an error has occurred?
The reciever will get 01100011100. When he or she calculates the CRC he will get 110, which is not 100, so he or she knows that an error has occured.

6: Suppose you are designing a sliding window protocol for a 1-Mbps point-to-point link to the moon, which has a one-way latency of 1.25 seconds. Assuming that each frame carries 1 KB of data, what is the minimum number of bits you need for the sequence number?

propegation delay is 1.25 seconds, that means in 1.25 seconds, there will be 1250000 bits. That is 1250 frames. This is our window size, we need 11 bits to represent this.

7: Suppose that we attempt to run the sliding window algorithm with SWS = RWS = 3 and with MaxSeqNum = 5. The Nth packet DATA[N] thus actually contains N mod 5 in its sequence number field. Give an example in which the algorithm becomes confused, that is, a scenario in which the receiver expects DATA[5] and accepts DATA[0]-which has the same transmitted sequence number in its stead. No packets may arrive out of order. Note this implies MaxSeqNum $\geq$ 6 is necessary as well as sufficient.

Consider the case where all frames are sent, and all of the ACKS are lost, the frames get resent and the reciever expexts 0 through 5, but the sender sends 0 through 4, times out, then sends 0 through 5. The reciever will have recieved 0,1,2,3,4,0 and assume that he has 0,1,2,3,4,5.

8: Suppose the round-trip propagation delay for Ethernet is $46.4\mu s$. This yields a minimum packet size of 512 bits (464 bits corresponding to propagation delay + 48 bits of jam signal).

(a) What happens to the minimum packet size if the delay time is held constant, and the signaling rate rises to 100 Mbps?
If the rate rises to 100Mbps, our propegation delay will allow us to transmit 100Mbps $\cdot 46.4\mu s$ =4640 bits. Our packet size now becomes 48+4640=4688 bits.

(b) What are the drawbacks to so large a minimum packet size?
It is a waste of all the nice bandwidth we have, if we dont need that much data.

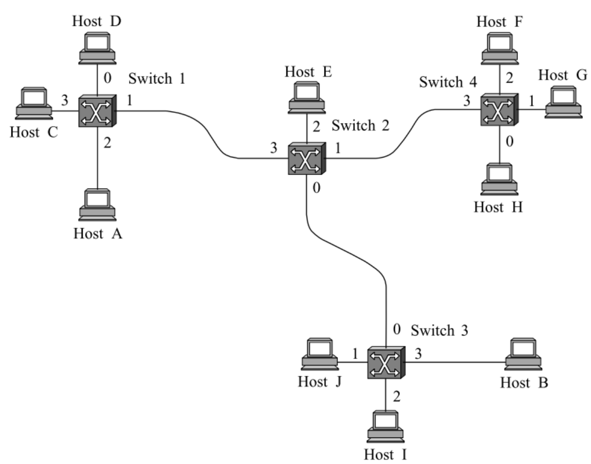
(c) If compatibility were not an issue, how might the specifications be written so as to permit a smaller minimum packet size?
shorten the wire up and loosen the strictness for the collision detection. You could lower the upper limit of the amount of hosts allowed on the network.

9: How can a wireless node interfere with the communications of another node when the two nodes are separated by a distance greater than the transmission range of either node?
If a third node sits in between the other two, in the range of both of thier communications, they will see interference from both nodes. Suppose our new third node wants to send a message to the second node wile the first node is trying to communicate with the third

node. Since our two nodes are not aware of eachother, there will be interference in these two signals.

10: How can hidden terminals be detected in 802.11 networks?
There are a few methods of doing this, one is for each node to send a detection request packet to each node, this will notify the original node of the other signals at each one-hop neighbor. They can then generate a list of all hidden nodes and respond accordingly. The other method is to listen for a data signal from a nearby node, then wait for the ACK. Since all nodes must ACK back, they can suspect that a hidden terminal must exist between that node and some other node. This requres much more guesswork but can easily be implemented if you know the constraints of the system.

11: Using the example network given in Figure 3.30, give the virtual circuit tables for all the switches after each of the following connections is established. Assume that the sequence of connections is cumulative, that is, the first connection is still up when the second connection is established, and so on. Also assume that the VCI assignment always picks the lowest unused VCI on each link, starting with 0.



(a) Host A connects to host B.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2

Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3

Switch 4:

[empty]

(b) Host C connects to host G.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1
3	4	1	1

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2
3	1	1	5

Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3

Switch 4:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	6	1	7

(c) Host E connects to host I.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1
3	4	1	1

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2
3	1	1	5
2	8	0	2

Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3
0	2	2	9

Switch 4:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	6	1	7

(d) Host D connects to host B.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1
3	4	1	1
0	10	1	1

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2
3	1	1	5
2	8	0	2

Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3
0	2	2	9

Switch 4:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	6	1	7

(e) Host F connects to host J.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1
3	4	1	1
0	10	1	1

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2
3	1	1	5
2	8	0	2
1	5	0	2

Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3
0	2	2	9
0	2	1	12

Switch 4:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	6	1	7
2	11	3	6

(f) Host H connects to host A.

