Pruning IP Forwarding Tables

Consider an IP router with a forwarding table composed of the 9 entries depicted on the right.

Write down an equivalent forwarding table by combining entries together into shorter ones such that the resulting table has the least number of entries. Your reduced forwarding table should be such that the forwarding decision made by the router for any IP packet is equivalent to the initial one.

prefix	next-hop
82.130.32.0/20	1
82.130.64.0/20	1
82.130.80.0/20	2
82.130.96.0/20	1
82.130.112.0/21	1
82.130.120.0/21	1
82.130.122.0/24	1
82.130.123.0/24	1
82.130.124.0/24	2

Pruning IP Forwarding Tables

Consider an IP router with a forwarding table composed of the 9 entries depicted on the right.

Write down an equivalent forwarding table by combining entries together into shorter ones such that the resulting table has the least number of entries. Your reduced forwarding table should be such that the forwarding decision made by the router for any IP packet is equivalent to the initial one.

prefix	next-hop
82.130.32.0/20	1
82.130.64.0/18	1
82.130.80.0/20	2
82.130.124.0/24	2

Pruning IP Forwarding Tables

Network admins also can install “default routes” that are catch-alls for traffic that doesn’t fit into a specific rule or for those they want to aggregate. The default route is typically 0.0.0.0/0

How would the table change with a default route?

prefix	next-hop
82.130.32.0/20	1
82.130.64.0/20	1
82.130.80.0/20	2
82.130.96.0/20	1
82.130.112.0/21	1
82.130.120.0/21	1
82.130.122.0/24	1
82.130.123.0/24	1
82.130.124.0/24	2

Pruning IP Forwarding Tables

Network admins also can install “default routes” that are catch-alls for traffic that doesn’t fit into a specific rule or for those they want to aggregate. The default route is typically 0.0.0.0/0

How would the table change with a default route?

prefix	next-hop
0.0.0.0/0	1
82.130.80.0/20	2
82.130.124.0/24	2

prefix	next-hop
82.130.32.0/20	1
82.130.64.0/20	1
82.130.80.0/20	2
82.130.96.0/20	1
82.130.112.0/21	1
82.130.120.0/21	1
82.130.122.0/24	1
82.130.123.0/24	1
82.130.124.0/24	2

NAT

There are two pcaps: NAT_home_side.pcap and NAT_ISP_side.pcap

Open the NAT_home_side file.

1. What is the IP address of the client?

2. The client actually communicates with several different Google servers in order to implement “safe browsing.” The main Google server that will serve up the main Google web page has IP address 64.233.169.104. In order to display only those frames containing HTTP messages that are sent to/from this Google, server, enter the expression “http && ip.addr == 64.233.169.104” (without quotes) into the Filter: field in Wireshark.

3. Consider now the HTTP GET sent from the client to the Google server (whose IP address is IP address 64.233.169.104) at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

4. At what time is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

5. Before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. At what time is the client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267? What are the source and destination IP addresses and source and destination ports for the TCP SYN segment? What are the source and destination IP addresses and source and destination ports of the ACK sent in response to the SYN. At what time is this ACK received at the client? (Note: to find these segments you will need to clear the Filter expression you entered above in step 2. If you enter the filter “tcp”, only TCP segments will be displayed by Wireshark).

NAT

There are two pcaps: NAT_home_side.pcap and NAT_ISP_side.pcap

Open the NAT_home_side file.

1. What is the IP address of the client?
   192.168.1.100
2. The client actually communicates with several different Google servers in order to implement “safe browsing.” The main Google server that will serve up the main Google web page has IP address 64.233.169.104. In order to display only those frames containing HTTP messages that are sent to/from this Google, server, enter the expression “http && ip.addr == 64.233.169.104” (without quotes) into the Filter: field in Wireshark.

3. Consider now the HTTP GET sent from the client to the Google server (whose IP address is IP address 64.233.169.104) at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?
   192.168.1.100 -> 64.233.169.104, Ports 4335 -> 80
4. At what time is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?
   Time 7.158797, 64.233.169.104 -> 192.168.1.100, Ports 80 -> 4335
5. Before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. At what time is the client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267? What are the source and destination IP addresses and source and destination ports for the TCP SYN segment? What are the source and destination IP addresses and source and destination ports of the ACK sent in response to the SYN. At what time is this ACK received at the client? (Note: to find these segments you will need to clear the Filter expression you entered above in step 2. If you enter the filter “tcp”, only TCP segments will be displayed by Wireshark).
   Time 7.07675, 192.168.1.100 -> 64.233.169.104, 4335 -> 80, ACK received 7.108986, reverse IPs and ports

NAT

Open the NAT_ISP_side file.

1. Find the HTTP GET message was sent from the client to the Google server at time 7.109267 (where t=7.109267 is time at which this was sent as recorded in the NAT_home_side trace file). At what time does this message appear in the NAT_ISP_side trace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET (as recording in the NAT_ISP_side trace file)? Which of these fields are the same, and which are different, than in your answer to question 3 above?

2. Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: Version, Header Length, Flags, Checksum. If any of these fields have changed, give a reason (in one sentence) stating why this field needed to change.

3. In the NAT_ISP_side trace file, at what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different than your answer to question 4 above?

4. In the NAT_ISP_side trace file, at what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding to the segments in question 5 above captured? What are the source and destination IP addresses and source and destination ports for these two segments? Which of these fields are the same, and which are different than your answer to question 5 above?

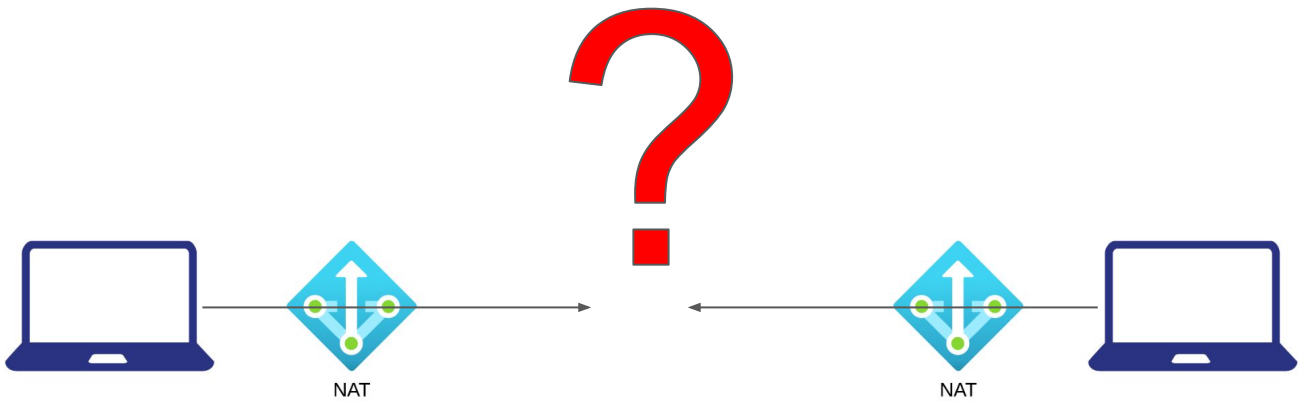
NAT

Open the NAT_ISP_side file.

1. Find the HTTP GET message was sent from the client to the Google server at time 7.109267 (where t=7.109267 is time at which this was sent as recorded in the NAT_home_side trace file). At what time does this message appear in the NAT_ISP_side trace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET (as recording in the NAT_ISP_side trace file)? Which of these fields are the same, and which are different, than in your answer to question 3 above?
   6.069168, Src: 71.192.34.104, Dst: 64.233.169.104, Src Port: 4335, Dst Port: 80
2. Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: Version, Header Length, Flags, Checksum. If any of these fields have changed, give a reason (in one sentence) stating why this field needed to change.
   Checksum is different because it is calculated over the fields of the header, including a different source IP address
3. In the NAT_ISP_side trace file, at what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different than your answer to question 4 above?
   6.117570, Src: 64.233.169.104, Dst: 71.192.34.104, Src Port: 80, Dst Port: 4335
4. In the NAT_ISP_side trace file, at what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding to the segments in question 5 above captured? What are the source and destination IP addresses and source and destination ports for these two segments? Which of these fields are the same, and which are different than your answer to question 5 above?
   6.035475 and 6.067775, Src: 71.192.34.104, Dst: 64.233.169.104 and reverse, Src Port: 4335, Dst Port: 80 and reverse

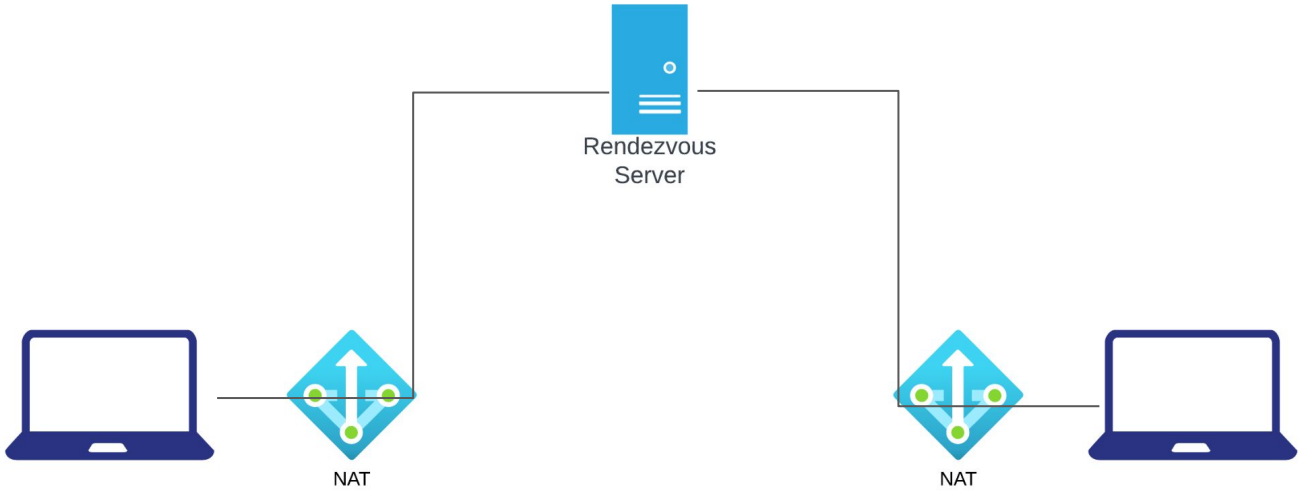
Connectivity Between NAT Machines

If you have two machines behind NATs that wish to communicate, how do they know the correct source and destination IPs / ports to use?



