CS 349/449
Internet Protocols
Midterm Solutions
Winter 2004

10/18/2004

Question	349 Points	449 Points	Score
1	10	10	
2	20	10	
3	20	20	
4	20	20	
5	30	20	
6 (449 only)	-	20	
Total:	100	100	

Question 1 Answer the following using True/False. You do not need to explain your answers

1. The UDP transport protocol provides reliable transfer over unreliable channels. **FALSE**
2. IP fragments are re-assembled at the destination. **TRUE**
3. The IPv6 address space is 1 billion times larger than the IPv4 space. **FALSE**
4. The End-to-End argument argues for putting extra functionality in the network **FALSE**
5. A bridge works at the Data Link layer of the protocol stack **TRUE**
6. No collisions can occur when the CSMA/CD is used. **FALSE**
7. Virtual Circuit Switching requires a circuit establishment phase before any data can be sent. **TRUE**
8. The Internet checksum detects all two-bit errors. **FALSE**
9. An Ethernet bridge requires setting up its forwarding table before it can forward any packets. **FALSE**
10. In RIP each node broadcasts its routing table to all nodes in the network. **FALSE**

Question 2: Assume you wish to transfer an *n*-byte file along a path composed of the source, destination, seven point-to-point links, and five switches. Suppose each link has a propagation delay of 2ms, bandwidth of 4Mbps, and that the switches support both circuit and packet switching. Thus you can either break the file up into 1-KB packets, or set up a circuit through the switches and send the file as one contiguous bit stream. Suppose that packets have 24 bytes of packet header information and 1000 bytes of payload, that store-and-forward packet processing at each switch incurs a 1-ms delay after the packet has been completely received, that packets may be sent continuously without waiting for acknowledgements, and that circuit setup requires a 1-KB message to make one round-trip on the path incurring a 1-ms delay at each switch after the message has been completely received. Assume switches introduce no delay to data traversing a circuit. You may also assume that file size is a multiple of 1000 bytes.

(a) For what file size *n* bytes is the total number of bytes sent across the network less for circuits than for packets?

Answer: (10 points) The number of bytes sent in the VC case is

$B_c = 2(p+h)+n$

Note that the first packet has to complete a round trip before the circuit can be established. The number of bytes sent in the packet switching case is:

$B_p = n/1000*(p+h)$

Where p is the number of data in the packet and h is the size of the packet header. We want:

$B_c < B_p$

Solving for n we find:

$n > (p+h)*2000/(p+h-1000)$ (1)

Replacing p with 1000 and h with 24 we get:

$n > 85,333,33$

Therefore n = 86,000.

(b) For what file size *n* bytes is the total latency incurred before the entire file arrives at the destination less for circuits than for packets?

The time required to complete the data transfer in the packet switching case is:

$T_p = T_f + (c-1)*T_x$

Where T_f is the time needed for the first packet to completely arrive at the destination, T_x is the packet transmission time (p+h/b, b is the link bw) and c is the number of packets (n/1000). The intuition is that after the first packet is delivered at the destination one packet arrives for every transmission time. T_f can be expressed as:

$T_f = (s+1)* T_x + (s+1)* T_g + s* T_s$

Where T_g is the propagation delay, s is the number of switches, and T_s is the switch processing time. The time T_c for the VC case is:

$T_c = 2* T_f + 6* T_g + c* T'_x$

Where T'_x is the packet transmission time for the VC case (p/b). The reason is that in the VC case, $2* T_f$ seconds are required to establish the circuit, then $6*T_g$ are required for the first data byte to arrive at the destination and finally for $c*T'_x$ all the data to arrive. We want:

$T_p > T_c$ (2)

By solving (2) for n, we find that n = 903,000

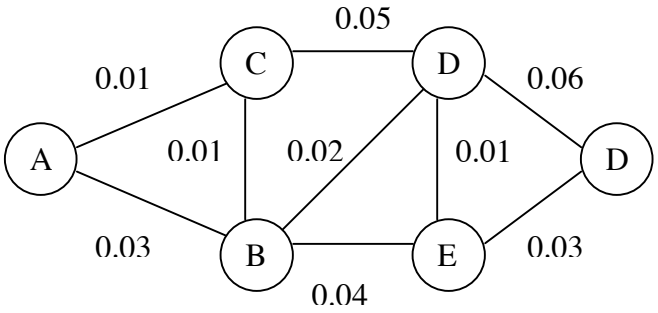
(c) For what file size *n* bytes is the total latency incurred before the entire file arrives at the destination less for circuits than for packets?

Answer: (5 points) From (1) we can see that in (a) only the payload to header size ratio affects the number of bits sent. On the other hand, number of switches and link bandwidth are irrelevant

(5 points) In similar way we can see from (2) that for (b) the link bandwidth, number of switches, and payload to header size ratio are all important. When the number of switches increases n increases. When the link bandwidth increases n also increases. Finally, when the size p decreases n decreases.

