

# Linux Containers

Tony Dal Santo

RIT CSCI 351

# What are Linux Containers

- Also known as Namespaces
- The clone(2) system call can create multiple instances of the network stack, file systems, process space, names, etc
- These instances operate independently, and a process can be bound to an alternate namespace instead of the default instance
- See also the man pages namespaces(7), unshare(2), and setns(2)

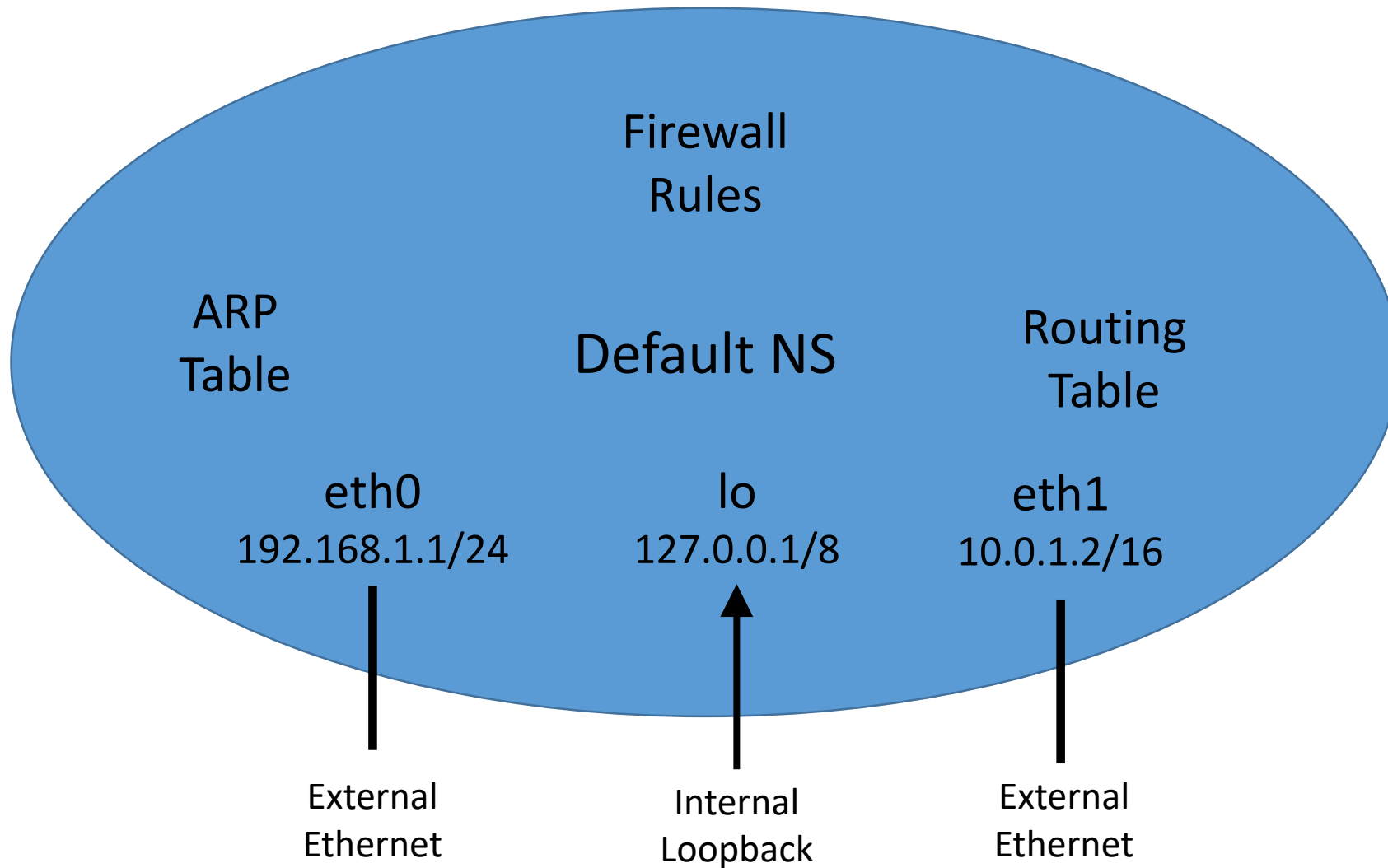
# Name Space IDs & Names

- Namespaces are created with the clone or unshare system calls
