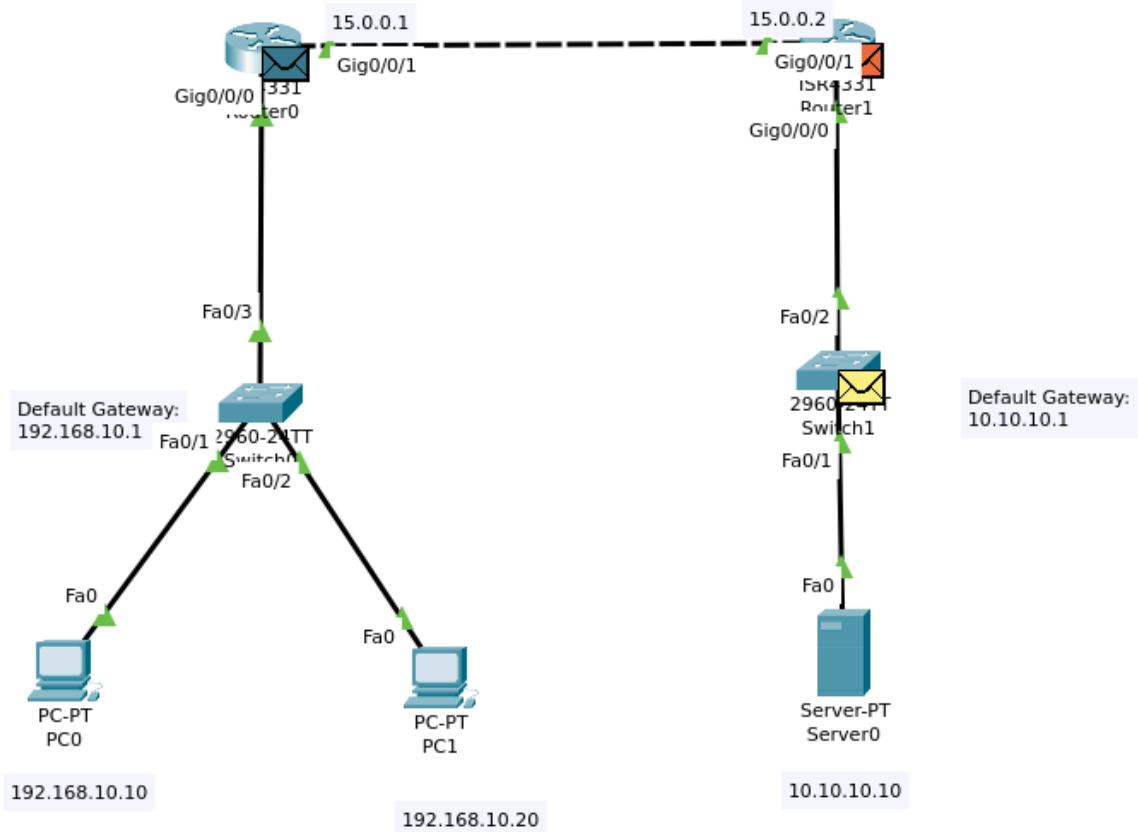


Name: J Kevin Immanuel
College: VIT Chennai

Q14)

Dynamic NAT:



Routing:

Router0

Static Routes

Network

10.0.0.0

Mask

255.0.0.0

Next Hop

15.0.0.2

Add

Network Address

10.0.0.0/8 via 15.0.0.2

Remove

Router 2:

CLI

Attributes

Static Routes

Network

192.168.10.0

Mask

255.255.255.0

Next Hop

15.0.0.1

Add

Network Address

192.168.10.0/24 via 15.0.0.1

Remove

ands

Creating a dynamic NAT:

```
Router(config)#
Router(config)#int g
Router(config)#int gigabitEthernet 0/0/1
Router(config-if)#ip nat outside
Router(config-if)#ex
Router(config)#int g
Router(config)#int gigabitEthernet 0/0/0
Router(config-if)#ip nat inside
Router(config-if)#ex
Router(config)#
Router(config)#ip nat pool DYNAT 15.0.0.5 15.0.0.20 netmask 255.255.255.0
Router(config)#access
Router(config)#access-list 1 permit 192.168.10.1 0.0.0.255
Router(config)#
Router(config)#
Router(config)#ip nat inside ?
    source    Source address translation
Router(config)#ip nat ?
    inside    Inside address translation
    outside   Outside address translation
    pool       Define pool of addresses
Router(config)#ip nat inside source ?
    list       Specify access list describing local addresses
    static     Specify static local->global mapping
Router(config)#ip nat inside source list 1 ?
    interface  Specify interface for global address
    pool       Name pool of global addresses
Router(config)#ip nat inside source list 1 DYNAT
^
% Invalid input detected at '^' marker.

Router(config)#ip nat inside source list 1 pool DYNAT
Router(config)#
Router(config)#ex
```

