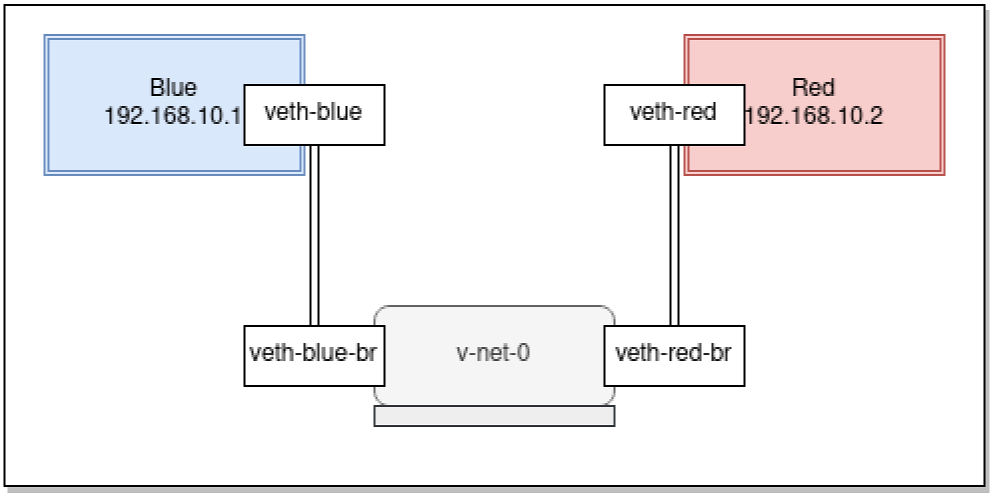
CS751: Network Engineering
Dept. of Computer Science and Engineering
NITK Surathkal

Lab task 4: Using Linux Bridge in netns

Subtask 1: Namespace talking to the outside world using the Linux bridge.



Step 1: Create two network namespaces

```
ip netns add blue
ip netns add red
```

Step 2: Show available network namespaces

```
ip netns show
```

Step 3: Create a bridge interface named v-net-0

```
ip link add v-net-0 type bridge
```

Step 4: Bring up the bridge interface

```
ip link set dev v-net-0 up
```

Step 5: Create two pairs of virtual Ethernet interfaces (veth pairs)

```
ip link add veth-blue type veth peer name veth-blue-br
ip link add veth-red type veth peer name veth-red-br
```

Step 6: Move one end of each veth pair to its respective namespace

```
ip link set veth-blue netns blue
ip link set veth-red netns red
```

Step 7: Attach the veth-blue-br to the bridge v-net-0

```
ip link set veth-blue-br master v-net-0
```

Step 8: Attach the veth-red-br to the bridge v-net-0

```
ip link set veth-red-br master v-net-0
```

Step 9: Assign IP addresses to the interfaces in their respective namespaces

```
ip -n blue addr add 192.168.10.1/24 dev veth-blue
ip -n red addr add 192.168.10.2/24 dev veth-red
```

Step 10: Bring up the interfaces in their respective namespaces

```
ip -n blue link set veth-blue up
ip -n red link set veth-red up
```

Step 11: Bring up the bridge

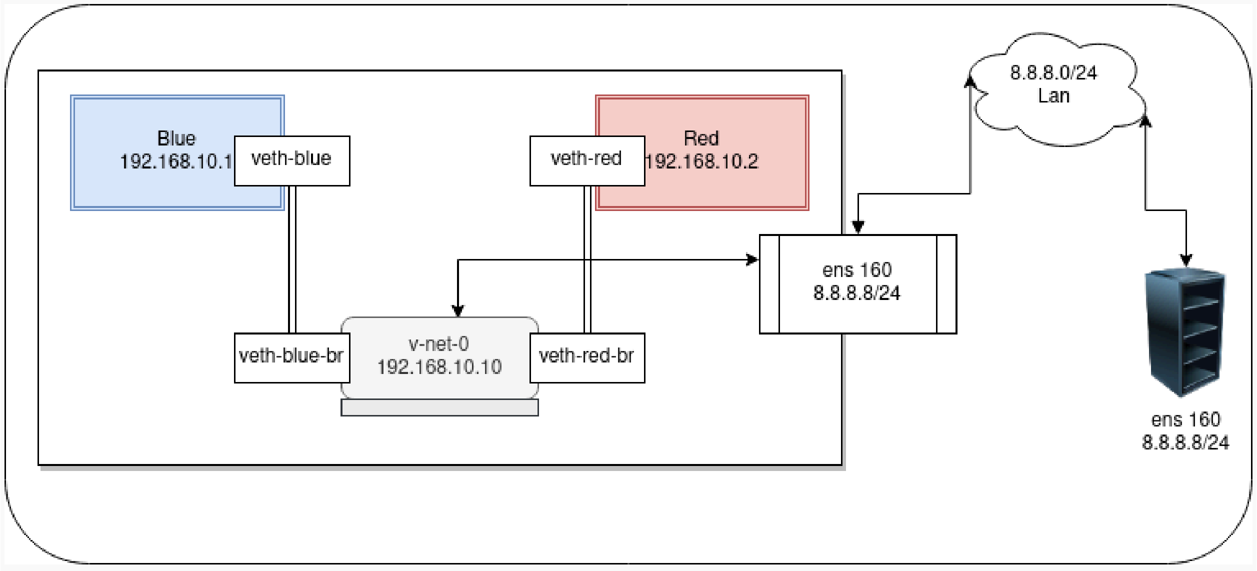
```
ip link set veth-blue-br up
ip link set veth-red-br up
```