- That call creates a new process (and new process ID)
- That ID becomes the handle for that namespace
- It is possible to give names to namespaces
- The command “ip netns add <ns>” creates a name, but no actual namespace
- The command “ip netns exec <ns> <cmd> <args>” executes a command inside of the specified namespace
- The namespace lives until all processes exit, and the name is removed

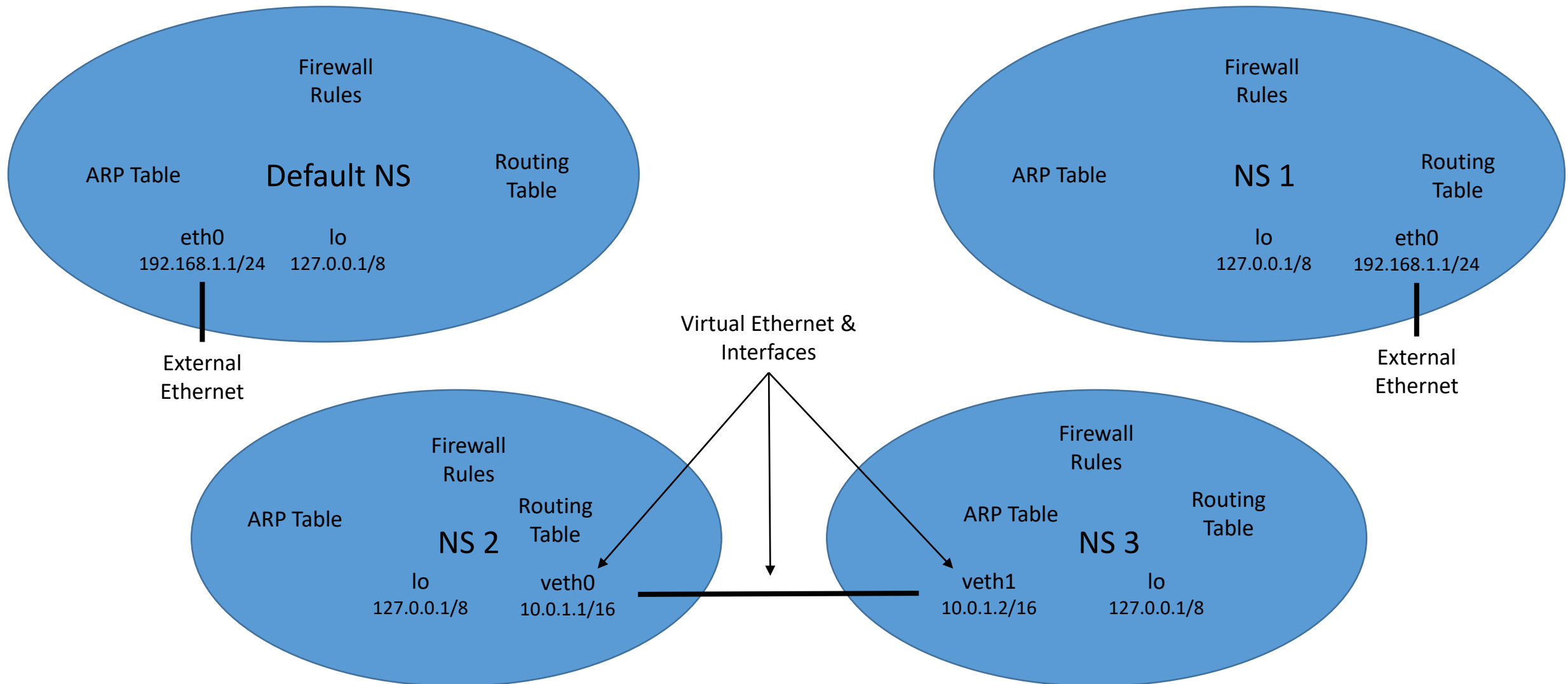
# What Resides in a Network Namespace?