Packet Analysis:

Before leaving router:

PDU Information at Device: Router0

OSI Model [Inbound PDU Details](#) Outbound PDU Details

PDU Formats

EthernetII

0	4	8		Bytes
PREAMBLE: 101010..10		DEST ADDR:0002.160 E.E501		
SRC ADDR:00 03.E489.3763	TYP E:0x	DATA (VARIA BLE LENGTH)	FCS:0x00000 000	

IP

0	4	8	16	20	24	Bits
VER:4	IHL:5	DSCP:0x00	TL:28			
ID:0x0015			FLAG S:0x	<div><div></div><div></div><div></div></div>	FRAG OFFSET:0x000	
TTL:255		PRO:0x01	CHKSUM			
SRC IP:192.168.10.20						
DST IP:10.10.10.10						
DATA (VARIABLE LENGTH)						

ICMP

0	8	16	Bits
TYPE:0x08		CODE:0x00	CHECKSUM

After leaving Router:

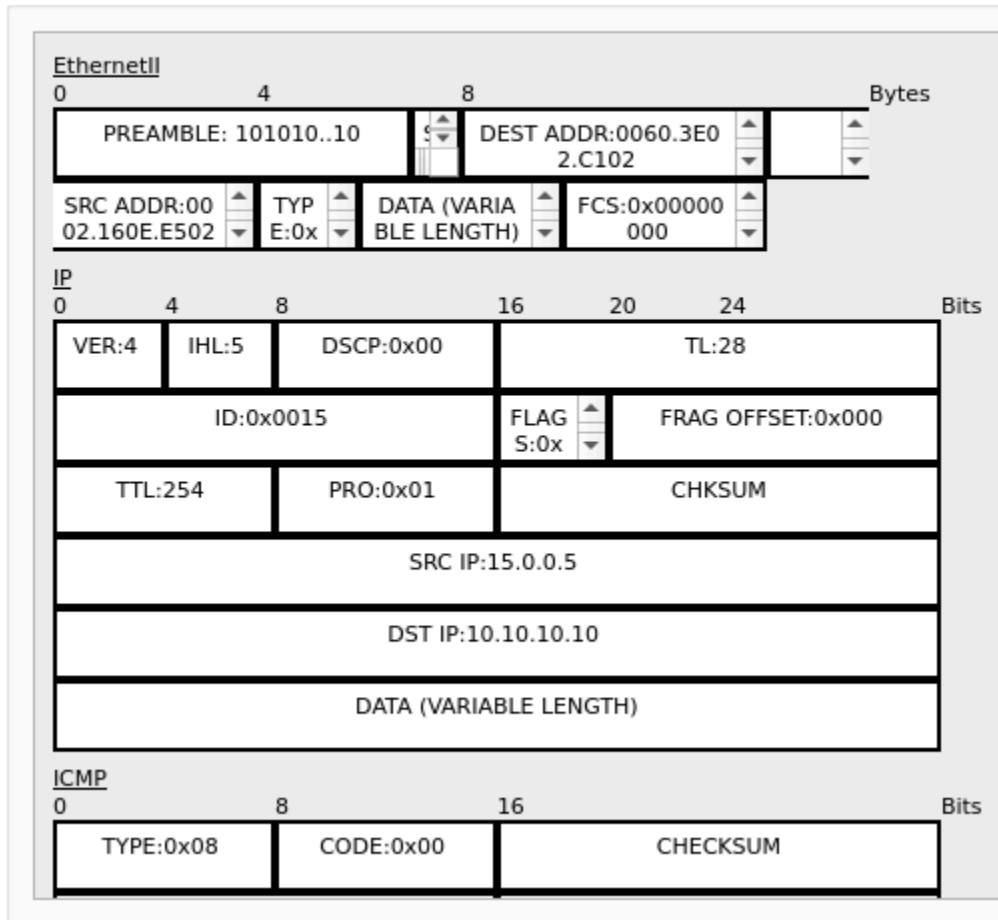
PDU Information at Device: Router0

OSI Model

Inbound PDU Details

Outbound PDU Details

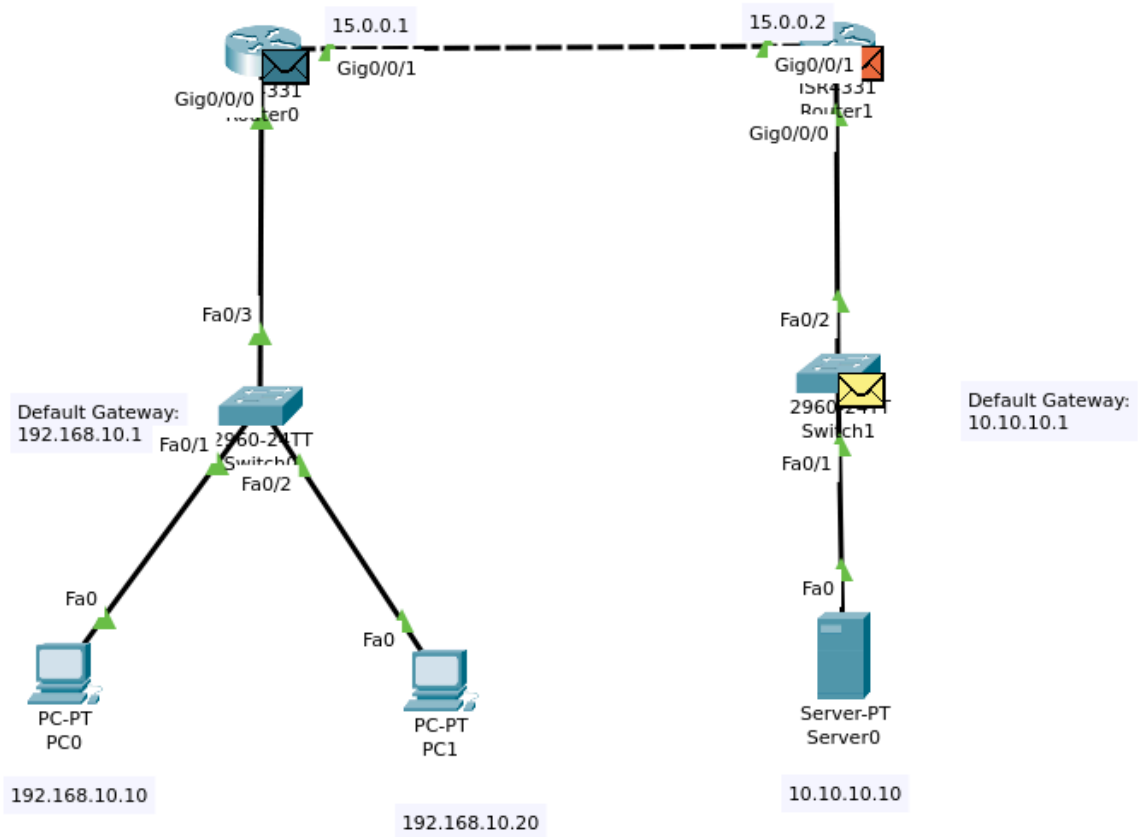
PDU Formats



As we can see, the source IP is different and has been changed to part of the NAT address pool we assigned.

PAT:

We will use the same network as above:



Every setting is same as the above.

