

Week #10

ICMP Error Handling

Objective:

To understand ICMP Redirect and implement the following

- 1) ICMP Echo Request/reply
- 2) ICMP Redirect

Hardware Requirements:

- Desktop/Laptop – 4
- Switch – 2
- Patch Cords (1.5 Mtrs) – 5
- NIC Cards – 1

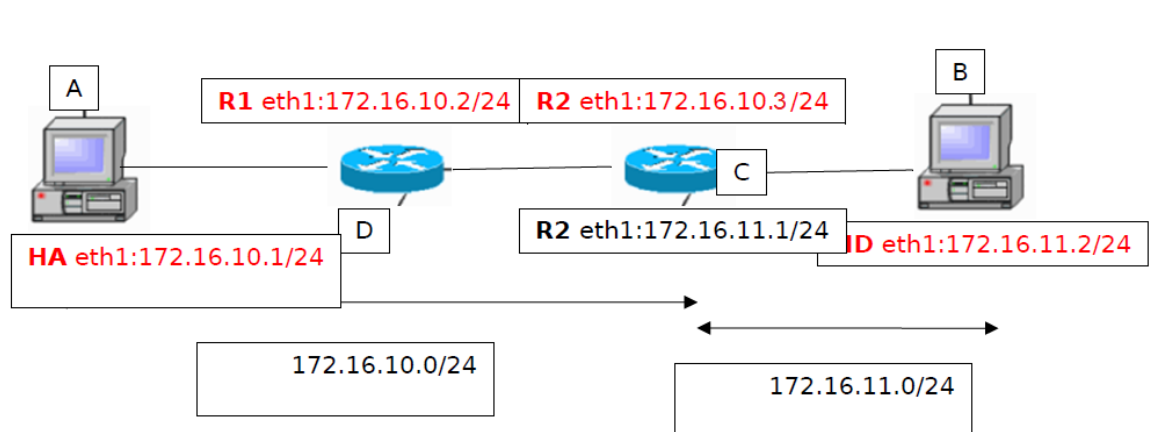
Software Requirements:

- Wireshark Tool
- Linux Operating System

Description:

This exercise shows a situation in which a router sends an ICMP redirect message to the host indicating alternative shortest/optimal path. Other possible ICMP errors include destination unreachable, TTL expiry, checksum errors, port unreachable etc.

Topology:



In the above Figure, H_A, R₁ and R₂ are connected to same subnet 172.16.10.0/24. The host H_B is configured with R₂ as router for communicating with network 172.16.11.0/24. When we try to communicate from H_A to H_B through R₁, **the router R₁ will send an ICMP redirect to H_A** indicating that the packets can directly be sent to R₂ instead of R₁.

Before ICMP Redirect path followed : H_A -> R₁ -> R₂ -> H_B

After ICMP Redirect path followed : H_A -> ~~R₁~~ -> R₂ -> H_B => **H_A -> R₂ -> H_B**

Step 1: Assign IP addresses to each computer

At H_A:

```
$ sudo ip addr add 172.16.10.1/24 dev eth1
```

```
$ ip addr show
```

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 50:e5:49:1b:f1:55 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.43/21 brd 192.168.15.255 scope global eth1
    inet 172.16.10.1/24 scope global eth1
    inet6 fe80::52e5:49ff:fe1b:f155/64 scope link
        valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

At H_B:

```
$ sudo ip addr add 172.16.11.2/24 dev eth1
```

```
$ ip addr show
```

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether b8:a3:86:98:42:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.44/21 brd 192.168.15.255 scope global eth1
    inet 172.16.11.2/24 scope global eth1
    inet6 fe80::baa3:86ff:fe98:42c9/64 scope link
        valid_lft forever preferred_lft forever
3: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 50:e5:49:1b:f1:78 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.44/21 brd 192.168.15.255 scope global eth2
    inet6 fe80::52e5:49ff:fe1b:f178/64 scope link
        valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

At R2:

\$ sudo ip addr add 172.16.11.1/24 dev eth2

\$ sudo ip addr add 172.16.10.3 /24 dev eth1

\$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether fc:75:16:e1:23:76 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.1/24 scope global eth2
    inet6 fe80::fe75:16ff:fe01:2376/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 50:e5:49:1b:f0:12 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.45/21 brd 192.168.15.255 scope global eth1
    inet 172.16.10.3/24 scope global eth1
    inet6 fe80::52e5:49ff:fe1b:f012/64 scope link
        valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$ |
```

At R1:

\$ sudo ip addr add 172.16.10.2/24 dev eth1

\$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 50:e5:49:1c:33:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.46/21 brd 192.168.15.255 scope global eth1
    inet 172.16.10.2/24 scope global eth1
    inet6 fe80::52e5:49ff:fe1c:33de/64 scope link
        valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$ |
```

Note 1: The machines are physically on the same LAN, thus you may get ICMP redirect messages from other machines (in case you make some configuration mistakes), so as a precautionary measure disable accepting the ICMP Redirect packets. By default, the linux enables accepting the ICMP redirect packets. To have precautionary measures issue below command line in Ha and Hd.

\$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.all.accept_redirects = 0
student@student-H81H3-I:~$ |
```

Note 2: Since machines are on same physical interface, the router is going to send ICMP redirect message disturbing the routing decision by hosts. Thus, disable sending of the ICMP redirect packets by these routers with aliased interfaces. To have precautionary measures issue below command line in R1 and R2.

```
$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
```

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.all.send_redirects = 0
student@student-H81H3-I:~$ █
```

Step 2: Convert C and D systems into routers R2 and R1 respectively by issuing below command.

Note 1: Check if IP forwarding is enabled or not.

We need to query the sysctl kernel value *net.ipv4.ip_forward* to see if forwarding is enabled or not.

```
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

Other alternative to check out if IP forwarding is enabled or not through the value in the /proc system:

```
$ cat /proc/sys/net/ipv4/ip_forward
0
```

The above command response states that forwarding is not enabled. So, we need to set ip_forward=1 to carry out IP forwarding.

Command to set the value of net.ipv4.ip_forward is as given below:

At R1:

```
$ sudo sysctl -w net.ipv4.ip_forward=1
```

At R2:

```
$ sudo sysctl -w net.ipv4.ip_forward=1
```

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Step 3: Verify the Local Network connection using ping command.

Initially test the connection of systems within the same network

At Ha:

\$ ping 172.16.10.1(Local network)

```
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_req=1 ttl=64 time=0.027 ms
64 bytes from 172.16.10.1: icmp_req=2 ttl=64 time=0.020 ms
64 bytes from 172.16.10.1: icmp_req=3 ttl=64 time=0.018 ms
64 bytes from 172.16.10.1: icmp_req=4 ttl=64 time=0.018 ms
64 bytes from 172.16.10.1: icmp_req=5 ttl=64 time=0.022 ms
^C
--- 172.16.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.018/0.021/0.027/0.003 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

At Hb:

\$ ping 172.16.11.1(Local network)

```
student@student-H81H3-I:~$ ping 172.16.11.1
PING 172.16.11.1 (172.16.11.1) 56(84) bytes of data.
64 bytes from 172.16.11.1: icmp_seq=1 ttl=64 time=0.030 ms
64 bytes from 172.16.11.1: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 172.16.11.1: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 172.16.11.1: icmp_seq=4 ttl=64 time=0.039 ms
^Z
[11]+  Stopped                  ping 172.16.11.1
student@student-H81H3-I:~$
```

Step 4: Insert Routing Table entries on each system to direct ipv4 packets

At Ha:

\$ sudo ip route add 172.16.11.0/24 via 172.16.10.2

\$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip route show
default via 192.168.8.1 dev eth1 proto static
169.254.0.0/16 dev eth1 scope link metric 1000
172.16.10.0/24 dev eth1 proto kernel scope link src 172.16.10.1
172.16.11.0/24 via 172.16.10.2 dev eth1
192.168.8.0/21 dev eth1 proto kernel scope link src 192.168.13.43 metric 1
student@pesit-To-be-filled-by-O-E-M:~$
```

At R1:

\$ sudo ip route add 172.16.11.0/24 via 172.16.10.3

\$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip route show
default via 192.168.8.1 dev eth1 proto static
169.254.0.0/16 dev eth1 scope link metric 1000
172.16.10.0/24 dev eth1 proto kernel scope link src 172.16.10.2
172.16.11.0/24 via 172.16.10.3 dev eth1
192.168.8.0/21 dev eth1 proto kernel scope link src 192.168.13.46 metric 1
student@pesit-To-be-filled-by-O-E-M:~$
```