- Each Network Namespace has its own IP stack instance, networking tables (routing, ARP, addresses), unique 5-Tuple, firewalls, etc
- Each physical and virtual interface resides in only one namespace
- A virtual Ethernet (veth) has two interfaces, defaulting to the same namespace, then typically moving one to a different namespace
- Data into one interface of a veth pair exits the other, and thus are useful for connecting namespaces
- A virtual bridge can connect multiple real and virtual Ethernets

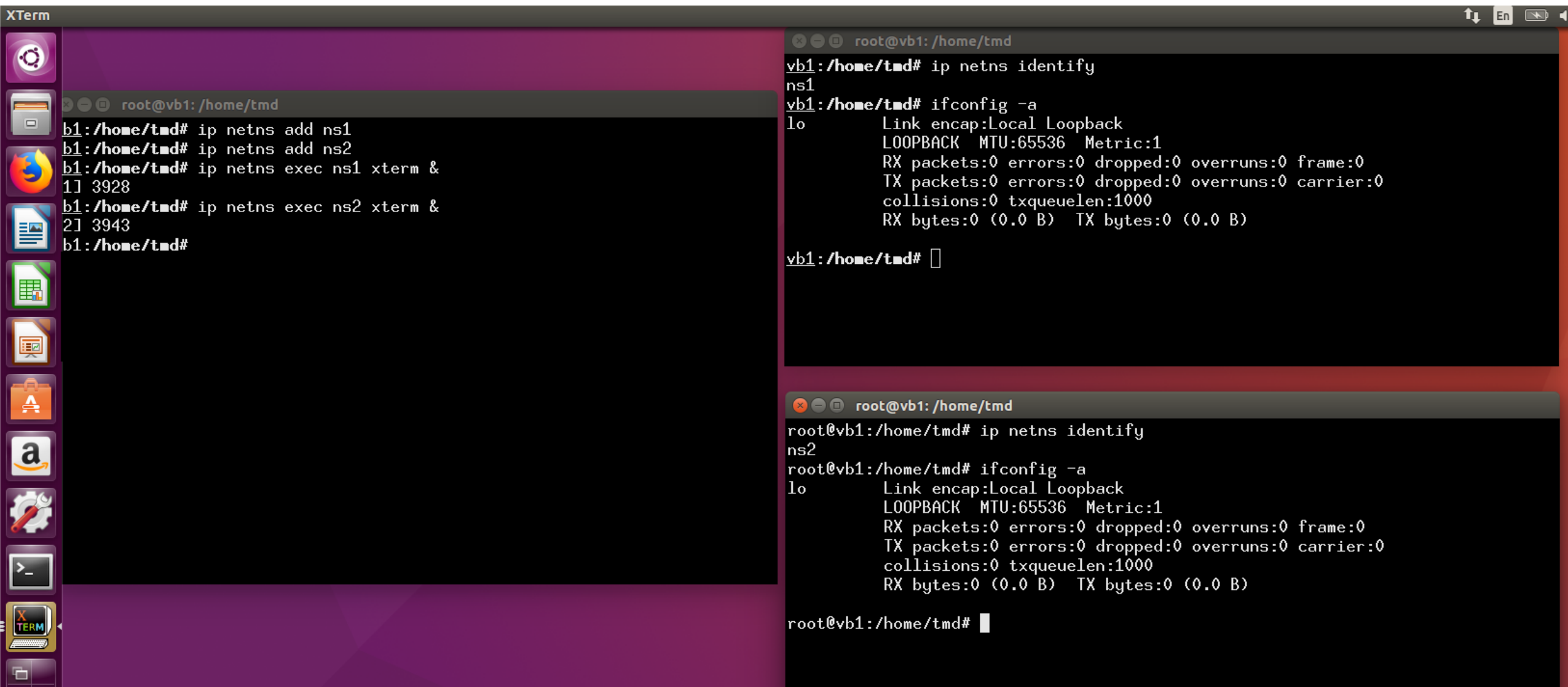
# Typical Default Network Namespace



# Multiple Simultaneously Active Namespaces



# Creating two namespaces



```
XTerm
root@vb1: /home/tmd
b1:/home/tmd# ip netns add ns1
b1:/home/tmd# ip netns add ns2
b1:/home/tmd# ip netns exec ns1 xterm &
1] 3928
b1:/home/tmd# ip netns exec ns2 xterm &
2] 3943
b1:/home/tmd#

root@vb1: /home/tmd
vb1:/home/tmd# ip netns identify
ns1
vb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            LOOPBACK MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vb1:/home/tmd#

root@vb1: /home/tmd
root@vb1:/home/tmd# ip netns identify
ns2
root@vb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            LOOPBACK MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@vb1:/home/tmd#
```

# Two Namespaces – What To Notice

- Launched an xterm in each namespace
  - The “ip netns identify” command shows the NS of the process
  - The “ifconfig” command shows network interfaces
  - Each new NS only contains a loopback
  - The loopback interface is down!
- 
- Next step, create a virtual Ethernet in ns1



```
root@vb1: /home/tmd
```

```
b1:/home/tmd# ip netns add ns1
b1:/home/tmd# ip netns add ns2
b1:/home/tmd# ip netns exec ns1 xterm &
1] 3928
b1:/home/tmd# ip netns exec ns2 xterm &
2] 3943
b1:/home/tmd#
```

```
root@vb1: /home/tmd
```

```
vb1:/home/tmd# ip link add veth1 type veth peer name veth2
