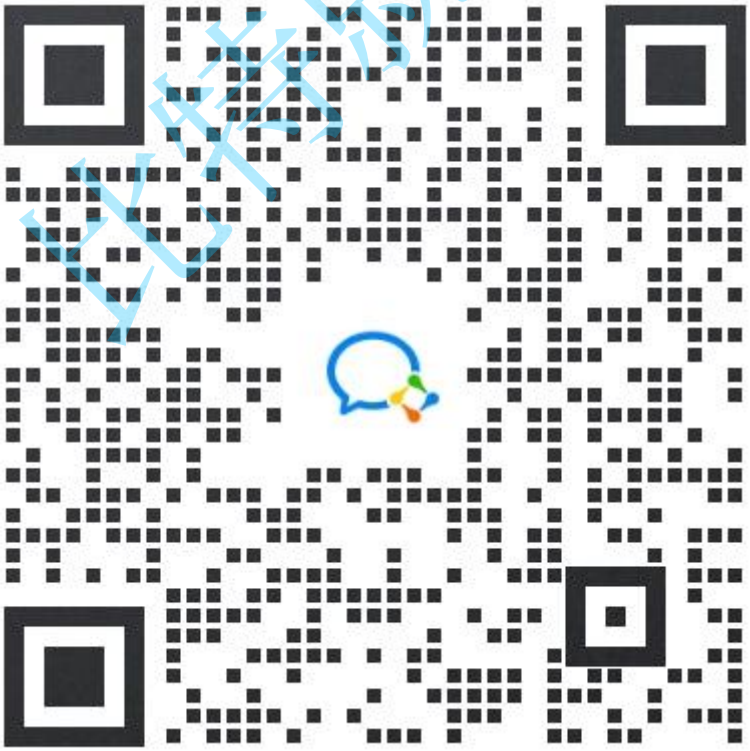
# veth 实战

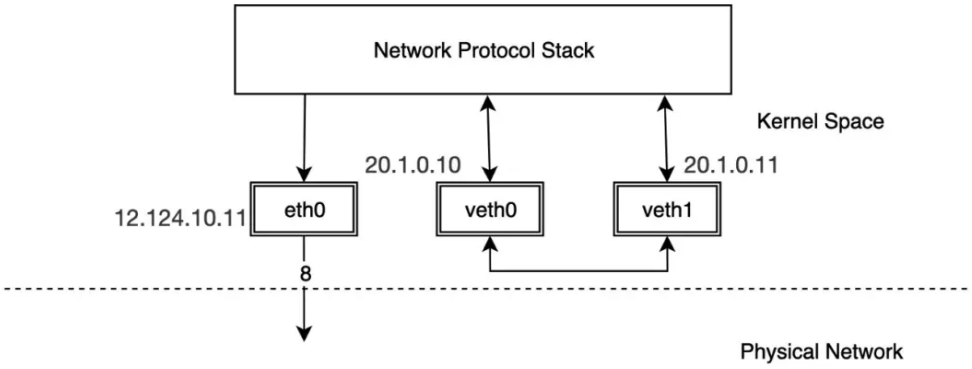
## 版权说明





## 基础知识

veth pair 是成对出现的一种虚拟网络设备接口，一端连着网络协议栈，一端彼此相连。




由于它的这个特性，常常被用于构建虚拟网络拓扑。例如连接两个不同的网络命名空间(netns)，连接 docker 容器，连接网桥(Bridge)等.

**常见命令**

1. veth 操作

```
Shell
# 添加 veth
ip link add <veth name> type veth peer name <peer name>
# 删除 veth
ip link delete  <veth name>
# 查看 veth
ip link show
```

2. 命名空间操作

```
Shell
#添加 ns
ip netns add <name>
#删除 ns
ip netns del  <name>
#执行命令
ip netns exec  <name> <cmd>
#遍历 ns
ip netns list
```

