Communication Networks

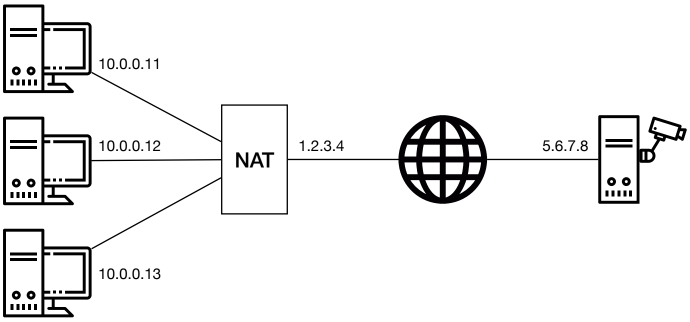
Prof. Laurent Vanbever

Solution: Exercise 7 – NAT & Routing

NAT

7.1 NAT (Exam Question 2018)

Consider the network topology below. Alice has multiple PCs at home (10.0.0.11–13) which share a single public IP address (1.2.3.4) via a NAT device. Further, she operates a surveillance camera server which is directly connected to the Internet with a public IP address (5.6.7.8). The camera transmits the live video signal as a stream of UDP packets with source port 1000 to a configurable destination IP address and port.



Alice operates three PCs and one camera server

a) Alice wants to receive the live video stream on one of her PCs and thus configures the camera to send the video signal to IP 10.0.0.11 and port 1234. However, she does not receive it on her PC. Why? Where is this traffic sent to?

Solution: All IP addresses in the 10.0.0.0/8 prefix are private and not routed in the Internet. As 10.0.0.11 is one of these internal IPs, the camera has no route to this address. Consequently, that traffic is dropped.

b) Now Alice configures the camera to send the video signal to IP 1.2.3.4 and port 1234. But she still does not receive it on any of her PCs. Why? Where is this traffic sent to?

Solution: The IP address 1.2.3.4 is a globally routed address and therefore, the traffic arrives at the NAT box. However, as there is no corresponding address translation rule in the NAT for that specific destination port, the NAT does not know how to rewrite the packet and where to forward the traffic to. The traffic is dropped at the NAT box.

c) What can Alice do such that she receives the video signal at her PC with IP address 10.0.0.11 and at port 1234 assuming that she *cannot* modify the configuration of the NAT? Describe step-by-step what she can do if she has the following possibilities:

- send one single UDP packet with arbitrary source and destination addresses and ports from each of her PCs;
- observe the received packets at each of her PCs and the camera server;
- specify the destination IP address and port for the video signal.

Solution: First, you want to “open a hole” in the NAT box for the video stream to enter your network. To do this, you send a packet from an internal host, for example 10.0.0.11:1234, to the camera 5.6.7.8:1000. This leads the NAT box to install an address translation rule.

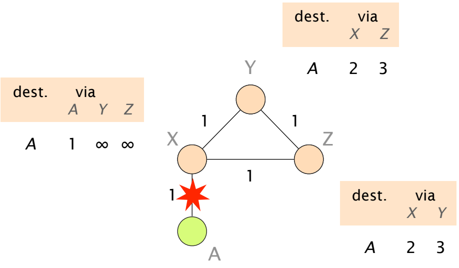
Now, you have to configure the camera with the correct destination IP address and port. The destination IP address is clear: it is the one of the NAT box (1.2.3.4). The port, however, you do not know yet.

Therefore, you start observing the packets arriving at the camera while sending packets from the interal host to the camera, from 10.0.0.11:1234 to 5.6.7.8:1000. At the camera, you will see to what port the NAT changed the source port of the packet.

Finally, configure the camera to send the video stream to the IP address of the NAT box and set the destination port to the port observed previously.

Routing

7.2 Convergence with Poisoned Reverse



dest.	via A	via Y	via Z
A	1	∞	∞

dest.	via X	via Z
A	2	3

dest.	via X	via Y
A	2	3

Consider the network on the left which uses distance vector routing with poisoned reverse. Each link is associated with a weight that represents the cost of using it to forward packets. Link weights are bi-directional.

Assume that the link between X and A fails (as shown in the figure) and use the table below to show the first 8 steps of the convergence process. How many steps does it take until the network has converged to a new forwarding state? Explain your observations.

