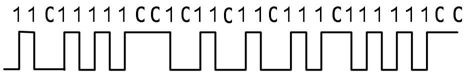
CS 640: Introduction to Computer Networks
Homework 2
Handed out: 10/05/2007
Due: 10/16/2007, in class

SOLUTION

1. Encoding [1]

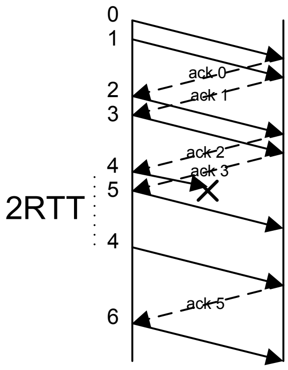


2. Sliding window

2.1) [1]

RTT = 1.25 * 2 = 2.5 s
Total packet out = 2.5 * 1M bps / 1KB + 1 = 305
Need 9 bits

2.2) [1]



3. Ethernet

3.1) [1] For a collision to be effectively detected, in this case, the sending packet should be able to last for at least

$1000 / 10^8 * 2 = 2*10^{-5}$ s

Thus, it should be at least

$100 * 10^6 * 2*10^{-5}$ = 1000 bits = 250 byte => Need to pad 250 – 40 = 210 bytes

3.2) [1] Node A senses empty channel from t=0 to t = 250, so A can transmit at any time during this interval. At time t=250, A senses a busy channel and will refrain from transmitting. So x=249 bit times and is equivalent to 24.9 microseconds

3.3 [1] Both nodes detect a collision at t=250. At t=250+48, both nodes stop transmitting their jam signals. The last bit of jam signal from B arrives at A at t=298+250=548 bit times. Similarly, the last bit of A's jam signal arrives at B at 548 bit times. This is when both nodes sense an idle channel. At 100Mbps, 548 bit times is equivalent to 5.48 microseconds.

4. Ethereal

4.1) [0.5]
TCP – 0x06
UDP- 0x11
ICMP – 0x01

4.2) [0.5] For all packet in the same "flow" (i.e. between the same source and destination IP's and ports), the Identifier and checksum fields will change.

4.3) [0.5] It is monotonically increasing

4.4) [0.5] There are 5 packets in this flow. Say each packet has a header of length B (note: this includes UDP+IP+Ethernet), and a 1000B payload. Therefore, we get:

5*B bytes of non-data (headers in the five packets)
5*(1000+B) total bytes transferred

Overhead = B/(1000+B)

UDP header length = 8 bytes
IP header length = 20 bytes
Ethernet header length = 14 bytes (not counting the preamble and the trailing CRC).
So B = 42 bytes and overhead due to *headers* alone = 4.03%.
With the Ethernet preamble (8 bytes) and the CRC (4 bytes), B = 54 bytes and the *total* overhead = 5.1%.

4.5) [0.5] ARP packets do not contain any data as the information about the MAC address is conveyed in the header itself.

4.6) [0.5] 00:02:2d:44:e5:07
00:07:e9:53:87:d9

4.7) [0.5] 0x0800

5. Traceroute revisited

5.1) [1] Not listed

5.2) [0.5] mren.org (this is the Internet2 educational network – all traffic between educational institutions goes over this network)
Wiscnet.net (this is our "commercial" provider – all traffic to commercial websites uses this provider)

5.3) [0.5] From traceroutes to the destinations mentioned in the problem, these are the only two providers that emerge. UW-Madison also has a direct connection with TDS. Actually, this is a peering relationship with. For example, try tracerouting to tds.net.

6. Distance vector

6.1 [0.5]

	A	B	C	D	E	F
A	0	inf	3	8	inf	inf
B	Inf	0	inf	inf	2	inf
C	3	inf	0	inf	1	6
D	8	Inf	inf	0	2	inf
E	Inf	2	1	2	0	inf
F	inf	inf	6	inf	inf	0

6.2[0.5]

	A	B	C	D	E	F
A	0	inf	3	8	4	9
B	Inf	0	3	4	2	inf
C	3	3	0	3	1	6
D	8	4	3	0	2	inf
E	4	2	1	2	0	7
F	9	inf	6	inf	7	0

6.3[0.5]

	A	B	C	D	E	F
A	0	6	3	6	4	9
B	6	0	3	4	2	9
C	3	3	0	3	1	6
D	6	4	3	0	2	9
E	4	2	1	2	0	7
F	9	9	6	9	7	0

7. Link-State [1.5]

Done	**Open**	**Unseen**
D(0,-)	**A(8), E(2)**	**B, C, F**
D(0,-), E(2, E)	**A(8), C(3), B(4)**	**F**
D(0,-), E(2,E), C(3,E)	**A(6), B(4), F(9)**	
D(0,-), E(2,E), C(3,E), B(4,E)	**A(6), F(9)**	
D(0,-), E(2,E), C(3,E), B(4,E), A(6, C)		
D(0,-), E(2,E), C(3,E), B(4,E), A(6, E), F(9, E)		

8. Routing

You need to do a bitwise AND between the IP address and subnet mask to get the subnet number, and correspondingly, the interface number.

8.1) [0.5] Interface 0
8.2) [0.5] R2
8.3) [0.5] R4
8.4) [0.5] R3
8.5) [0.5] R4