Assigning ACL for NAT:

```
Router(config)#int gig0/0/0
Router(config-if)#ip nat inside
Router(config-if)#ex
Router(config)#int gig0/0/1
Router(config-if)#ip nat outside
Router(config-if)#ex
Router(config)#
Router(config)#
Router(config)#access-
Router(config)#access-list standar
Router(config)#access-list 1 permit 192.168.10.0 0.0.0.255
```

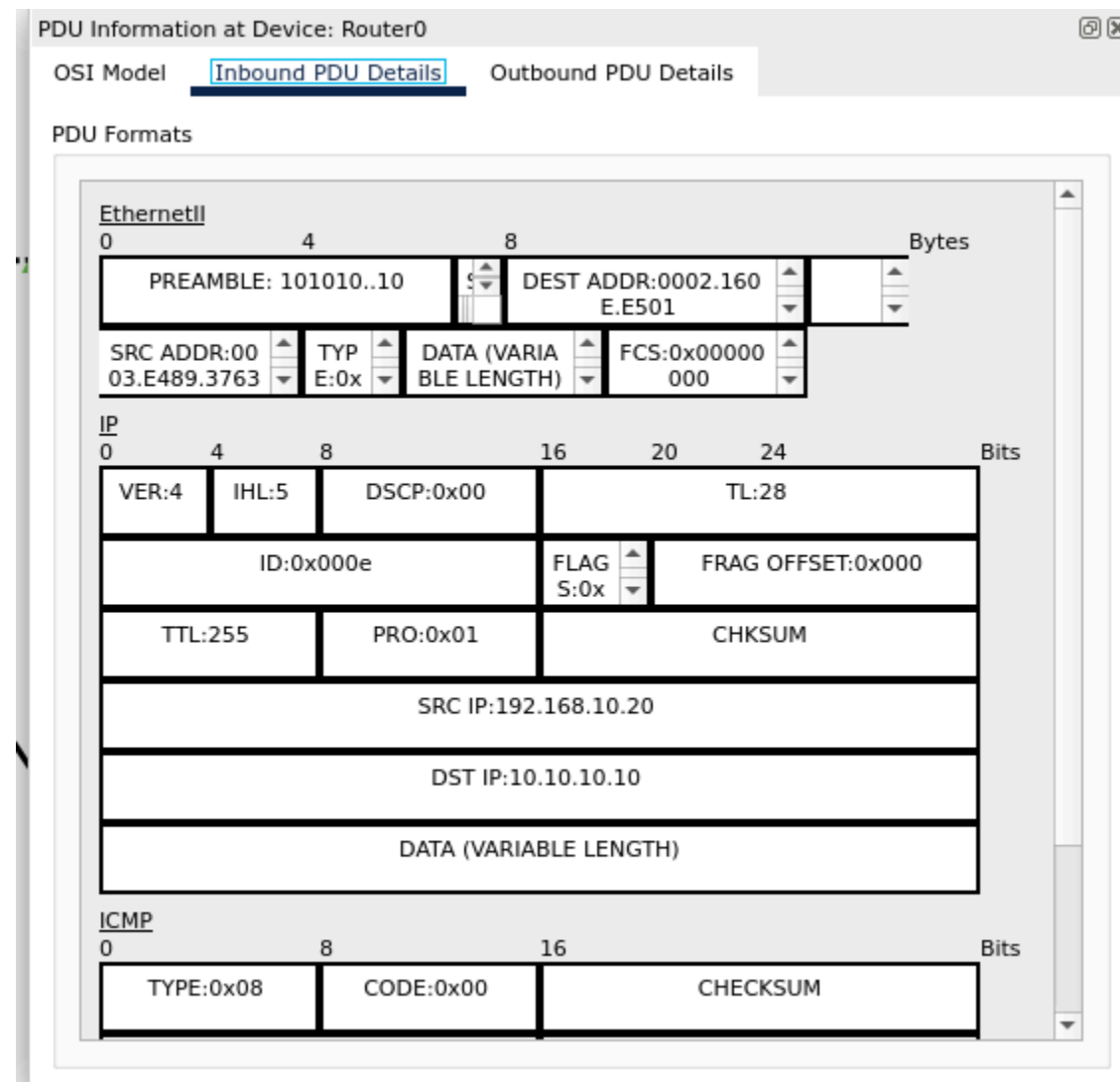
NAT Overloading:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip nat inside source list 1 interface gigabitEthernet 0/0/0 overload
Router(config)#ip nat inside source list 1 interface gig
Router(config)#ip nat inside source list 1 interface gigabitEthernet 0/0/1 overload
Router(config)#ex
Router#
%SYS-5-CONFIG_I: Configured from console by console

