Static and source based routing

Lab setup

For this lab students have to work in teams of two. Two team of two students (that is overall four students) should form a group and perform lab tasks together. It is important that teams belonging to same group, use near by computers so that they can discuss configuration done and output obtained after each step.

In this lab handout term 'partner team' is used to refer to other team in same group. So if there are two teams, say 'A' and 'B' which together form a group, say 'group1' and the lab handout mentions "ping to eth0 interface of partner team". Then students in team A are supposed to ask students in team B, IP address of their machine and try to ping team B's machine from their own machine.

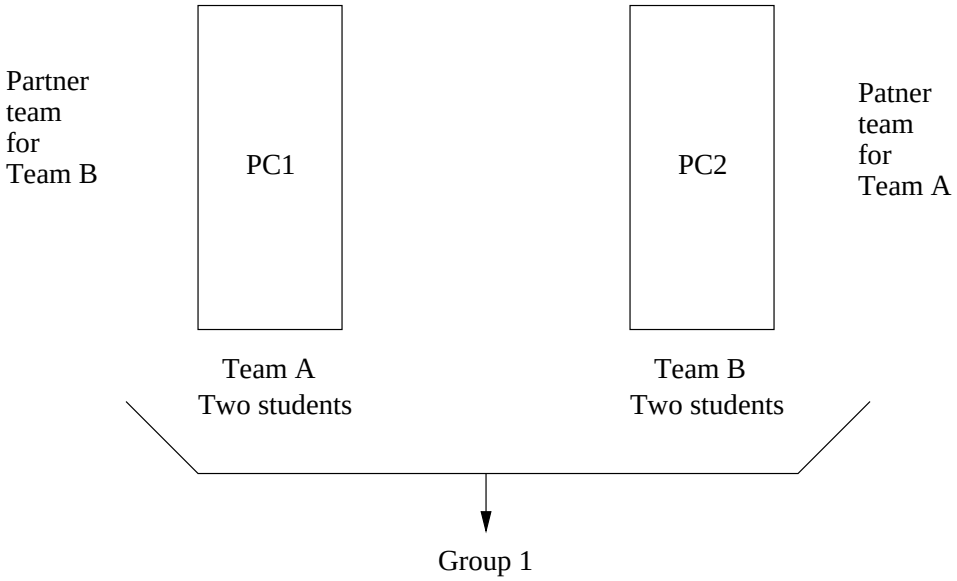

Figure 1: Team and Group

Similarly for the same line students in team B are supposed to ask students in team A, IP address of their machine and try to ping team A's machine from their own machine.

There are lab queries mentioned in lab handout. One common answer for lab queries should be submitted by any one member of group on courses portal before 27th January 2011, evening 05:00pm.

Booting into non-xen kernel with minimum interfaces

1. Boot into non-xen kernel.

2. Run '`ifconfig`' and ensure that only two interfaces eth0 and lo are visible. If you see interfaces with names like peth0, virbr0, xenbr0, br0, etc. then do not proceed. On few machines the interface name could be eth1, eth2, etc. Also on few machines you can have multiple interfaces like eth1, eth2, etc. These is due to those machine having multiple NICs.

   If you see interface named 'xenbr0' then you have not booted into non-xen kernel.

3. If you see interface named virbr0 then use commands:

   ```
   service libvirtd stop
   /etc/xen/scripts/network-bridge stop bridge=virbr0
   service network restart
   ifconfig
   ```

   this should remove interface named virbr0 from list.

4. Run command '`service network restart`'.

5. Run command '`ifconfig`'. Now also you should see only two interfaces eth<n> and lo. On machines with more than one network card you can see all interfaces available using '`ifconfig -a`' command. Note that by default `ifconfig` only reports interfaces which are up.

   Usually in lab on machines with multiple interfaces since only one interface is up and running, by default `ifconfig` shows only one interface. You may see interface named '`sit0`' which is used for IPv6 in IPv4 tunneling, ignore it.

6. Ping to IP address of lo interface. Then ping to IP address of eth<n> interface. Try to ping to IP address of eth<n> interface of partner team. Do not proceed if you are not able to ping partner machine. Connectivity between the two machines is important for performing tasks mentioned in rest of the handout.

