CSE123 Spring 2015 Homework #3
Instructor: Stefan Savage

Due Friday 5/27 **at the beginning of class**
(Please TYPE your answers… this will allow our TAs to be much more efficient)

1. IP addressing

Consider the IPv4 address 128.2.145.6

a) Assuming we're using class-based addressing, explain i) what "class" the address is and ii) what is the address of the network part and iii) what is the address of the host part

i) this is a class B address
ii) the network part is 128.2.x.x (saying 128.2 or 128.2/16 is fine too)
iii) the host part is 145.6 (or any reasonable description thereof

b) Assume that we're using classless (i.e., CIDR) addressing and the longest matching advertised route cover this address is 128.2.128.0/17. As with part "a" above explain i) what is the network part of the address and ii) what is the host part

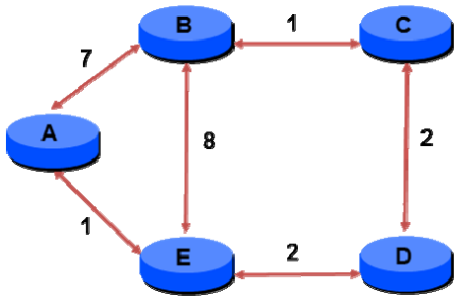
i) 128.2.128.0/17 (i.e. the first 17 bits of the address)
ii) 0 (the last 15 bits of the address)

c) Inside the organization that owns this network (CMU incidentally) they further subdivide the host part of the address (from part b above) into a subnet prefix and host address (let suppose the least significant 8 bits represent the host and the remainder identifies the subnet inside CMU). How many different LANs can be supported with this subnetting architecture?

If there are 15 bits in the host part and 8 are used for hosts within a subnet, then there are 7 bits for subnetting, or 128 distinct subnets (I'm also ok with the answer 127 if they want to argue that the all 0 subnet won't be allowed)

2. DV Routing

Consider the following network using distance vector routing:



Info at node	Distance to Node				
	A	B	C	D	E
A	0	6	5	3	1
B	6	0	1	3	5
C	5	1	0	2	4
D	3	3	2	0	2
E	1	5	4	2	0

Suppose an additional link is added connecting A and C with **cost 2**. List the distance vector updates that will be sent for the system to reconverge (the first update, from A, is listed below). Assume that updates occur simultaneously in rounds (i.e., at a set time each node transmits their distance vectors to their neighbors). You only need list DV updates in a route that have changed since the last round. [also: ignore split horizon or poison reverse in this example]

Round 1:
A: 0, 6, 2, 3, 1 (reflecting only the change in the AC link)
C: 2, 1, 0, 2, 4 (reflecting only the change in the AC link)

Round 2:
A: 0, 3, 2, 3, 1 (A's cost to B is updated by C's vector from round 1)
B: 3, 0, 1, 3, 5 (B's cost to A is updated by C's vector from round 1)
C: 2, 1, 0, 2, 3 (C's cost to E is updated by A's vector from round 1)

Round 3:
B: 3, 0, 1, 3, 4 (B's cost to E is updated by A's vector from round 2)
E: 1, 4, 3, 2, 0 (E's cost to B and C is updated by A's vector from round 2)
[done]

3. DHCP/ARP

Suppose a host A (Ethernet address: 3c:15:c2:cd:46:4c) , starts up and wants to send a packet to some remote Internet host with IP address 128.95.1.2. On the same LAN as host A is a DHCP server (IP address: 132.239.1.2, Ethernet address: 00:c0:ff:72:dc:2c) and the default router for the LAN (IP address: 132.239.1.1, Ethernet address: 00:c0:ff:65:23:49). Assume that the DHCP server will allocate 132.239.1.3 to the next host asking for an IP address and that all ARP caches are empty. Describe the sequence of DHCP and ARP requests/responses among these parties from when host A starts to the point where they are able to send a packet to 128.95.1.2. Label each request/response with the appropriate source and destination Ethernet and IP addresses.

DHCP Discover
MAC Source: 3c:15:c2:cd:46:4c
MAC Destination: ff:ff:ff:ff:ff:ff
IP Source: 0.0.0.0
IP Destination: 255.255.255.255

DHCP Offer
MAC Source: 00:c0:ff:72:dc:2c
MAC Destination: ff:ff:ff:ff:ff:ff
IP Source: 132.239.1.2
IP Destination: 255.255.255.255

DHCP Request
MAC Source: 3c:15:c2:cd:46:4c
MAC Destination: ff:ff:ff:ff:ff:ff
IP Source: 0.0.0.0
IP Destination: 255.255.255.255

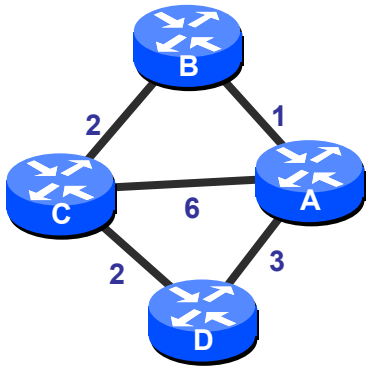
DHCP Ack
MAC Source: 00:c0:ff:72:dc:2c
MAC Destination: 3c:15:c2:cd:46:4c
IP Source: 132.239.1.2
IP Destination: 132.239.1.3

ARP Request
MAC Source: 00:cO:ff:72:dc:2c
MAC Destination: ff:ff:ff:ff:ff:ff
IP Source: 132.239.1.3
IP Destination: 132.239.1.1

ARP Reply
MAC Source: 00:c0:ff:65:23:49
MAC Destination: 00:c0:ff:72:dc:2c
IP Source: 132.239.1.1
IP Destination: 132.239.1.3

4. Link-state routing

A link-state routing protocol provides each node with a complete map of the topology. In typical usage each node then runs Dijkstra's algorithm to compute the shortest path from that node to every other node. The appropriate next hop for each destination is extracted from this result and installed into the router's packet forwarding table. However, this approach restricts packets bound for a given destination to use a *single path* and therefore the network may not be optimally utilized. For example, in the network below, A will always send to C through B, since that is the lowest cost path – even though the direct path (cost 6) could be used as well.



To help address this, most routers support a feature – *equal cost multipath* – that allows a router to split traffic between multiple next hop routers, so long as they are on paths of equal cost. For example, using this feature, router D could forward half of its traffic bound for B through A and the other half through C (both paths have cost 4). This works. Indeed, if the traffic is split in this way, D can be guaranteed that A and C will deliver it successfully to B.

However, this is not true in general for non-equal cost paths.

a) Give an example in this network where a router forwarding traffic between two non-equal costs paths will lead to a problem.

Consider what happens if B tries to split its load to C. Half of its traffic goes directly to C and the other half to A. For the traffic arriving at A, the shortest path to C is back through B. Consequently, this will create a routing loop for this traffic between A and B for traffic destined to C.

b) Explain why this works in general for equal cost paths, but not for non-equal cost paths?

Link-state protocols create an environment in which each router has complete knowledge of the entire topology. Since each router calculates the shortest path from itself to each destination, when it forwards a packet to the next hop on the shortest path it can assume that the next hop router's version of the shortest path is a "suffix" of its own shortest path and thus each "hop" will strictly decrease the cost to get to the destination. When we send traffic along multiple equal-cost paths, this property is still true and at each and every "hop" the shortest path to the destination decreases. Since the cost always decreases on each hop, there can be no loops. However, with non-equal cost paths a packet that takes the a non-shortest-path can actually cause the distance to the destination to increase (as seen in the example above)