Note: For 449 the final grade is computed by (a+b+c)*2/3

Question 3: The number shown next to each link of the network shown below is the probability of the link failing. It is assumed that links fail independently of each other. Give the forwarding table for Node A where for each destination the path used is the one which the probability that all the links will stay intact is maximal. Each entry in the forwarding table should contain the next path to the destination and the "cost" of the path to the destination.

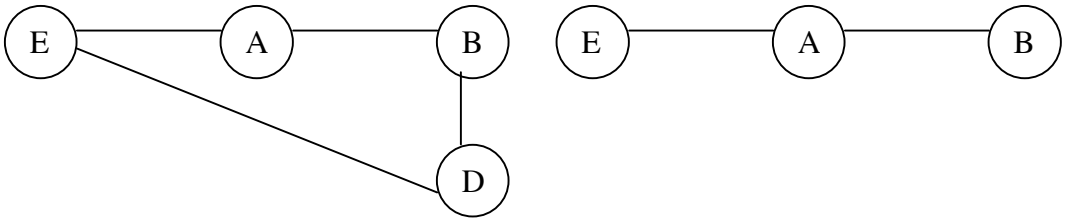


Solution: This is just the normal shortest path problem but in this case the cost of a path A,B,C is equal to $(1-P_{ab})*(1-P_{bc})$ where P_{ab} is the link failure probability for link AB.

The routing table for node A will then be:

Destination	**Next Hop**	**Cost**
B	C	0.9801
C	C	0.99
D	C	0.9605
E	C	0.9509
F	C	0.922

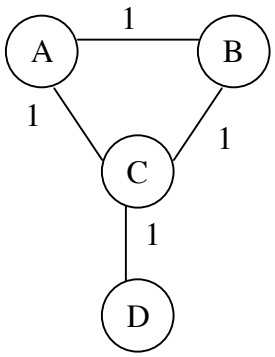
Question 5: *Hold-down* is another distance-vector loop-avoidance technique, whereby hosts ignore updates for a period of time until link failure news has had a chance to propagate. Consider the networks below, where all links have cost 1, except E-D with cost 10. Suppose that the E-A link breaks and B reports its loop-forming E route to immediately afterward (that is the false route, via A, no poison reverse is used). Specify the details of a hold-down interpretation, and use this to describe the evolution of the routing loop in both networks. To what extent can hold down prevent the loop in the EAB network without delaying the discovery of the alternative route in the EADB network?



We will implement hold-down as follows: when an update record arrives that indicates a destination is unreachable, all subsequent updates within some given time interval are ignored and discarded. Given this, then in the EAB network A ignores B's reachability news for one time interval, during which time A presumably reaches B with the correct unreachability information.

Unfortunately, in the EABD case, this also means A ignores the valid B–D–E path. Suppose, in fact, that A reports its failure to B, D reports its valid path to B, and then B reports to A, all in rapid succession. This new route will be ignored. One way to avoid delaying discovery of the B–D–E path is to keep the hold-down time interval as short as possible, relying on triggered updates to spread the unreachability news quickly. Another approach to minimizing delay for new valid paths is to retain route information received during the hold-down period, but not to use it. At the expiration of the hold-down period, the sources of such information might be interrogated to determine whether it remains valid. Otherwise we might have to wait not only the hold-down interval but also wait until the next regular update in order to receive the new route news.

Question 5: (CS449 Only)
Given an example of a situation where the poison reverse mechanism does not prevent the count to infinity problem in RIP.



In this network if link CD fails the following sequence of events can lead to count-to-infinity.

1. Link CD fails
2. C sends update (D, inf) to A and B (M1, M2)
3. A receives update (M1) from C and selects node B as next hop to D with distance of three
4. A sends update to C (D, 3) (M3) and to B (D, inf) (M4) (poison reverse)
5. B receives update (M2) from C (D, inf) and selects A as next hop to D with distance 3
6. B sends update to C (D, 3) (M5) and to A (D, inf) (M6) (poison reverse)
7. C receives A's advertisement (M3) and selects A as the next hop to D with distance of 4
8. C sends update to B (D, 4) (M7)
9. B receives A's advertisement (D,inf) (M4) and at this point B has no route to D
10. B sends an advertisement (D, inf) to C (M8)
11. A receives B's message (M6) and at this point A has no route to D
12. A sends (D, inf) to C (M9)
13. B receives (D, 4) (M7) from C and selects C as next hop to D with distance 5
14. B sends update (D, 5) to A (M10) and (D, inf) to C (M11)
15. C receives M9, M11 and has no route to D
16. A receives M10 from B and sets B as it's next hop to D with distance of 6

At this point you can see that the nodes will keep accepting routes with increasing distance to D until the metric reaches infinity