   Query: Look at RTT values obtained while pinging IP address of eth<n> of your own machine and IP address of eth<n> of partner team machine. Why is huge difference between the two RTT values?

Configuring temporary interface aliases

1. We can use `ifconfig` command to configure interface aliases. If the name of original interface is eth0, then aliased interfaces can be named eth0:0, eth0:1, etc.

   There can be very large number of such aliased interfaces. Use the `ifconfig` command to create aliased interface eth<n>:0 with IP address 172.18.<group_number>.<team_number>.

   To create a liased interface using `ifconfig` command you only need to replace interface name eth0, eth1, etc. in normal ifconfig command with aliased interface name eth0:0, eth1:0, etc. Use netmask 255.255.255.0 while configuring aliases interface.

   Query: What are the advantages of using above scheme for IP addressing? Mention as many reasons as you can think of.

   The lab handout is written using group_number 10 and team_numbers 100 and 101. Hence handout will use IPs 172.18.10.100 and 172.18.10.101 at various places. You should replace these IPs with your own IP addresses obtained using group number and team number throughout rest of the lab.

2. Ping to your own aliased interface IP address. Ping to your partner teams aliased interface.

   Query: How is ping working with your own defined IP address? Has any special configuration been done in lab network so that these new IP addresses also work? If not, does this mean that any one can configure any IP address he / she wants and start using it for communication? How can then we hold some one accountable based on IP address logs as any one could have been using that IP address? Elaborate.

3. Use '`ifdown eth0`'. (Replace eth0 with appropriate eth<n> for rest of the handout, in case interface on your machine is not named eth0). Now use '`ifup eth0`'. Now run '`ifconfig`' command. This would cause aliased interface to get removed.

   Query: Why aliased interface got removed after using ifdown / ifup.

Understading normal routing table

1. Use command 'route -n and look at the routing table. You should find three routes like this:
   (Flags, Metric, Ref, Use columns have been removed)

```
Kernel IP routing table
Destination     Gateway         Genmask         Iface
10.3.3.0        0.0.0.0         255.255.255.0   eth0
169.254.0.0     0.0.0.0         255.255.0.0     eth0
0.0.0.0         10.3.3.1        0.0.0.0         eth0
```

   The first route is added automatically when interface is configured with IP address in range 10.3.3.0/24. This route signifies that all machines in 10.3.3.0/24 can be reached directly via eth0 without requiring any gateway or another Layer 3 device in between.

   Hence whenever we configure an IP address in Linux machine, a route also gets added that signifies that machines in same subnet as that of IP address just configured, can be reached directly without requiring any gateway as we have IP address in the network.

   The second route is used for zeroconf service discovery. In linux there is service called 'avahi-daemon' which can be used for zeroconf service discovery and mDNS protocol. Unfortunately zeroconf is very unsecure due to its trusting nature and avahi-daemon is configured to run automatically on most Linux distributions.

   Thus it is very important for tech-savvy users to stop avahi-daemon on properly configured infrastructure networks to avoid DNS poisioning and other service discovery attempts using zeroconf. To stop avahi-deamon use:

```
service avahi-daemon stop
chkconfig avahi-daemon off
```

   Query: Why people who design Linux distributions would make such a poor security choice of having avahi-daemon run by default after new Linux installation? Comment on whether avahi-daemon is really as bad as described above on infrastructure networks where machines have obtained proper IPs using DHCP server or appropriate IPs have been configured statically.

2. Now run '`route -n`' command again and you would notice that route for 169.254.0.0/16 is still present. Now use command

```
route del -net 169.254.0.0 netmask 255.255.0.0
```

   This would cause route for 169.254.0.0/16 to get deleted. Verify using '`route -n`' command that route actually got deleted. Now use '`service network restart`' and '`route -n`' command. You would observe that route for 169.254.0.0/16 again got added.

   Query: How route for 169.254.0.0/16 gets added when we use '`service network restart`'?

3. The third and the last route which shows up in '`route -n`' command output is the route that defined default gateway and hence is also sometimes called default route. This route is used when no other route in routing table matches.

   This route is very important as most end user machines are connected to edge networks and these machines are connected to Internet / LAN via only single edge router. Hence to reach to any other machine which is not directly connected the edge hosts have to send packets to edge router to which they are connected.