At Hb:

```
$ sudo ip route add 172.16.10.0/24 via 172.16.11.1
```

```
$ ip route show
```

```
student@pesit-To-be-filled-by-O-E-M:~$ ip route show
default via 192.168.8.1 dev eth2 proto static
169.254.0.0/16 dev eth2 scope link metric 1000
172.16.10.0/24 via 172.16.11.1 dev eth1
172.16.11.0/24 dev eth1 proto kernel scope link src 172.16.11.2
192.168.8.0/21 dev eth1 proto kernel scope link src 192.168.13.44 metric 1
192.168.8.0/21 dev eth2 proto kernel scope link src 192.168.13.44 metric 1
student@pesit-To-be-filled-by-O-E-M:~$
```

Step 5: After adding routing table entries again verify the connection from Ha and Hb using ping command.

5.1 Open Wireshark in all the systems to capture packets and set capture filter.

```
$ sudo wireshark &
```

Note: Capture -> Options -> Host 172.16.11.2 and ICMP

5.2 Testing path from Ha and Hd

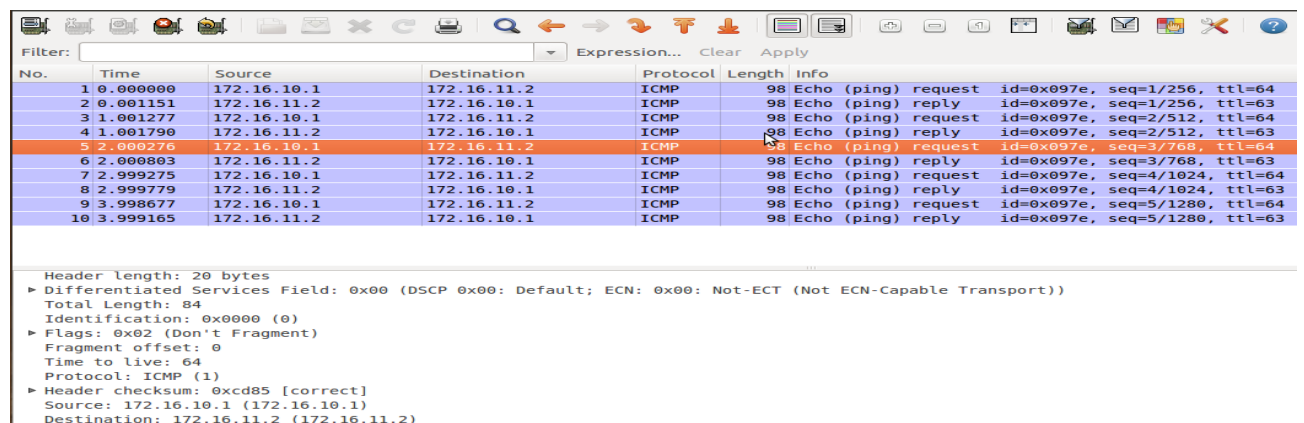
From Ha: \$ ping 172.16.11.2

```
student@pesit-To-be-filled-by-O-E-M:~$ ping -c 5 172.16.11.2
PING 172.16.11.2 (172.16.11.2) 56(84) bytes of data:
From 172.16.10.2: icmp_seq=1 Redirect Host(New nexthop: 172.16.10.3)
64 bytes from 172.16.11.2: icmp_req=1 ttl=63 time=1.17 ms
64 bytes from 172.16.11.2: icmp_req=2 ttl=63 time=0.530 ms
64 bytes from 172.16.11.2: icmp_req=3 ttl=63 time=0.541 ms
64 bytes from 172.16.11.2: icmp_req=4 ttl=63 time=0.517 ms
64 bytes from 172.16.11.2: icmp_req=5 ttl=63 time=0.504 ms

--- 172.16.11.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.504/0.652/1.172/0.261 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

5.3 Capture packets in all systems

Ha:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=1/256, ttl=64
2	0.001151	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=1/256, ttl=63
3	1.001277	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=2/512, ttl=64
4	1.001790	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=2/512, ttl=63
5	2.000295	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=3/768, ttl=64
6	2.000803	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=3/768, ttl=63
7	2.999275	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=4/1024, ttl=64
8	2.999779	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=4/1024, ttl=63
9	3.998677	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=5/1280, ttl=64
10	3.999165	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=5/1280, ttl=63

Header Length: 20 bytes	
► Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))	
Total Length: 84	
Identification: 0x0000 (0)	
► Flags: 0x02 (Don't Fragment)	
Fragment offset: 0	
Time to live: 64	
Protocol: ICMP (1)	
► Header checksum: 0xcd85 [correct]	
Source: 172.16.10.1 (172.16.10.1)	
Destination: 172.16.11.2 (172.16.11.2)	

R1:

Filter: icmp

No.	Time	Source	Destination	Protocol	Length	Info
4927	57.321314	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=1/256, ttl=64
4928	57.321352	172.16.10.2	172.16.10.1	ICMP	126	Redirect (Redirect for host)
4931	57.321706	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=1/256, ttl=63

Protocol: ICMP (1)
 ▶ Header checksum: 0x7685 [correct]
 Source: 172.16.10.2 (172.16.10.2)
 Destination: 172.16.10.1 (172.16.10.1)

