CMPT 371: Midterm 2

August 1, 2006

1 Short Answer Questions

1. How does the main service provided by the network layer differ from the main service provided by the transport layer?

   Solution: The transport layer provides communication between processes, while the network layer provides communication between hosts.

2. True or False:

   (a) Queuing at a router occurs only before the packet goes through the switching fabric.

   (b) A single transport layer segment can be divided into multiple IP datagrams.

   (c) Every virtual circuit (VC) connection has a single numerical identifier.

   Solution: FTF

3. What does AS stand for and what is it? Of RIP, BGP and OSPF, which are intra-AS routing protocols and which are inter-AS routing protocols?

   Solution: AS stands for "Autonomous System." An AS is a network under the control of a single organization where certain network administration tasks can and must be performed locally. RIP and OSPF are intra-AS routing protocols. BGP is an inter-AS routing protocol (even though it has an iBGP and an eBGP part).

4. Briefly describe what Random Early Detection (RED) is.

   Solution: A router can use RED to notify any senders that are using it that it is becoming overwhelmed. It does this "early," that is, before a lot of packets are lost due to buffer overflow. Specifically, a router keeps track of its recent average queue length. If this average queue length exceeds a certain threshold, the router will drop packets randomly. If this threshold exceeds another, higher threshold, the router will drop every packet. Any senders that are using TCP, for instance, will respond by adjusting their `CongWin` parameters to slow down their sending rate.

5. Explain why a virtual circuit (VC) can give a bandwidth guarantee to the traffic between two hosts and why a datagram network (under normal circumstances) cannot.

Solution: In a VC, there is a "connection" between every pair of communicating hosts. It is important to explain how this can lead to a bandwidth guarantee. Two of the essential reasons are (i) there is a fixed path between the two hosts, so after the connection is established, we will know the minimum transmission rate of any link in the path; and (ii) every router knows how many connections it is involved in and can decide to reject new connection requests if it already has too much traffic going through it. In contrast, packets travelling between two hosts in a datagram network can take many different paths and routers have to handle whatever traffic is directed towards them.

2 Longer Questions

1. Consider a router with 3 outgoing links (0,1,2) and the following forwarding table given by ranges of 4-bit addresses:

Address range	Outgoing link
0000–0011	0
0100–0111	1
1000–1011	2
1100–1111	0

(a) (2 points) Represent this table in longest-prefix-matching form such that the longest prefix has length 2 and there are 3 entries.

Solution:

Prefix	Outgoing link
01	1
10	2
ϵ	0

(b) (1 point) Can it be done with longest prefix 1?

Solution: No.

(c) (4 points) Now consider a router with k outgoing links and assume the addresses have m bits where $m > k$. At this router, each link has at least one destination address that gets forwarded onto it. Give a lower bound on the length of the longest prefix in the forwarding table for this router. Explain why.

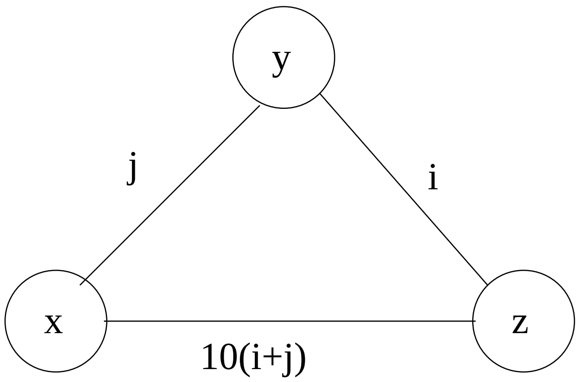
Solution: We need to construct a set of k binary strings (not necessarily all of the same length) such that every string in the set can be extended to an m-bit string that is not an extension of any of the other strings in the set. Certainly this can be done if we make every string of length $\ell = \lceil \log k \rceil$. In fact, ℓ is the smallest integer such that $2^\ell \geq k$, so if all the strings are of the same length, then that length must be at least ℓ. What if we use strings of different lengths? Can we get away with strings of length at most $\ell - 1$? This turns out to be equivalent to the question of how many leaves there can be in a binary tree of depth $\ell - 1$ (why?). The answer is $2^{\ell-1} < k$, so we do need at least one string of length ℓ. Full credit for this question required at least acknowledging the possibility that the prefixes need not be of the same length.

(d) (4 points) Show one possible forwarding table for this router where the longest prefix is as close as possible to the lower bound in part (c).

Solution: "This router" refers to the router in part (c); that is, a router with k outgoing links, not 3 outgoing links. In most cases, 1 point was taken off for assuming k was a specific number. Again, let $\ell = \lceil \log k \rceil$. For any number $0 \leq x < k$, let $\langle x \rangle_\ell$ denote the binary representation of that number with 0's added to the left to make it a string of length ℓ. Then we can use the following table:

Prefix	Outgoing link
$\langle 0 \rangle_\ell$	0
$\langle 1 \rangle_\ell$	1
.	
.	
.	
$\langle k-1 \rangle_\ell$	k-1

2. This question is about the Distance Vector (DV) algorithm. Consider the following network, where i and j are constants such that $j > i > 1$.



Recall that each node stores its own distance vector and the distance vectors of its neighbors. Assume that the initial phase of the algorithm is over and each node has three correct distance vectors. That is, each node has the information

	x	y	z
D_x	0	j	$i+j$
D_y	j	0	i
D_z	$i+j$	i	0

(a) (4 points) Suppose the cost of link (x, y), $c(x, y)$, changes to 1. Describe what the algorithm does: that is, what messages are passed and what vector updates are made. How many messages does y send before every node's information is correct?

Solution: x and y detect the change in $c(x, y)$, so y updates D_y using the new $c(x, y)$ and its old copy of D_x and D_z. Likewise, x updates its copy of D_x using $c(x, y)$ and its old copies of D_y and D_z. y sends D_y to x and z, and x sends D_x to y and z. When y receives D_x from x, nothing changes in D_y. Likewise, when x receives D_y from y, nothing changes in D_x. After z receives D_y, it updates D_z (nothing changes when z

receives D_x). Finally, z sends D_z to x and y, but nothing changes. All in all, y sends two messages.

(b) (6 points) Starting from the initial costs, assume that the cost of link (x, y) changes to $20(i + j)$. How many messages does y pass before every node's information is correct?

Solution: To get more than 1 point, you had to recognize that this was an instance of "count to infinity" or "bad news travels slowly," and that the number of messages sent by y would depend on i and j. When y registers the change in $c(x, y)$, it updates $D_y(x)$ to $2i + j$ (the path that goes from y to z and then takes the shortest path from z to x). It then sends D_y to x and z. z updates $D_z(x)$ to $3i + j$ (going through y and then taking the shortest path from y to x) and sends D_z to x and y. This continues until $D_y(x)$ gets to $10(i + j) + i$. $D_y(x)$ increases by $2i$ for every two messages sent by y (one to z, one to x). Therefore, the total number of messages sent by y is $2\ \frac{10(i+j)+i-j}{2i}$. You didn't have to get this exact expression to get full credit; just something close.