   This third route or default route in example routing table above causes machine to send packets to '10.3.3.1' in case no other route line matches.

   Query: Why do all default routes end with .1 like 10.3.3.1, 10.3.1.1, 10.1.67.1, etc.? Is it necessary for default route IP addresses to end with .1?

   Query: In case your answer to above question is that it is not necessary for default gateway IP addresses to end with .1 then who told machine that 10.3.3.1 is the IP address of default router? Why did not machine assume IP address of default router to be 10.3.3.17, if that is also possible?

   Query: What type of device has IP address 10.3.3.1 in our network? (a) Router (b) L3 Switch (c) L2 switch (d) Linux machine (e) Windows machine (f) Firewall How many other IPs does that device has besides 10.3.3.1?

4. Again configure alias with IP 172.18. <group_number> . <team_number> /24. Now ping partner teams IP 172.18. <group_number> . <partner_team_number>. Now run command '`route -n`' and notice that a line similar to

```
Destination     Gateway         Genmask         Iface
172.18.10.0     0.0.0.0         255.255.255.0   eth0
```

gets added to the routing table.

Query: Will ping to your partner teams IP work without this route? Provide a test with which you can prove that your answer to this question is correct.

Query: The interface name in route is stored as eth0 and not eth0:0 or eth0:1. Is it going to cause problem while using lot of aliased interfaces and having various routes using those aliased interfaces?

Configuring permanent interface aliases

Now use '`service network restart`' and run '`ifconfig`' to ensure that aliased interface is actually down. Go to folder '`/etc/sysconfig/network-scripts`'. Copy file ifcfg-eth0 to ifcfg-eth0:0. Edit file ifcfg-eth0:0 and put appropriate values for various parameters such that next time we run '`service network restart`' command we see both eth0 and eth0:0 wtih IP address 172.18 . <group_number> . <team_number> /24.

Query: What value of GATEWAY=" " you have used in configuration file ifcfg-eth0:0 and why?

Understanding ARP

Ping to partner teams 10.3.3.0/24 series IP and IP within range 172.18.0.0/16. Now use commnad '`arp -a`. Now delete all entries using '`arp -d <IP address>`". You should see something like

```
Address         HWaddress               Iface
10.3.3.232      (incomplete)            eth0
10.3.3.1        (incomplete)            eth0
172.18.10.101   (incomplete)            eth0
```

after deleting all ARP entries. Now ping from only one machine to partner teams machine. Only one team should ping and not both. For example ping from 172.18.10.100 to 172.18.10.101 and not vice-versa. Do same for 10.3.3.0 series IP address.

Now run '`arp -n`' command on both machines. You would see that both machines again cached ARP entries.

> **Query:** Why has destination machine cached ARP entries (172.18.10.101 in above example)? Does destination machine (172.18.10.101 in above example) also performs ARP query for source machine to learn MAC address of source machine, so that it can construct proper reply packets?
> (Verify your answer using some test in lab before writing.)

IP forwarding

Use command '`sysctl net.ipv4.ip_forward`' to check value for IP forwarding. If it is '1' then use '`sysctl net.ipv4.ip_forward = 0`' to disable IP forwarding.

Follow below steps only on one machine not on both machines:

1. Now use '`route del -net 172.18.10.0 netmask 255.255.255.0`'
2. Try to ping to 172.18.10.100 or 172.18.10.101 whichever is partner teams IP.
3. Now add route '`route add -net 172.18.10.0 netmask 255.255.255.0 gw <partner_teams_10.3.3.0/24_series_IP>`'
4. Now again try to ping 172.18.10.100 or 172.18.10.101 whichever is partner teams IP.
5. Delete ARP entry for 172.18.10.100/101 if cached.
6. Now again try to ping 172.18.10.100 or 172.18.10.101 whichever is partner teams IP.
7. Look at '`arp -n`' output.

Query: How is ping to 172.18.10.100 / 101 working without any ARP entry for destination IP? Run commnad '`traceroute 172.18.10.101`'. Can you verify your answer using traceroute command output? Did you account for fact that IP forwarding is disabled on both machines while giving explaination on how ping to 172.18.10.100 is working?