Internet Control Message Protocol
 Type: 5 (Redirect)
 Code: 1 (Redirect for host)
 Checksum: 0x44eb [correct]
 Gateway address: 172.16.10.3 (172.16.10.3)

Internet Protocol Version 4, Src: 172.16.10.1 (172.16.10.1), Dst: 172.16.11.2 (172.16.11.2)
 Version: 4
 Header length: 20 bytes
 ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
 Total Length: 84
 Identification: 0x0000 (0)
 ▶ Flags: 0x02 (Don't Fragment)
 Fragment offset: 0
 Time to live: 63
 Protocol: ICMP (1)
 ▶ Header checksum: 0xc85 [correct]

R2:

Filter: icmp

No.	Time	Source	Destination	Protocol	Length	Info
5802	65.786311	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=1/256, ttl=63
5803	65.786563	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=1/256, ttl=63
5963	66.787040	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=2/512, ttl=64
5964	66.787279	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=2/512, ttl=63
6129	67.786065	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=3/768, ttl=64
6130	67.786296	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=3/768, ttl=63
6267	68.785030	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=4/1024, ttl=64
6268	68.785273	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=4/1024, ttl=63
6367	69.784449	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=5/1280, ttl=64
6368	69.784657	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=5/1280, ttl=63

Hb:

Filter: icmp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=1/256, ttl=64
2	0.000026	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=1/256, ttl=63
3	1.000739	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=2/512, ttl=63
4	1.000756	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=2/512, ttl=64
5	1.999782	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=3/768, ttl=63
6	1.999800	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=3/768, ttl=64
7	2.998764	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=4/1024, ttl=63
8	2.998780	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=4/1024, ttl=64
9	3.998180	172.16.10.1	172.16.11.2	ICMP	98	Echo (ping) request id=0x097e, seq=5/1280, ttl=63
10	3.998198	172.16.11.2	172.16.10.1	ICMP	98	Echo (ping) reply id=0x097e, seq=5/1280, ttl=64

Step 6: Check each system neighbors to verify the connection

ip neigh provides a command line interface to display the neighbor table (ARP cache)

Ha: \$ ip neigh show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
fe80::6d86:d60f:f1f7:e568 dev eth1 lladdr 24:ec:99:50:22:4d router STALE
192.168.8.1 dev eth1 lladdr 00:23:47:b8:8f:40 STALE
172.16.10.3 dev eth1 lladdr 50:e5:49:1b:f0:12 REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

In the above diagram, we can observe 172.16.10.3 (R2) as neighbor to 172.16.10.1 (H_A) because of ICMP Redirect happened from R1. New Gateway is selected i.e., 172.16.10.3 (R2)

R1: \$ ip neigh show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
fe80::6d86:d60f:f1f7:e568 dev eth1 lladdr 24:ec:99:50:22:4d router STALE
172.16.10.3 dev eth1 lladdr 50:e5:49:1b:f0:12 REACHABLE
192.168.8.1 dev eth1 lladdr 00:23:47:b8:8f:40 STALE
student@pesit-To-be-filled-by-O-E-M:~$
```