Ping the other Google DNS(8.8.8.8) at this stage from the blue namespace.

```
ip netns exec blue ping 8.8.8.8
```

Output: `ping: connect: Network is unreachable`

This is expected since the namespace does not have any routes configured to send the packet out to 8.8.8.8/24



For this, we need to add a route to the namespace . However, to do that we first need to **set a gateway for the subnet** for which the namespace is a part (192.168.10.0/24). This can be set on the **Linux bridge** with the termination of the patch cables from both the namespaces (red & blue) in this example, here (ip addr add 192.168.10.10/24 **dev** v-net-0) **dev** mining bridge device.

Set the gateway on the linux bridge:

```
ip addr add 192.168.10.10/24 dev v-net-0
```

Add the route to namespace Blue

```
ip -n blue route add 8.8.8.0/24 via 192.168.10.10
```

Check the route table

```
ip netns exec blue route -n
```

Ping 8.8.8.8 from blue namespace

```
ip netns exec blue ping 8.8.8.8
```

So now we see that even after adding a route the ping fails, but the output is different from the previous failure. In this case, the namespace is able to send the packet to the correct network via “veth-blue” however something fails after that. At this stage, the packet leaves the namespace and arrives at the Linux bridge. The Linux bridge now acts as the gateway between the network namespaces.

We'll need to enable NAT here to resolve the above error.

In this case, the namespace network 192.168.10.0/24 needs to be translated to 8.8.8.0/24 (the VM infrastructure network). As the routers running outside the VM are not aware of the namespace network (192.168.10.0/24) hence will drop all traffic arriving from those networks eventually. NAT functions by translating container-private IP addresses to the host's public IP during outbound communication.

Set up the NAT rule in the NAT table

```
iptables --table nat -A POSTROUTING -s 192.168.10.0/24 -j MASQUERADE
```

Enable IPV4 forwarding

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Now will again try to ping the google dns server and it will work perfectly

```
ip netns exec blue ping 8.8.8.8 -c4
```

NOTE: On completion of Subtask 1, do not delete the created topology.

SubTask 2: Setting Up FTP Server in Network Namespace.

File Transfer Protocol (FTP) is a standard network protocol used for the transfer of files between a client and a server on a computer network. In technical terms, FTP operates on a client-server model where a client initiates a connection to a server to perform file transfers.

FTP utilizes two separate channels to facilitate communication: the command channel (often operating over Transmission Control Protocol or TCP) and the data channel. The command channel is responsible for sending commands from the client to the server, such as requests for file transfers or directory listings. Meanwhile, the data channel is utilized to actually transfer the files themselves.

FTP can operate in either anonymous or authenticated modes. In anonymous mode, users can log in with a generic username (such as "anonymous" or "ftp") and typically use their email address as the password. Authenticated mode requires users to provide valid credentials to access the FTP server.

Step 1: Setup Directories

```
$ mkdir /home/{USER}/server
$ touch /home/{USER}/server/file1.txt
$ touch /home/{USER}/server/file2.txt
```

We create a directory, server, to simulate. Two files, file1.txt and file2.txt, are created in the server directory.

Step 2: Add Dummy Text to Files

You can use a text editor to add some dummy text to file1.txt and file2.txt.

Step 3: Install FTP Server

```
$ sudo apt install vsftpd
```

Step 4: Check Server Status

```
$ sudo systemctl status vsftpd
```

Step 5: Take Backup of vsftpd.service config file

Note: Before editing a config file, the best practice is to make a copy beforehand in case something goes wrong.

```
$ sudo cp /lib/systemd/system/vsftpd.service
/lib/systemd/system/vsftpd.service.copy
```

Step 6: Add Network Namespace Path to Service File

Since vsftpd runs as a service we cannot just start it up in a namespace the way that we normally do. Instead we need to configure the service such that it knows which namespace it needs to run in.

```
$ sudo nano /lib/systemd/system/vsftpd.service
```

Add/Edit the following line under [Service]:

```
“NetworkNamespacePath=/var/run/netns/blue”
```

Edit the vsftpd service file to include the network namespace path.

```
$ nano /etc/vsftpd.conf
```

Add the following lines:

```
“local_root=/home/anmol/server
chroot_local_user=YES
allow_writeable_chroot=YES
write_enable=YES
”
```

Step 6: Restart the Service

```
$ systemctl daemon-reload
```

```
$ systemctl restart vsftpd
```

Restart the vsftpd service to apply the changes made to the service file.

Step 7: Check the Status Again

```
$ systemctl status vsftpd
```

Step 8: Log in to the Server from the Red Namespace

```
$ ip netns exec red ftp 192.168.10.1
```

Step 9: Perform various FTP commands within the FTP session

```
$ ls
```

List files on the FTP server.

```
$ get file1.txt /home/{USER}/Desktop/file1.txt
```

Download a file from the server to the client.

```
$ put /home/{USER}/Desktop/file2.txt file2.txt
```

Upload a file from the client to the server.

```
$ mkdir dummy
```

Create a directory on the server.

```
$ rmdir dummy
```

Remove a directory on the server.

```
$ delete file2.txt
```

Delete a file on the server.

```
$ exit
```

Logout of the FTP session.