Exercise Session 9: Network Layer

COM-208: Computer Networks

Before we start

A few basic rules:

- An IP prefix A/M is the range of IP addresses whose M most significant bits are the same as A's M most significant bits. E.g., 10.0.0.16 belongs to IP prefix 10.0.0.0/24 , because the 24 most significant bits of 10.0.0.16 are the same as the 24 most significant bits of 10.0.0.0.
- This implies that a /M IP prefix contains 2^{32-M} IP addresses. E.g., a /24 IP prefix contains $2^{32-24} = 2^8 = 256$ IP addresses. In other words, we don't have the freedom to create an IP prefixes that contains an arbitrary number of IP addresses, it must always contain a number that is a power of 2.
- /M is called the "subnet mask", representing the bitmask consisting of M consecutive 1's followed by enough 0's to reach 32 bits. The subnet mask is applied to any IP address using bitwise AND to obtain the IP range the address belongs to. E.g. 100.52.12.18 belongs to the prefix 100.52.12.16/28 because: (1) the /28 mask represents the mask **1111 1111.1111 1111.1111 1111.1111 0000**, (2) applying the mask to 100.52.12.18 (**0110 0100.0011 0100.0000 1100.0001 0010** in binary) results in 100.52.12.16 (**0110 0100.0011 0100.0000 1100.0001 0000** in binary).
- Each IP subnet must have its own IP prefix. Hence, IP prefixes allocated to different IP subnets must not overlap.

When assigning IP addresses to network interfaces in an IP subnet, assume that the following IP addresses cannot be assigned to any network interface:

- **The first IP address in the subnet's IP prefix (called "network address").** E.g., the first IP address in 10.0.0.0/24 is 10.0.0.0. This address is sometimes reserved for special uses, e.g., a discovery service provided by the subnet.
- **The last IP address in the subnet's prefix (called "broadcast address").** E.g., the last IP address in 10.0.0.0/24 is 10.0.0.255. This address is sometimes reserved to be used as the subnet's broadcast address, i.e., as the destination IP address for packets that should be received by all end-systems in a subnet.

IP prefix allocation

<u>*Basic*</u>

IP subnets A, B and C contain 10, 5, and 3 network interfaces, respectively. Allocate an IP prefix to each subnet, and assign an IP address to each network interface, from IP prefix 1.2.3.0/27.

Consider two cases for allocating prefixes to subnets. In each case, follow the given order:

(a) A, B, C

(b) B, A, C

In the context of this exercise, when we say that allocation "follows a given order," we mean that, if Subnet X comes before Subnet Y in that order, the IP addresses for Subnet X should be arithmetically smaller than the IP addresses for Subnet Y (in the sense that IP address 1.2.3.4 is arithmetically smaller than IP address 1.2.3.5).

Note: Allocating addresses might be infeasible in some cases.

Network configuration

Consider the topology shown in Figure 1. There are three IP subnets (A, B and C) that contain some end-systems, and two IP subnets (D and E) that contain no end-systems. The green boxes (a, b, c, . . . g) denote network interfaces for routers R1, R2 and R3.

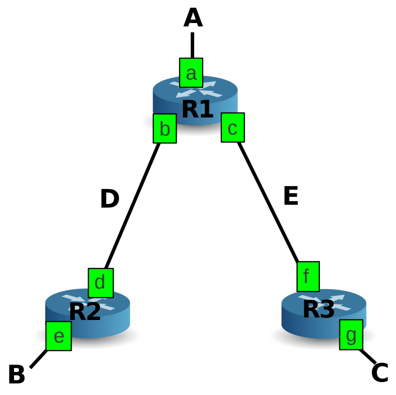

Figure 1: Problem topology.

- (*Basic*) Allocate an IP prefix to each subnets. Your allocation must respect the following constraints:
  - All prefixes must be allocated from 214.97.254/23.
  - Subnet A should have enough addresses to support 250 interfaces.
  - Subnet B should have enough addresses to support 120 interfaces.
  - Subnet C should have enough addresses to support 60 interfaces.
  - Each of subnets D and E should have enough addresses to support 2 interfaces.
- (*Basic*) Using your previous answer, provide the forwarding tables for each of the three routers (R1, R2, R3). Each table should contain two columns which show (i) the destination IP prefix, and (ii) the corresponding output link.
- (*Advanced*) Can you reduce the number of entries of each forwarding table, i.e., for each table create an equivalent one, which has the same outcome but consists of fewer entries?

Network Address Translation

Intermediate

Figure 2 illustrates how Network Address Translation (NAT) works. Consider a similar topology, but suppose that the NAT gateway has external IP address 24.34.112.235, while the private IP address space is 192.168.1.0/24.