R2: \$ ip neigh show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
fe80::6d86:d60f:f1f7:e568 dev eth1 lladdr 24:ec:99:50:22:4d router STALE
192.168.8.1 dev eth1 lladdr 00:23:47:b8:8f:40 STALE
172.16.11.2 dev eth2 lladdr b8:a3:86:98:42:c9 REACHABLE
172.16.10.1 dev eth1 lladdr 50:e5:49:1b:f1:55 REACHABLE
172.16.10.2 dev eth1 lladdr 50:e5:49:1c:33:de REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

Hb: \$ ip neigh show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
fe80::6d86:d60f:f1f7:e568 dev eth2 lladdr 24:ec:99:50:22:4d router STALE
192.168.8.1 dev eth2 lladdr 00:23:47:b8:8f:40 STALE
172.16.11.1 dev eth1 lladdr fc:75:16:e1:23:76 REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

PORT Unreachable

We are trying to send data from System Ha and Hb using nc command. **nc** (or **netcat**) utility is used for just about anything under the sun involving TCP or UDP. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and deal with both IPv4 and IPv6.

At Hb (172.16.11.2):

\$ nc -l 1002

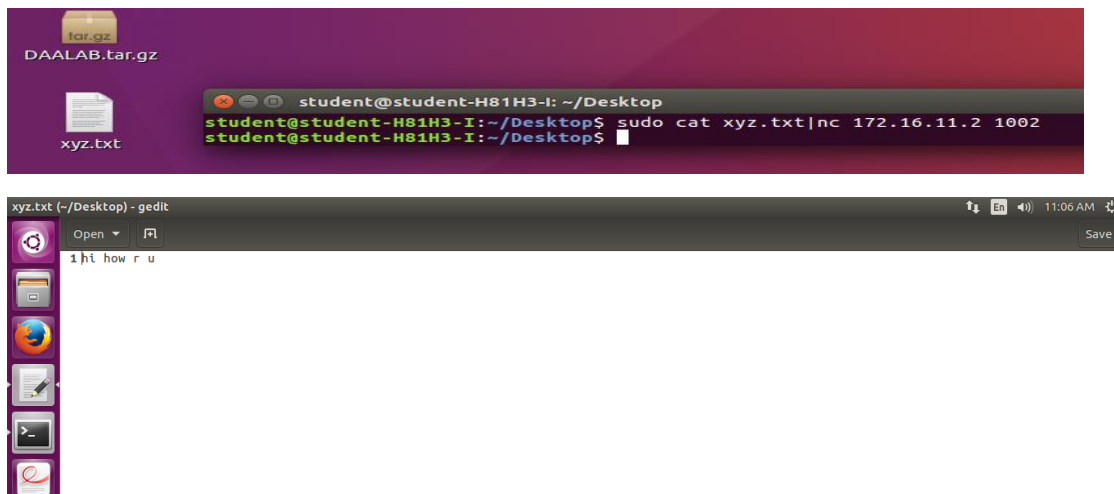
Here Hb system acts as server which is in listening mode through port 1002.

```
student@student-H81H3-I:~$ sudo nc -l 1002
hi how r u
student@student-H81H3-I:~$
```

At Ha (172.16.10.2):

\$ cat xyz.txt | nc 172.16.11.2 1002

Here Ha system acts as a client which is sending a file **xyz.txt** by identifying host **172.16.10.2** through port **1002**.



Note: If we give wrong port number which is not matching on both systems. Connection will fail and we get PORT UNREACHABLE error.

Conclusion:

1. Configuration of network interfaces.
2. Understanding how ICMP redirect message is sent for optimal routing.
3. More understanding of how TTL decreases with each router hop.