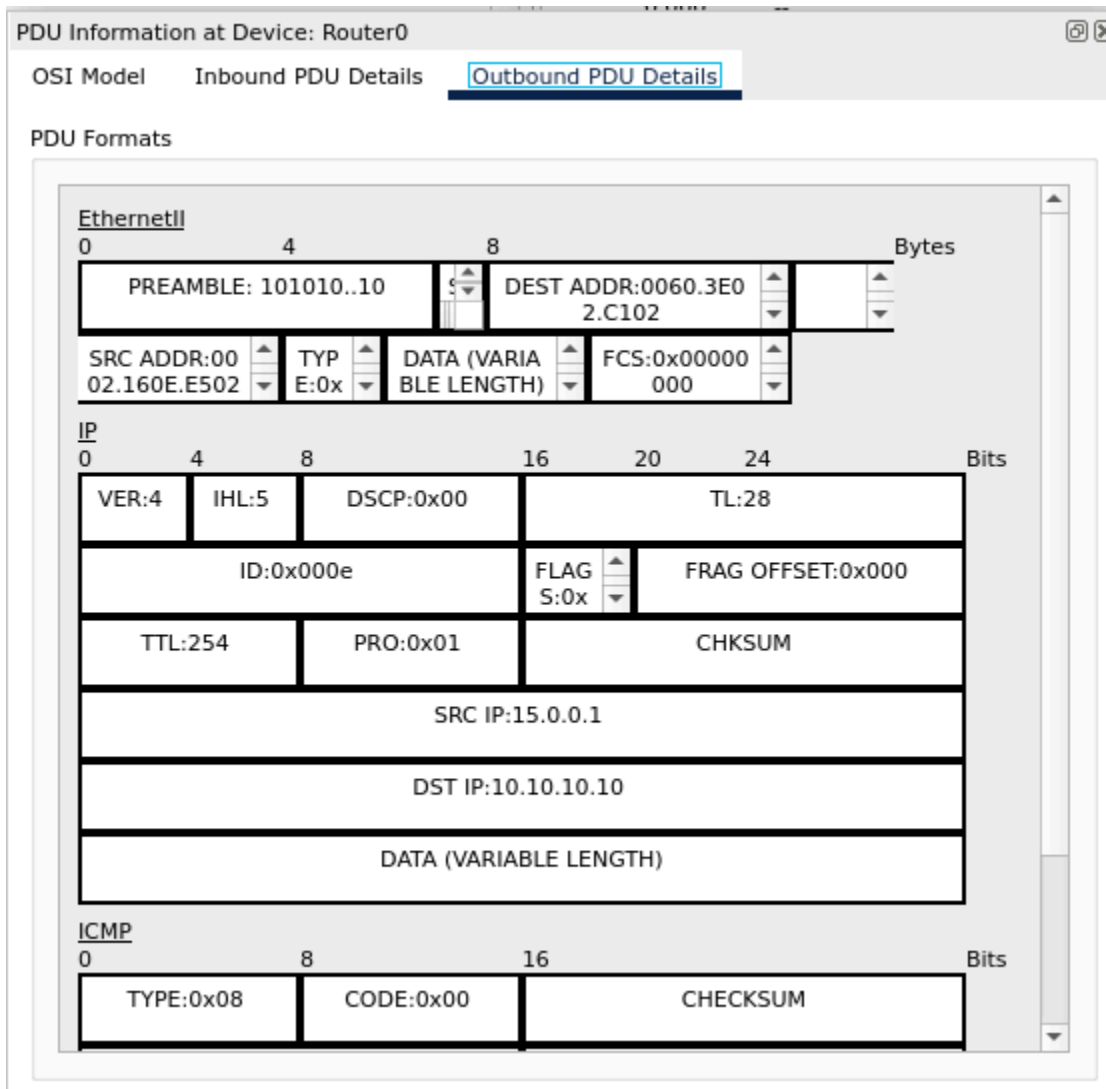
Router#
Router#sh ip nat translations
```


Packet Sample:

Inbound:



Outbound:



As we can see, the IP has been converted by the NAT

Q15) IPERF:

We will be using iperf on both Vm1 (Ubuntu) and VM2 (Linux Mint):

iperf -s is the command to run as the server

iperf -c 192.168.0.3 is the command to run as the client. The ip mentioned is the ip of the server.

Client