# 实战目的

了解 veth 虚拟网卡对的创建

# 实战步骤

1. 查看当前网卡信息

```
Shell
root@139-159-150-152:~# ifconfig
br-df863876204e: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.0.1  netmask 255.255.0.0  broadcast
192.168.255.255
        inet6 fe80::42:20ff:feb7:55bb  prefixlen 64  scopeid
0x20<link>
        ether 02:42:20:b7:55:bb  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29  bytes 4216 (4.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast
172.17.255.255
        inet6 fe80::42:d7ff:fe87:d11b  prefixlen 64  scopeid
0x20<link>
        ether 02:42:d7:87:d1:1b  txqueuelen 0  (Ethernet)
        RX packets 586466  bytes 36737314 (36.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 860320  bytes 1655179375 (1.6 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.183  netmask 255.255.255.0  broadcast
192.168.0.255
        inet6 fe80::f816:3eff:fe9d:f4ac  prefixlen 64  scopeid
0x20<link>
        ether fa:16:3e:9d:f4:ac  txqueuelen 1000  (Ethernet)
        RX packets 4032792  bytes 4360076332 (4.3 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2643585  bytes 470587289 (470.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1126057  bytes 449407907 (449.4 MB)
```

```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1126057  bytes 449407907 (449.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lxcbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.0.3.1  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::216:3eff:fe00:0  prefixlen 64  scopeid
0x20<link>
        ether 00:16:3e:00:00:00  txqueuelen 1000  (Ethernet)
        RX packets 595  bytes 72213 (72.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 817  bytes 73558 (73.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. 我们创建两个网络空间

```
Shell
root@139-159-150-152:~# ip netns add ns1
root@139-159-150-152:~# ip netns add ns2
root@139-159-150-152:~# ip netns list
ns2
ns1
```

3. 创建一个虚拟网卡对

```
Shell
# 创建一对 veth
ip link add veth11 type veth peer name veth12
```

4. 执行 ifconfig -a 可以看到网卡多了 2 个

```
Shell
root@139-159-150-152:~# ifconfig -a
br-df863876204e: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.0.1  netmask 255.255.0.0  broadcast
192.168.255.255
        inet6 fe80::42:20ff:feb7:55bb  prefixlen 64  scopeid
0x20<link>
        ether 02:42:20:b7:55:bb  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
        TX packets 29  bytes 4216 (4.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast
172.17.255.255
        inet6 fe80::42:d7ff:fe87:d11b  prefixlen 64  scopeid
0x20<link>
        ether 02:42:d7:87:d1:1b  txqueuelen 0  (Ethernet)
        RX packets 586466  bytes 36737314 (36.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 860320  bytes 1655179375 (1.6 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.183  netmask 255.255.255.0  broadcast
192.168.0.255
        inet6 fe80::f816:3eff:fe9d:f4ac  prefixlen 64  scopeid
0x20<link>
        ether fa:16:3e:9d:f4:ac  txqueuelen 1000  (Ethernet)
        RX packets 4033449  bytes 4360150167 (4.3 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2644071  bytes 470646936 (470.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1126165  bytes 449417393 (449.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1126165  bytes 449417393 (449.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lxcbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.0.3.1  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::216:3eff:fe00:0  prefixlen 64  scopeid
0x20<link>
        ether 00:16:3e:00:00:00  txqueuelen 1000  (Ethernet)
        RX packets 595  bytes 72213 (72.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 817  bytes 73558 (73.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
veth11: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 86:f6:83:2e:31:61  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

veth12: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 5e:b9:7a:5d:f2:b8  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. 我们将网卡挪到不同的命名空间中

```
Shell
root@139-159-150-152:~# ip link set veth11 netns ns1
root@139-159-150-152:~# ip link set veth12 netns ns2
```

6. 激活我们的网卡

```
Shell
root@139-159-150-152:~# ip netns exec ns1 ip link set veth11 up
root@139-159-150-152:~# ip netns exec ns2 ip link set veth12 up
```

7. 给网卡分配 IP 地址

```
Shell
ip netns exec ns1 ip addr add 10.5.0.1/24 dev veth11
ip netns exec ns2 ip addr add 10.5.0.2/24 dev veth12
```