Static routing

Change ifcfg-eth0:0 file and now change IP address to 172.18. <team_number> . <group_number> from 172.18. <group_number> . <team_number> on both your and your partners machine. Use '`service network restart`'. Try to ping to your partners IP address.

Query: Why machines are not able to communicate while giving IP of type 172.18. <group_number> . <team_number> when they were able to communicate without any problem while using IP of type 172.18. <team_number> . <group_number>?

Query: How can we make 172.18.100.10 and 172.18.101.10 IPs communicate without:

1. Changing existing IP address or netmasks.
2. Adding more aliases to system
3. Using any other L3 device other than these two machines in between.

For all below exercises two groups should work together so that you have four machines, say PC1, PC2, PC3 and PC4.

Configure random unused subnets as aliases in PC1 and PC4. Configure IPs in same subnet in PC2 and PC3 as aliases. All the machines PC1-4 should still have an IP in 10.3.3.0/24 range as primary eth0 IP. Now make ping from PC1 to PC4's aliased IP work. Also make ping from PC4 to PC1's aliases IP work.

The packet should traverse through PC2 and PC3's aliased IPs while going from PC1 to PC4 or vice-versa. Use traceroute to verify that packet is actually travelling using desired path.

Source based routing

Basic steps

With advanced IP routing II features of Linux we can route based on source, destination and various other parameters. To configure IP routing II features for routing, the basic steps are:

1. Create routing tables
2. Add routes to routing tables
3. Add rules for kernel IP routing

Creating routing tables

To create routing table we have to edit file '`/etc/iproute2/rt_tables`'. In this file we have to add lines like at end

```
200 dmztolocal
210 intranettolocal
220 publictolocal
230 localtopublic
```

Here 200, 210, etc. are routing table numbers. The gap of 10 is just example we can use 200, 201 or 200, 300 any sort of numbers. 'dmztolocal', 'intranettolocal' etc. are names of routing tables. This is one time thing and this information would persists after rebooting.

Adding routes to routing tables

We can add routes to routing tables with command '`ip route add table <table_name> via <gateway>`'. We can also specify gateway specific to destinations as we do in normal routing with syntax '`ip route add table <table_name> <destination> via <gateway>`'.

Here, table_name is name of routing table that we specify in file '`/etc/iproute2/rt_tables`', destination can be entered in IP address/mask format to specify both single host or network, gateway can be any IP reachable by directly connected interfaces.

Using this approach we can populate all routing tables that we have created. The routing within routing table is done with longest prefix match the way it is done in normal routing tables.

Note that routes added to routing table do not persist after rebooting. Hence permanent configuration of routing should be done by adding route

lines in script and calling that script in some start-up file like '`/etc/rc.d/rc.local`' or from '`/etc/init.d/network`', etc.

Seeing routes of particular table

We can use command '`ip route list table <table_name>`' to see routing entries of particular table.

Seeing default rules

We can see existing routing rules by '`ip rule list`' command. The default rule list looks like

```
0:      from all lookup 255
32766:  from all lookup main
32767:  from all lookup default
```

Here 255, main and default are name of tables and 0, 32766, 32767 etc. are priorities. We can list the rules stored in tables 255, main or default too.

255 table generally has rules related to broadcast or directly connected interfaces. Table main has routing table that we typically see with '`route -n`' command. 'default' table is usually empty.

Note: It is possible to delete rules for table main and default but there is no need to do it. We can always add routes with higher priority without disturbing existing tables.

Adding rules which specify which routing table to use

After creating and populating various routing tables we can add rules which based on source, destination and other selection parameters determine which routing table should be used. To add rule we can use syntax: '`ip rule add [ from <source network or host> ] [ to <destination network or host > ] [ pref <pref number >] table <table name>`'.

Note that except table name most parameters from, to, pref number etc. are optional.

Rules added via above approach are not persistent same as routes added to routing tables are not persistent. Hence even these lines should be added to some script for permanent configuration.

Removing rules from rules table

We can remove rules from rules table by using syntax '`ip rule del pref <pref number>`'. Here preference number is same that we specify while adding rule or listed when we use '`ip rule list`'. We can specify more constraints in case multiple rules with same preference exist in rule table.

Do not use route command

After using IP routing II the output of normal route command cannot be trusted as it will just list rules stored in main table which may not have higher priority than rules that we added in custom table. Hence '`route`' command should not be used at all while using IP routing II.