vb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            LOOPBACK MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth1       Link encap:Ethernet  HWaddr 0e:1c:58:ad:d7:aa
            BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth2       Link encap:Ethernet  HWaddr ba:80:b4:ba:c2:0d
            BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
vb1:/home/tmd#
```

```
root@vb1: /home/tmd
```

```
root@vb1: /home/tmd#
```

# Virtual Ethernet

- The command to create a virtual Ethernet:  
`ip link add <name 1> type veth peer name <name 2>`
- Two interfaces appear in the same namespace
- The interfaces have no configuration, and are down
- Next step, move one interface to another ns, and configure them
- The move command  
`ip link set <interface> netns <namespace>`
- The IP address configuration command  
`ifconfig <interface> <ip address>/<mask length>`

root@vb1: /home/tmd

vb1:/home/tmd# ip link set veth2 netns ns2

vb1:/home/tmd# ifconfig -a

```
lo          Link encap:Local Loopback
            LOOPBACK MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
veth1       Link encap:Ethernet HWaddr 0e:1c:58:ad:d7:aa
            BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

vb1:/home/tmd#

root@vb1: /home/tmd

root@vb1:/home/tmd# ifconfig -a

```
lo          Link encap:Local Loopback
            LOOPBACK MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
veth2       Link encap:Ethernet HWaddr ba:80:b4:ba:c2:0d
            BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

root@vb1:/home/tmd#

root@vb1: /home/tmd

vb1:/home/tmd# ifconfig veth1 10.0.0.1/24

vb1:/home/tmd# ifconfig

```
veth1      Link encap:Ethernet  HWaddr 0e:1c:58:ad:d7:aa
            inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

vb1:/home/tmd#

root@vb1: /home/tmd

root@vb1:/home/tmd# ifconfig veth2 10.0.0.2/24

root@vb1:/home/tmd# ifconfig

