**Bold – Complete change to the part**

*Italics – Minor changes*

~~Strike out~~ – No changes needed

* Problem statement
  + ~~Title, team, URL~~
  + **Description of problem**
* Components
  + *Platform(s) for project – Kernel version*
  + ~~Area(s) for project~~
  + **Major components decomposition**
* Preliminary design and development plan
  + **High-level proposed design**
  + **Per-component development phases**
  + ~~Team composition~~
  + *Per-member responsibilities*
  + **Per-member learning objectives**
  + **Timeline**
  + **Descriptive Diagram of the system**
* Verification and validation plan
  + *Test application*
  + *Test plan*
  + **Demo playbook**
* Design and development plan (replaces preliminary section)
  + **High-level design (replaces tentative design)**
  + **Low-level design – Same as per component**

1 Load balancing NAT using least rate:

a) Destination NAT:

Destination NAT functionality is placed in the PREROUTING hook of the netfilter architecture as shown in Figure 2. It modifies the destination IP address of a packet to one of the server IP addresses in the private network. The selection of server for a connection is based on least rate scheduling.

The steps taken by the DNAT for incoming packets (packets from the client) before routing is shown in Figure 3.

The destination IP address and port of packets is compared with the assigned public IP address and public port of the NAT and the packets are dropped if either or both of the comparison fail. If both match, the packet is destined to one of the servers.

The load balancing is accomplished by maintaining 2 tables. Table 1 consists of the mapping from {Client IP address, Client port} to {Server IP address}. {Client IP address, Client port} in Table 1 is searched by using {Source IP address, Source port} of the UDP packet as the key. If the key is found, the destination IP address of the packet is changed to the corresponding Server IP address. If the key is not found, it corresponds to a new connection and an entry {Client IP address, Client port, Server IP address} is added to Table 1. The destination IP address of the packet is changed to the corresponding server IP address. The selection of Server IP address is based on the packet rate entry for each server in Table 2.

Server IP address = IP address of the server with the minimum rate of packets flowing through it as per Table 2.

The timestamp for the corresponding row in Table 1 is updated for each packet. To compute the packet rate in Table 2 for the assigned server, the packet count of the server is incremented by 1.

Before the packet is sent to the routing function, the UDP and IP checksum values are updated. This is done as the destination IP address of the packet is changed by the DNAT function.

The timestamps in Table 1 serve the purpose of replacing table entries when the table is full. If the number of rows in the table has reached the maximum limit when a new connection request arrives, the entry with the earliest timestamp is replaced with the current value of {Client IP address, Client port, Server IP address}. This functionality helps to remove inactive entries in the table. The maximum size of the table is chosen so that active connections are not compromised by the deletion function.

b) Source NAT:

Source NAT functionality is placed in the POSTROUTING hook of the netfilter architecture as shown in Figure 2. It modifies the source IP address of a packet from the server to the public IP address of the NAT in order to hide the IP addresses in the private network.

The steps taken by the SNAT for outgoing packets (packets from the servers to the clients) after routing is shown in Figure 4.

The source IP of the packet from the server is changed to the public IP address of the NAT. This hides the private network from the clients as the server IP addresses are not known to the clients.

Table 1 is searched with the key {Destination IP address, Destination port}. The packet is dropped if the key is not found. If the entry is found, the timestamp for the corresponding row in Table 1 is updated. The packet count of the server is incremented by 1.

Before the packet is sent out to the network, the UDP and IP checksum values are updated. This is required as the source IP address of the packet is changed by the SNAT function.

2 Load balancing NAT using round robin:

a) Destination NAT:

Destination NAT functionality is placed in the PREROUTING hook of the netfilter architecture as shown in Figure 2. Destination NAT performs IP address translation similar to the DNAT in least rate as given in section 1a but the load balancing is based on round robin algorithm.

The steps taken by the DNAT for incoming packets (packets from the client) before routing is shown in Figure 5.

The destination IP address and port of packets is compared with the assigned public IP address and public port of the NAT and the packets are dropped if either or both of the comparison fail. If both match, the packet is destined to one of the servers.

The load balancing is accomplished by using Table 1. Table 1 consists of the mapping from {Client IP address, Client port} to {Server IP address}. {Client IP address, Client port} in Table 1 is searched by using {Source IP address, Source port} of the packet as the key. If the key is found, the destination IP address of the packet is changed to the corresponding Server IP address. If the key is not found, it corresponds to a new connection and an entry {Client IP address, Client port, Server IP address} is added to Table 1. The destination IP address of the packet is changed to the corresponding server IP address. The selection of Server IP address is based on round robin. For the first connection, Server 1 is chosen and for further connections, the chosen IP address is given by:

Server IP address = (Server IP address assigned to previous connection) modulo (Total number of servers).

The computation of timestamp, deletion of entries from Table 1 and the checksums in the packet headers are updated similar to DNAT for least rate.

b) Source NAT:

Source NAT functionality is placed in the POSTROUTING hook of the netfilter architecture as shown in Figure 2. Source NAT performs IP address translation similar to the SNAT in least rate as given in section 1b.

The steps taken by the SNAT for outgoing packets (packets from the servers to the clients) after routing is shown in Figure 6.

Table 1 is searched with the key {Destination IP address, Destination port}. The packet is dropped if the key is not found. If the entry is found, the source IP of the packet from the server is changed to the public IP address of the NAT. The timestamp for the corresponding row in Table 1 is updated and the checksums are computed as given in section 1b.

* + (Other sections as before)
* *Updates to V&V plan if any*
* **Self-study Plan**
  + **Description of base case**
  + **Characteristics to observe**
  + **Range of scenarios to investigate**