Hole Punching

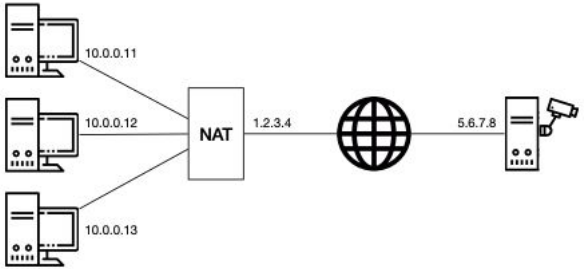
Each machine sets up a connection to a publicly reachable rendezvous server, which then relays the streams for the clients. This is known as hole punching



NAT

Consider the network topology shown. 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.

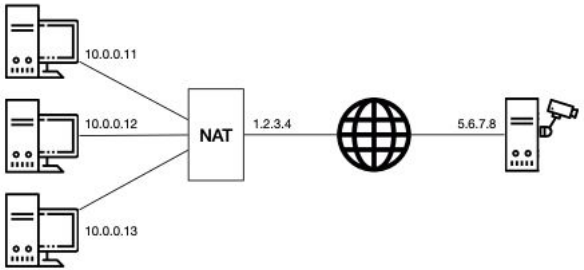
Alice wants to receive the live video stream on one of her PCs and thus configures the camera to send the video signal to IP 10.0.0.11 and port 1234. However, she does not receive it on her PC. Why? Where is this traffic sent to?



Alice operates three PCs and one camera server

NAT

Consider the network topology shown. 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.



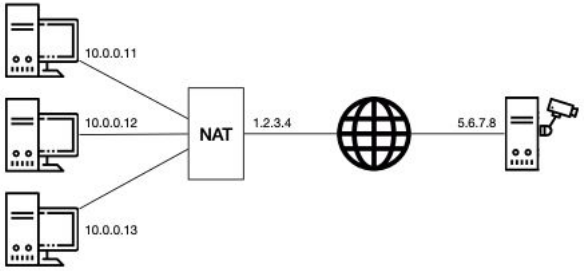
Alice operates three PCs and one camera server

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.

NAT

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?

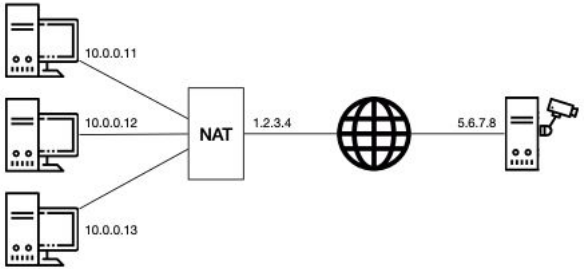


Alice operates three PCs and one camera server

NAT

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.

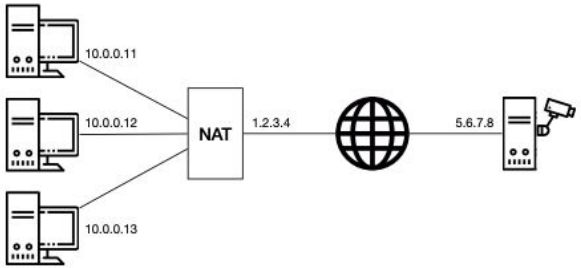


Alice operates three PCs and one camera server

NAT

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.



Alice operates three PCs and one camera server

NAT

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:

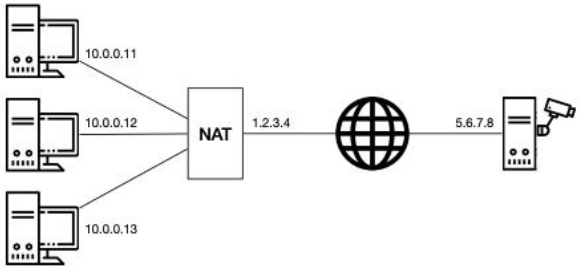
- Send one single UDP packet with arbitrary source and destination addresses and ports from each of her PCs;
- observe the received packets at each of her PCs and the camera server;
- specify the destination IP address and port for the video signal.

Solution: First, you want to “punch 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 (or to a rendezvous server). 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 internal 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.



Alice operates three PCs and one camera server

IPv4 and IPv6 - Many OSes and Applications Use a Dual Stack