```
veth2      Link encap:Ethernet  HWaddr ba:80:b4:ba:c2:0d
            inet addr:10.0.0.2  Bcast:10.0.0.255  Mask:255.255.255.0
            inet6 addr: fe80::b880:b4ff:feba:c20d/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:3 errors:0 dropped:0 overruns:0 frame:0
            TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:266 (266.0 B)  TX bytes:426 (426.0 B)
```

root@vb1:/home/tmd#

# Ping Between Namespaces

- Try ping
- Notice it fails (at least for local addresses)!
- Remember to bring up loopback via “ifconfig lo up”
- Try ping again and succeed!

root@vb1: /home/tmd

vb1:/home/tmd# ifconfig lo up

vb1:/home/tmd# ping 10.0.0.2

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=0.033 ms

64 bytes from 10.0.0.2: icmp\_seq=2 ttl=64 time=0.063 ms

64 bytes from 10.0.0.2: icmp\_seq=3 ttl=64 time=0.060 ms

64 bytes from 10.0.0.2: icmp\_seq=4 ttl=64 time=0.063 ms

^C

--- 10.0.0.2 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3062ms

rtt min/avg/max/mdev = 0.033/0.054/0.063/0.015 ms

vb1:/home/tmd#

root@vb1: /home/tmd

root@vb1:/home/tmd# ifconfig lo up

root@vb1:/home/tmd# ifconfig

lo Link encap:Local Loopback  
inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:65536 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

veth2 Link encap:Ethernet HWaddr ba:80:b4:ba:c2:0d

inet addr:10.0.0.2 Bcast:10.0.0.255 Mask:255.255.255.0

inet6 addr: fe80::b880:b4ff:feba:c20d/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:21 errors:0 dropped:0 overruns:0 frame:0

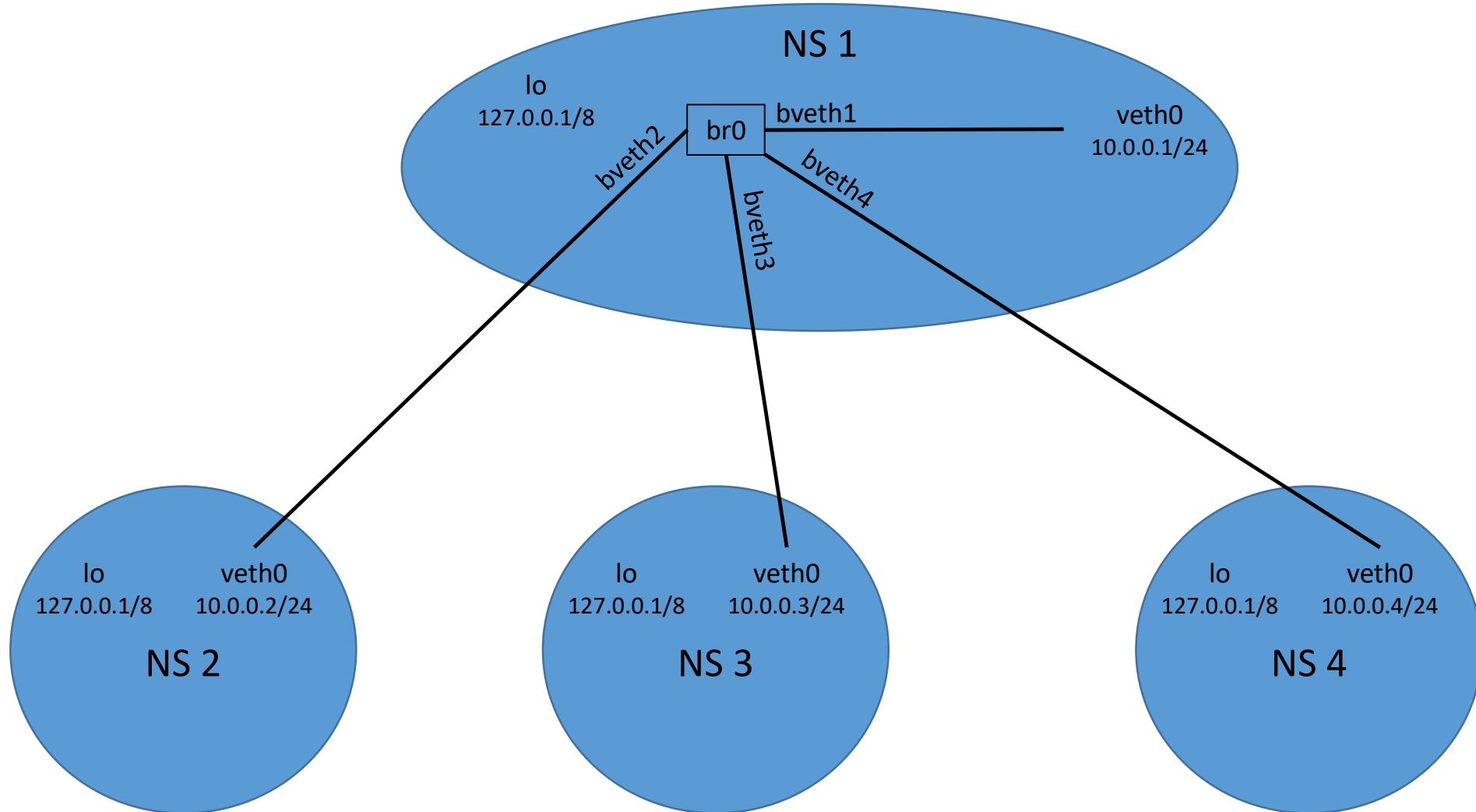
TX packets:18 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:1538 (1.5 KB) TX bytes:1412 (1.4 KB)

root@vb1:/home/tmd#

# Connecting Multiple Namespaces



# Tasks & Commands

- Create the namespaces
- Create a veth in each namespace, and move one end to bridge NS
  - `ip link add veth0 type veth peer bvethN (N=NS number)`
  - `ip link set bvethN netns ns1`
- Create a bridge, and configure it to be up
  - `ip link add br0 type bridge`
  - `ifconfig br0 up`
- Attach all interfaces to the bridge
  - `ip link set bvethX master br0 (repeat X for all NS)`



## Step1:

- Four namespaces, each creating a veth, and moving one half of that veth in NS1 (NS1 creation not shown)
- The names can be anything, but preceding name by “b” shows it will be attached to the bridge
- Names can be the same across namespaces, but must be unique within a namespace