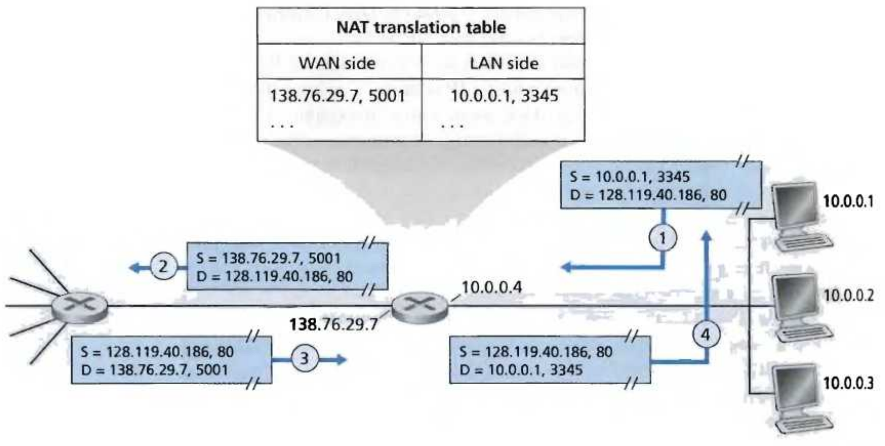

Figure 2: Network Address Translation (NAT) Process

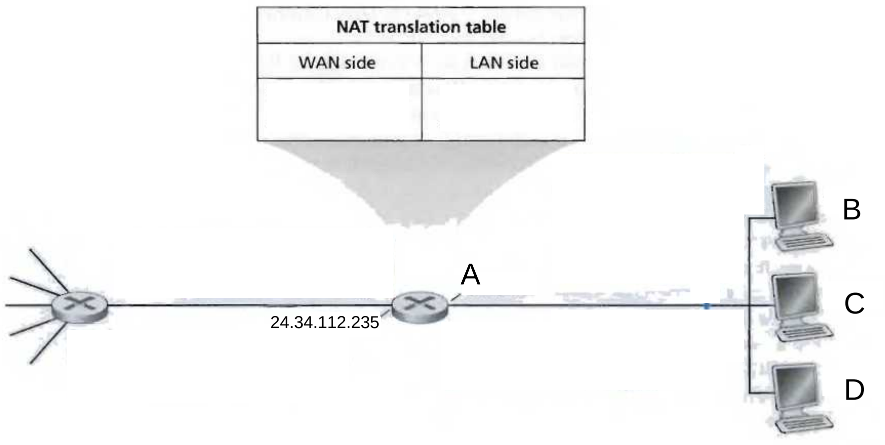

Figure 3: Problem topology.

- Complete Figure 3 by assigning IP addresses to all interfaces (labels A, B, C, and D) in the internal (private) network.
- Suppose that each end-system has two ongoing TCP connections, all to IP address 128.119.40.86, port 80. Provide the six corresponding entries in the NAT translation table.

Link-state routing

Basic

Consider the network in Figure 4. Execute the link-state (Dijkstra's) algorithm we saw in class to compute the least-cost path from each of x, v, and t to all the other routers.

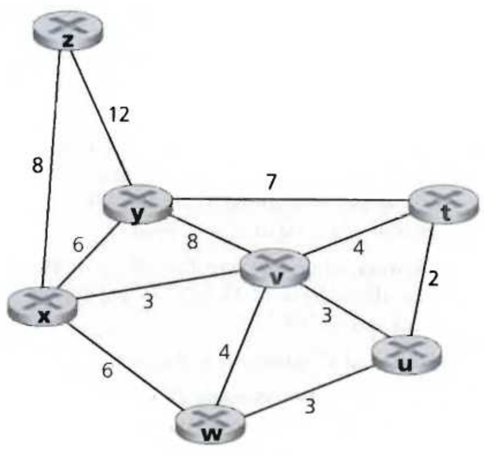

Figure 4: Network topology.

Distance-vector routing

Basic

Consider the network in Figure 5. Execute the distance-vector (Bellman-Ford) algorithm we saw in class and show the information that router z knows after each iteration.

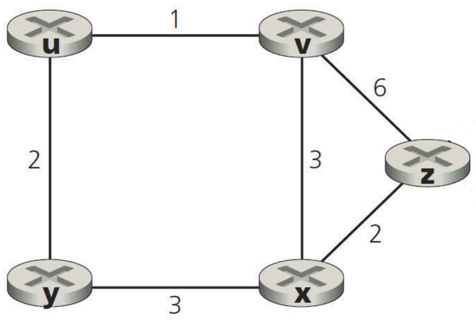

Figure 5: Network topology.

Convergence

Intermediate

What is the maximum number of iterations required for the distance-vector (Bellman-Ford) algorithm that we saw in class to converge (i.e., to finish, assuming no change occurs in the network graph and link costs)? Justify your answer.

Poisoned reverse

Advanced

Consider the network in Figure 6. The routers in the network run the distance-vector (Bellman-Ford) algorithm we saw in class, with poisoned reverse enabled. Suppose the algorithm has run for some time and has converged to the correct least-cost paths.

Table 7 contains the reachability information from each router to router A (e.g., from router D we can reach router A through router C with cost 3).

Now suppose that link $A - B$ goes down.

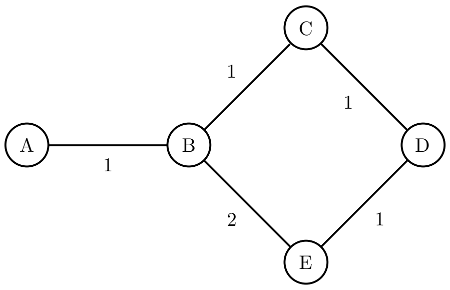

Figure 6: Network topology.

- Describe the next 6 steps of the algorithm. For each step, show the reachability information from each router to router A (cost of the least-cost path and next hop).

		from router B	from router C	from router D	from router E
route to router A	step 0	1 via A	2 via B	3 via C	3 via B
	step 1				
	step 2				
	step 3				
	step 4				
	step 5				
	step 6				

Figure 7: Reachability information from each router to router A.

- Will the algorithm converge to the correct least-cost path values? If yes, in how many steps?
- Propose a simple way to make the algorithm converge faster.