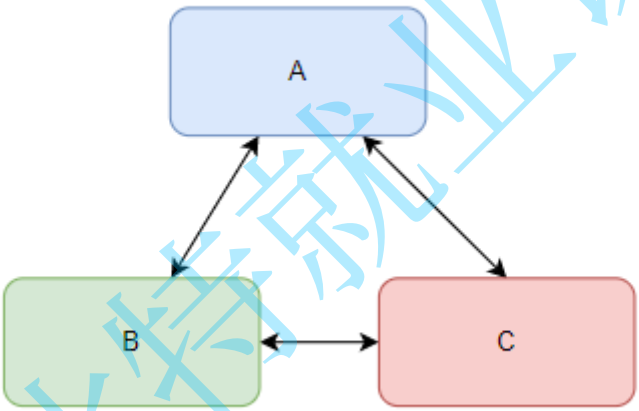
8. 然后我们在各自的命名空间中 ping 对方，可以看到我们的网卡是能通的

```
Shell
root@139-159-150-152:~# ip netns exec ns1 ping 10.5.0.2
PING 10.5.0.2 (10.5.0.2) 56(84) bytes of data.
64 bytes from 10.5.0.2: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 10.5.0.2: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 10.5.0.2: icmp_seq=3 ttl=64 time=0.033 ms
^C
--- 10.5.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
```

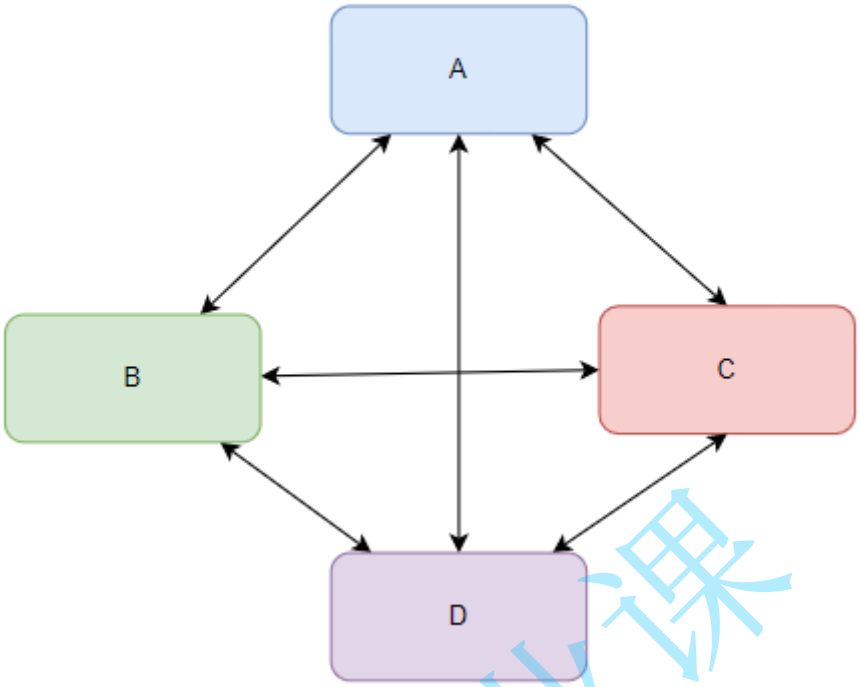
```
rtt min/avg/max/mdev = 0.031/0.032/0.033/0.000 ms
root@139-159-150-152:~# ip netns exec ns2 ping 10.5.0.1
PING 10.5.0.1 (10.5.0.1) 56(84) bytes of data.
64 bytes from 10.5.0.1: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 10.5.0.1: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 10.5.0.1: icmp_seq=3 ttl=64 time=0.037 ms
^X64 bytes from 10.5.0.1: icmp_seq=4 ttl=64 time=0.042 ms
^C
--- 10.5.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.023/0.033/0.042/0.007 ms
```

可以看到我们使用 veth pair 将两个隔离的 netns 成功的连接到了一起。

但是这样的网络拓扑存在一个弊端，随着网络设备的增多，网络连线的复杂度将成倍增长。 如果连接三个 netns 时，网络连线就成了下图的样子




而如果连接四个 netns 时，网络连线就成了下图的样子




如果有五台设备。。。

有没有什么技术可以解决这个问题呢？答案是有的，Linux Bridge（网桥）