```
ns1
vb1:/home/tmd# ip link | cut -c1-79
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4: bveth3@if5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
   link/ether 32:db:88:33:da:10 brd ff:ff:ff:ff:ff:ff link-netnsid 1
5: bveth2@if6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
   link/ether 66:c4:42:d2:9c:f2 brd ff:ff:ff:ff:ff:ff link-netnsid 0
6: bveth4@if5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
   link/ether 36:7b:29:03:8e:fe brd ff:ff:ff:ff:ff:ff link-netnsid 2
7: bveth1@veth0: <BROADCAST,MULTICAST,M-DOWN> mtu 1500 qdisc noop state DOWN mode
   link/ether 52:aa:7a:be:3d:fa brd ff:ff:ff:ff:ff:ff
8: veth0@bveth1: <BROADCAST,MULTICAST,M-DOWN> mtu 1500 qdisc noop state DOWN mode
   link/ether fe:01:63:e0:e1:85 brd ff:ff:ff:ff:ff:ff
vb1:/home/tmd#
```

```
ns2
vb1:/home/tmd# ip link add veth0 type veth peer name bveth2
vb1:/home/tmd# ip link set bveth2 netns ns1
vb1:/home/tmd#
```

```
ns3
vb1:/home/tmd# ip link add veth0 type veth peer name bveth3
vb1:/home/tmd# ip link set bveth3 netns ns1
vb1:/home/tmd#
```

```
ns4
vb1:/home/tmd# ip link add veth0 type veth peer name bveth4
vb1:/home/tmd# ip link set bveth4 netns ns1
vb1:/home/tmd#
```

## Step2:

- Assign IP addresses, bring up lo
- In NS1 bring up other end of veths, create bridge, and attach them all to the bridge

```
ns1
vb1:/home/tmd# ifconfig veth0 10.0.0.1/24
vb1:/home/tmd# ifconfig lo up
vb1:/home/tmd# ifconfig bveth1 up
vb1:/home/tmd# ifconfig bveth2 up
vb1:/home/tmd# ifconfig bveth3 up
vb1:/home/tmd# ifconfig bveth4 up
vb1:/home/tmd# ip link add br0 type bridge
vb1:/home/tmd# ifconfig br0 up
vb1:/home/tmd# ip link set bveth1 master br0
vb1:/home/tmd# ip link set bveth2 master br0
vb1:/home/tmd# ip link set bveth3 master br0
vb1:/home/tmd# ip link set bveth4 master br0
vb1:/home/tmd#

ns2
vb1:/home/tmd# ip link add veth0 type veth peer name bveth2
vb1:/home/tmd# ip link set bveth2 netns ns1
vb1:/home/tmd# ifconfig veth0 10.0.0.2/24
vb1:/home/tmd# ifconfig lo up
vb1:/home/tmd#

ns3
vb1:/home/tmd# ip link add veth0 type veth peer name bveth3
vb1:/home/tmd# ip link set bveth3 netns ns1
vb1:/home/tmd# ifconfig veth0 10.0.0.3/24
vb1:/home/tmd# ifconfig lo up
vb1:/home/tmd#

ns4
vb1:/home/tmd# ip link add veth0 type veth peer name bveth4
vb1:/home/tmd# ip link set bveth4 netns ns1
vb1:/home/tmd# ifconfig veth0 10.0.0.4/24
vb1:/home/tmd# ifconfig lo up
vb1:/home/tmd#
```

```
ns1
yb1:/home/tmd# brctl show
bridge name      bridge id        STP enabled      interfaces
br0              8000.3635478bc2ff no                bveth1
                 bveth2
                 bveth3
                 bveth4

yb1:/home/tmd#
```

```
ns2
yb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:4 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:308 (308.0 B)  TX bytes:308 (308.0 B)

veth0       Link encap:Ethernet  HWaddr c6:06:74:c2:03:b2
            inet addr:10.0.0.2  Bcast:10.0.0.255  Mask:255.255.255.0
            inet6 addr: fe80::c406:74ff:fec2:3b2/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:142 errors:0 dropped:0 overruns:0 frame:0
            TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:10536 (10.5 KB)  TX bytes:2294 (2.2 KB)

yb1:/home/tmd#
```

```
ns3
yb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth0       Link encap:Ethernet  HWaddr b6:8f:80:9f:5c:48
            inet addr:10.0.0.3  Bcast:10.0.0.255  Mask:255.255.255.0
            inet6 addr: fe80::b48f:80ff:fe9f:5c48/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:67 errors:0 dropped:0 overruns:0 frame:0
            TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:5214 (5.2 KB)  TX bytes:2266 (2.2 KB)

yb1:/home/tmd#
```

```
ns4
yb1:/home/tmd# ifconfig -a
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth0       Link encap:Ethernet  HWaddr 6e:dc:58:2a:40:55
            inet addr:10.0.0.4  Bcast:10.0.0.255  Mask:255.255.255.0
            inet6 addr: fe80::6cdc:58ff:fe2a:4055/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:69 errors:0 dropped:0 overruns:0 frame:0
            TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:5354 (5.3 KB)  TX bytes:2448 (2.4 KB)