```
kevin@pop-os:~$ iperf -c 192.168.0.3
-----
Client connecting to 192.168.0.3, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  1] local 192.168.0.117 port 59060 connected with 192.168.0.3 port 5001
[ ID] Interval           Transfer     Bandwidth
[  1] 0.0000-10.0219 sec   4.20 GBytes  3.60 Gbits/sec
kevin@pop-os:~$
```

Server

```
mint-vm@mintvm-VirtualBox: ~
mint-vm@mintvm-VirtualBox:~$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[  1] local 192.168.0.3 port 5001 connected with 192.168.0.117 port 59060 (icwnd
/mss/irrtt=14/1448/107)
[ ID] Interval           Transfer     Bandwidth
[  1] 0.0000-10.3495 sec   4.20 GBytes  3.49 Gbits/sec
^Cmint-vm@mintvm-VirtualBox:~$
```

By default, TCP is being tested. So the above is using TCP

UDP:

Server:

```
mint-vm@mintvm-VirtualBox:~$ iperf -s -u
-----
Server listening on UDP port 5001
UDP buffer size: 208 KByte (default)
-----
[ 1] local 192.168.0.3 port 5001 connected with 192.168.0.117 port 36606
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 1] 0.0000-10.3596 sec 1.25 MBytes 1.02 Mbits/sec 0.383 ms 0/895 (0%)
^Cmint-vm@mintvm-VirtualBox:~$
```

Client:

```
kevin@pop-os:~$ iperf -c 192.168.0.3 -u
-----
Client connecting to 192.168.0.3, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 1] local 192.168.0.117 port 36606 connected with 192.168.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.0153 sec 1.25 MBytes 1.05 Mbits/sec
[ 1] Sent 896 datagrams
[ 1] Server Report:
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 1] 0.0000-10.3596 sec 1.25 MBytes 1.02 Mbits/sec 0.383 ms 0/895 (0%)
kevin@pop-os:~$
```

Reverse:

Server:

```
mint-vm@mintvm-VirtualBox: ~  
mint-vm@mintvm-VirtualBox:~$ iperf -s -r  
WARNING: option -r is not valid for server mode  
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (default)  
-----  
[ 1] local 192.168.0.3 port 5001 connected with 192.168.0.113 port 38180 (icwnd/mss/irrt=14/1448/85758)  
[ ID] Interval      Transfer      Bandwidth  
[ 1] 0.0000-10.3220 sec 25.8 MBytes  20.9 Mbits/sec  
-----  
Client connecting to 192.168.0.113, TCP port 5001  
TCP window size: 16.0 KByte (default)  
-----  
[ *2] local 192.168.0.3 port 32846 connected with 192.168.0.113 port 5001 (reverse) (icwnd/mss/irrt=14/1448/2779)  
[ ID] Interval      Transfer      Bandwidth  
[ *2] 0.0000-10.7768 sec 27.5 MBytes  21.4 Mbits/sec
```

Client:

```
asd@pop-os: ~  
asd@pop-os:~$ iperf -c 192.168.0.3 -r  
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (default)  
-----  
Client connecting to 192.168.0.3, TCP port 5001  
TCP window size: 85.0 KByte (default)  
-----  
[ 1] local 192.168.0.113 port 38180 connected with 192.168.0.3 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 1] 0.0000-10.3376 sec 25.8 MBytes  20.9 Mbits/sec  
[ 2] local 192.168.0.113 port 5001 connected with 192.168.0.3 port 32846  
[ ID] Interval      Transfer      Bandwidth  
[ 2] 0.0000-10.7004 sec 27.5 MBytes  21.6 Mbits/sec  
asd@pop-os:~$
```

Note: For reversal to work, firewall has to be disabled. Hence using both in VMs.

Bidirectional:

```
asd@pop-os:~$ iperf -c 192.168.0.3 -d
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
Client connecting to 192.168.0.3, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[ 1] local 192.168.0.113 port 43206 connected with 192.168.0.3 port 5001
[ 2] local 192.168.0.113 port 5001 connected with 192.168.0.3 port 44932
[ ID] Interval      Transfer    Bandwidth
[ 2] 0.0000-10.8347 sec 19.6 MBytes 15.2 Mbits/sec
[ 1] 0.0000-11.1060 sec 9.63 MBytes 7.27 Mbits/sec
asd@pop-os:~$
```

We get two outputs instead of one, as this is bidirectional