Exercise

Configure PC1, PC2, PC3 with IPs in 10.3.3.0 series for eth0 using DHCP. Configure PC1, PC2, PC3 with IPs in some other range using interface alias like eth0:0. Remove 10.3.3.0 series DHCP IP from PC4. This way PC1, PC2 and PC3 are in same network with two different types of IP ranges and can communicate with each other using any of the two range IPs.

Configure IPs of some other subnet in PC2, PC3 and PC4. Do not configure IP for this subnet on PC1. This way PC1, PC2 and PC3 belong in same network with two different IP ranges and PC2, PC3 and PC4 belong in some other network with some third IP range.

Now configure source based routing such that when PC1 tries to ping to PC4 then packet travels through PC2 if PC1 uses IP in 10.3.3.0 while trying to ping. If however PC1 tries to ping PC4 using its other network IP then the packet should travel through PC3. Verify using traceroute and ping whether source routing is working or not.

Use man page to learn about '`traceroute -s <IP>`' and '`ping -I <IP>`' options. You can also try to use traceroute with -I switch to make traceroute use ICMP and not UDP.

Load balancing between two gateways

Theory

For load balancing between two gateways use:

```
ip route del default \\
ip route add default scope global \
```

```
nexthop via <gateway 1> dev <interface of gateway 1> weight 1 \
nexthop via <gateway 2> dev <interface of gateway 2> weight 1
```

Here weights can be changed to make one route more preferable to other. We can use '`ip route show`' command to verify that routes got added properly.

Exercise

1. Configure PC1, PC2 and PC3 with IPs in some random subnet. Also configure PC2 and PC3 to take IPs in range 10.3.3.0/24 using DHCP. Now configure routing such that PC1 uses PC2 and PC3 as gateways for reaching outside network.

   Try to connect to various networks using PC1 and verify that some connections are going via PC2 and some other are going via PC3 and that both gateways are being used for balancing load.

   Query: Why did we remove 10.3.3.0/24 series IP from PC1 in above case.

   Detailed steps on how to perform above exercise are:

   ON PC1 : (a) Remove 10.3.3.0/24 series IP from PC1.
   (b) Configure PC1 with 172.18.10.100/24 and no gateway.
   (c) Configure PC1 to load balance using '`ip route add default scope global nexthop via 172.18.10.101 dev eth0 weight 1 nexthop via 172.18.10.102 dev eth0 weight 1`'
   (d) Disable firewall using '`iptables -F`'

   ON PC2 : (a) Configure additional alias for 172.18.10.101/24 without removing 10.3.3.0/24 series IP.
   (Assuming 10.3.3.31 is the 10.3.3.0/24 series IP for PC2)
   (b) Use commands:
   1. `iptables -F`
   2. `iptables -t nat -I POSTROUTING \! -s 10.3.3.31 \! -d 172.18.10.0/24 -o eth0 -j SNAT --to-source 10.3.3.31`
   3. `sysctl net.ipv4.ip_forward=1`

   ON PC3 : (a) Configure additional alias for 172.18.10.102/24 without removing 10.3.3.0/24 series IP.
   (Assuming 10.3.3.74 is the 10.3.3.0/24 series IP for PC3)

(b) Use commands:
   1. `iptables -F`
   2. `iptables -t nat -I POSTROUTING \! -s 10.3.3.74 \! -d 172.18.10.0/24 -o eth0 -j SNAT --to-source 10.3.3.74`
   3. `sysctl net.ipv4.ip_forward=1`

2. Now add 10.3.3.0/24 series IP for PC1 in above case and do load balancing on 10.3.3.0/24 IPs of PC2 and PC3. Capture packets on PC1 using wireshark. Do you see any ICMP redirect message? Use IP routing commands and see whether routing tables have changed because of ICMP redirect messages?

Query: If routes have not changed then why have routes not changed even after receving ICMP route redirect messages? How can be make system accept ICMP route change requests? If routes did change what is the security problem with accepting ICMP route redirect messages from any machine? How do we disable machine from accepting ICMP route redirect messages?

Query: Did we use NAT throughout this lab anywhere? If not how is routing differnet than NATing? Can you suggest how we can configure Linux machines to NAT?