yb1:/home/tmd#
```

vb1:/home/tmd# ip address

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
5: bveth2@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br0 state UP group default qlen 1000
    link/ether 66:c4:42:d2:9c:f2 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::64c4:42ff:fed2:9cf2/64 scope link
        valid_lft forever preferred_lft forever
7: bveth1@veth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br0 state UP group default qlen 1000
    link/ether 52:aa:7a:be:3d:fa brd ff:ff:ff:ff:ff:ff
    inet6 fe80::50aa:7aff:febe:3dfa/64 scope link
        valid_lft forever preferred_lft forever
8: veth0@bveth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether fe:01:63:e0:e1:85 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global veth0
        valid_lft forever preferred_lft forever
    inet6 fe80::fc01:63ff:fee0:e185/64 scope link
        valid_lft forever preferred_lft forever
13: bveth3@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br0 state UP group default qlen 1000
    link/ether 36:35:47:8b:c2:ff brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::3435:47ff:fe8b:c2ff/64 scope link
        valid_lft forever preferred_lft forever
14: bveth4@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br0 state UP group default qlen 1000
    link/ether 52:32:0a:ea:c0:83 brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet6 fe80::5032:aff:feea:c083/64 scope link
        valid_lft forever preferred_lft forever
15: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 36:35:47:8b:c2:ff brd ff:ff:ff:ff:ff:ff
    inet6 fe80::54b6:9ff:fea3:7627/64 scope link
        valid_lft forever preferred_lft forever
vb1:/home/tmd#
```

ns1

```
^C
vb1:/home/tmd# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.034 ms
^C
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.034/0.034/0.034/0.000 ms
vb1:/home/tmd# ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.024 ms
^C
--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.024/0.024/0.024/0.000 ms
vb1:/home/tmd# ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.052 ms
^C
--- 10.0.0.4 ping statistics ---
```

ns2

```
vb1:/home/tmd# ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.066 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.048/0.063/0.076/0.013 ms
vb1:/home/tmd#
```

ns3

```
vb1:/home/tmd# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.085 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.036/0.065/0.085/0.021 ms
vb1:/home/tmd#
```

ns4

```
vb1:/home/tmd# ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.060 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.073 ms
^C
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.060/0.069/0.075/0.009 ms
vb1:/home/tmd#
```

# Automating It

- Instead of manually doing this in each window, could do it from a single script
- The command “ip netns exec <ns> <cmd> <args>” lets you run a command in another namespace
- It can configure interfaces in other namespaces, move veths, etc
- Can launch xterms/gnome terminals in other namespaces
- Could use LXC, which is driven off configuration files

<https://help.ubuntu.com/lts/serverguide/lxc.html>

# Useful Commands

ip netns add <ns>

ip netns exec <ns> <cmd> <args>

ip netns identify

ip link add <name 1> type veth peer name <name 2>

ip link set <interface> netns <ns>

ip link set <interface> name <new name>

ifconfig lo up

ifconfig <interface> <IP address>/<mask length>

ip link add <name> type bridge

ip link set <interface> master <bridge name>

ip link

ip address

brctl

apt-get install bridge-utils

apt-get install lxc

- Assign a name to a namespace
- Execute a command from within a specified namespace
- Identify the namespace of the current process
- Create a virtual Ethernet with two interfaces
- Move an interface into the specified namespace
- Change the name of an interface
- Enable the built-in loopback interface, needed for local networking
- Assign an IP address to an interface
- Create a virtual bridge (often named br0, remember to “ifconfig up” it)
- Attach an interface to a bridge
- See or modify interfaces
- See or modify protocol or hardware addresses
- See or modify virtual bridges, and attach/detach interfaces
- Install the bridge utilities on Ubuntu
- Install LXC, a container management system