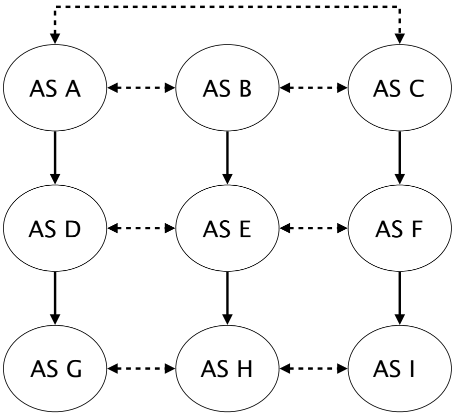
Solution: The network does not converge as the maximum link weight is increased by one in each round ("count to infinity problem"). Poisoned reverse does not solve the problem of counting to infinity if three or more nodes are involved. One possible workaround is to define ∞ as a small value (e.g. ∞ := 16).

Solution:

		X			**Y**		**Z**	
	dst=A	via A	via Y	via Z	via X	via Z	via X	via Y
$t = 0$	before the failure	1	∞	∞	2	3	2	3
$t = 1$	after X sends its vector	★	∞	∞	∞	3	∞	3
$t = 2$	after Y sends its vector	★	4	∞	∞	3	∞	∞
$t = 3$	after Z sends its vector	★	4	∞	∞	∞	∞	∞
$t = 4$	after X sends its vector	★	4	∞	∞	∞	5	∞
$t = 5$	after Y sends its vector	★	∞	∞	∞	∞	5	∞
$t = 6$	after Z sends its vector	★	∞	∞	∞	6	5	∞
$t = 7$	after X sends its vector	★	∞	∞	∞	6	∞	∞
$t = 8$	after Y sends its vector	★	7	∞	∞	6	∞	∞

Add the distance vectors to this table

7.3 Visibility (Exam Question 2016)



A simple BGP network

Consider now the network depicted on the left. Single-headed plain arrows point from providers to their customers (AS A is the provider of AS D), while double-headed dashed arrows connect peers (AS D and AS E are peers). Each AS in the network originates a unique prefix that it advertises to all its BGP neighbors. Each AS also applies the default selection and exportation BGP policies based on their customers, peers and providers.

a) What path (sequence of ASes) is followed when AS G sends packets destined to the prefix originated by AS E?

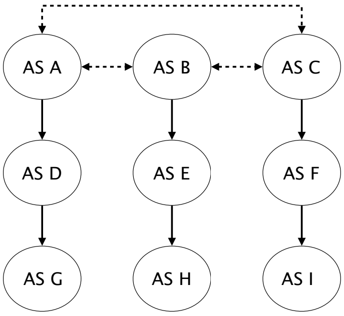
Solution: Path: [G, D, E]

b) What path (sequence of ASes) is followed when AS F sends packets destined to the prefix originated by AS G?

Solution: Path: [F, C, A, D, G]

c) Suppose AS A and AS C give you a "dump" of all the BGP routes they *learn* for every destination. You then extract all links from the AS paths seen in those "dumps" and use them to construct a view of the AS-level topology. Draw the resulting AS-level topology in the figure below.

Solution:



Solution

d) Give the minimum set of ASes that must provide a "dump" of each route they learn s.t. all the edges (the ones in the figure on the left) are visible? Justify your answer.

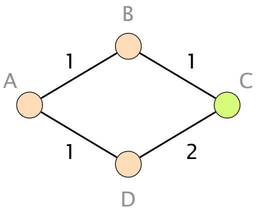
Solution: (A, H) or (C, H) are two possible answers (there are more possibilities). The set is of size 2. H "sees" all the links with exception of the top peering link between A and C. For this link you need A or C.

7.4 Convergence (Exam Style Question)

Consider this simple network running OSPF as link-state routing protocol. Each link is associated with a weight that represents the cost of using it to forward packets. Link weights are bi-directional.

Assume that routers A, B and D transit traffic for an IP destination connected to C and that link (B, C) fails. Which nodes among A, B and D could potentially see their packets being stuck in a transient forwarding loop? Which ones would not?

Solution: Nodes A and B could see their packets stuck in a forwarding loop if B updates its forwarding table before A, which is likely to happen as B would be the first to learn about an adjacent link failure. On the other hand, D would not see any loop as it uses its direct link with C to reach any destination connected beyond it.



Loopy or not?

Assume now that the network administrator wants to take down the link (B, C), *on purpose*, for maintenance reasons. To avoid transient issues, the administrator would like to move away all traffic from the link *before* taking it down and this, without creating any transient loop (if possible). What is the minimum sequence of increased weights setting on link (B, C) that would ensure that *no packet* destined to C is dropped?

Solution: One example of a minimum sequence of weight settings is [1, 3, 5].

Note: The problem highlighted above happens because B shifts traffic to A before A shifts traffic to D, hence creating a forwarding loop. By setting the (B, C) link weight to 3, (only) A shifts from using (A, B, C) to using (A, D, C). Once A has shifted, it is safe to shift B by setting the link weight to 5 (or higher). Once B has shifted has well, the link can be safely torn down.