Switch 1:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	0	1	1
3	4	1	1
0	10	1	1
1	1	2	0

Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	1	0	2
3	1	1	5
2	8	0	2
1	5	0	2
1	5	3	1

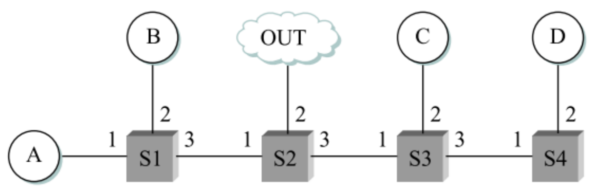
Switch 3:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	2	3	3
0	2	2	9
0	2	1	12

Switch 4:

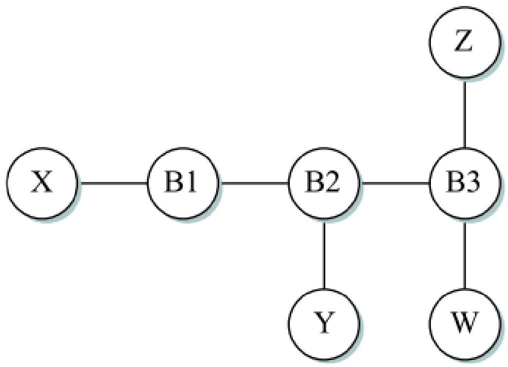
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	6	1	7
2	11	3	6
0	13	3	6

12: Give forwarding tables for switches S1-S4 in Figure 3.32. Each switch should have a default routing entry, chosen to forward packets with unrecognized destination addresses toward OUT. Any specific-destination table entries duplicated by the default entry should then be eliminated.



Switch	Host	Port
S1	A	1
	B	2
	default	3
S2	A	1
	B	1
	C	3
	D	3
	default	2
S3	C	2
	D	3
	default	1
S4	D	2
	default	1

13: Consider hosts X, Y, Z, W and learning bridges B1, B2, B3, with initially empty forwarding tables, as in Figure 3.36.



(a) Suppose X sends to Z. Which bridges learn where X is? Does Ys network interface see this packet?

Both B1, B2, and B3 will learn where X is. Y will see the packet as B2 is going to broadcast it to all ports.

(b) Suppose Z now sends to X. Which bridges learn where Z is? Does Ys network interface see this packet?

All bridges will know where Z is, Y wont see the packet this time Since B2 knows where X is.

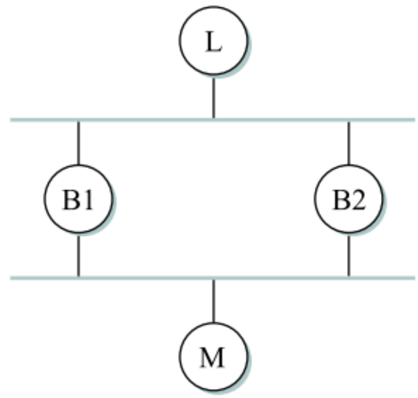
(c) Suppose Y now sends to X. Which bridges learn where Y is? Does Zs network interface see this packet?

B2 and B1 will learn where Y is, Z will not see this packet. since B2 will forward it on only one interface

(d) Finally, suppose Z sends to Y. Which bridges learn where Z is? Does Ws network interface see this packet?

All of the bridges already know where Z is, so none will learn. W sees the packet since B3 didnt learn where Y was.

14: Suppose two learning bridges B1 and B2 form a loop as shown in Figure 3.38, and do *not* implement the spanning tree algorithm. Each bridge maintains a single table of $<$*address*, *interface*$>$ pairs.



(a) What will happen if M sends to L?

Both bridges will contain an entry in thier table of the address and interface and L will recieve two packets. B1 will now send the packet to B2, which will forward it back to M. B1 will forward the packet from B2 to B1 in the opposite direction. There will be packets going in both directions from the same source.

(b) Suppose a short while later L replies to M. Give a sequence of events that leads to one packet from M and one packet from L circling the loop in opposite directions.

After the second message, B1 and B2 will contain entries for each node at each interface. This means that any packet sent to one will come out the other side regardless of address. if B1 deletes the table entry that says that M is located above it and B2 deletes the entry saying that L is above it, the bridges will endlessly forward packets in opposite directions.