

Management and Operations of Networks, Services, and Systems

Network Provisioning with Python/Linux

Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC

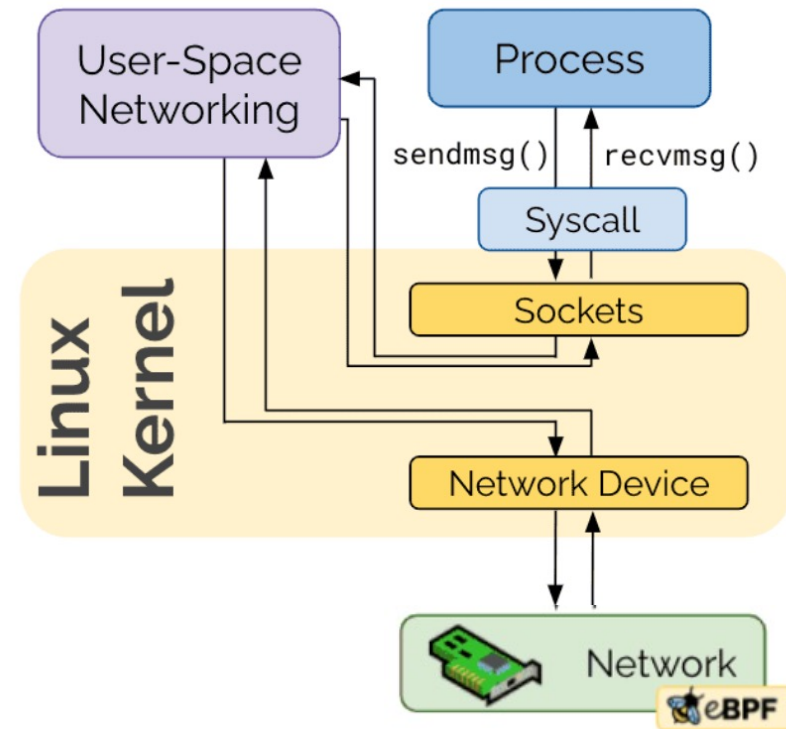
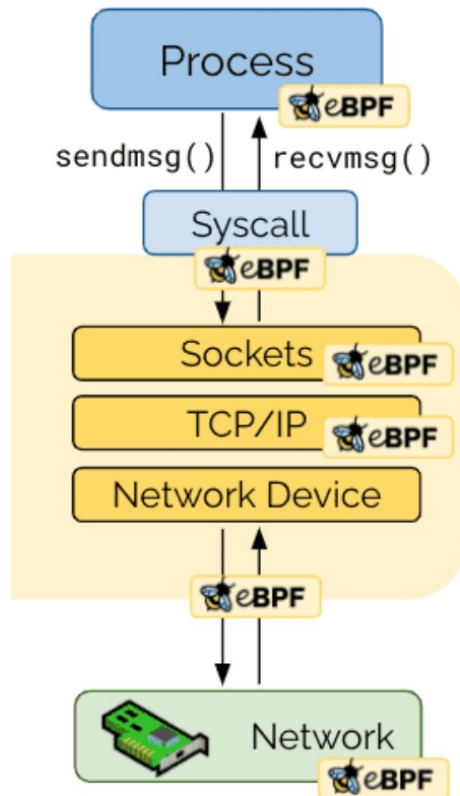


Linux networking

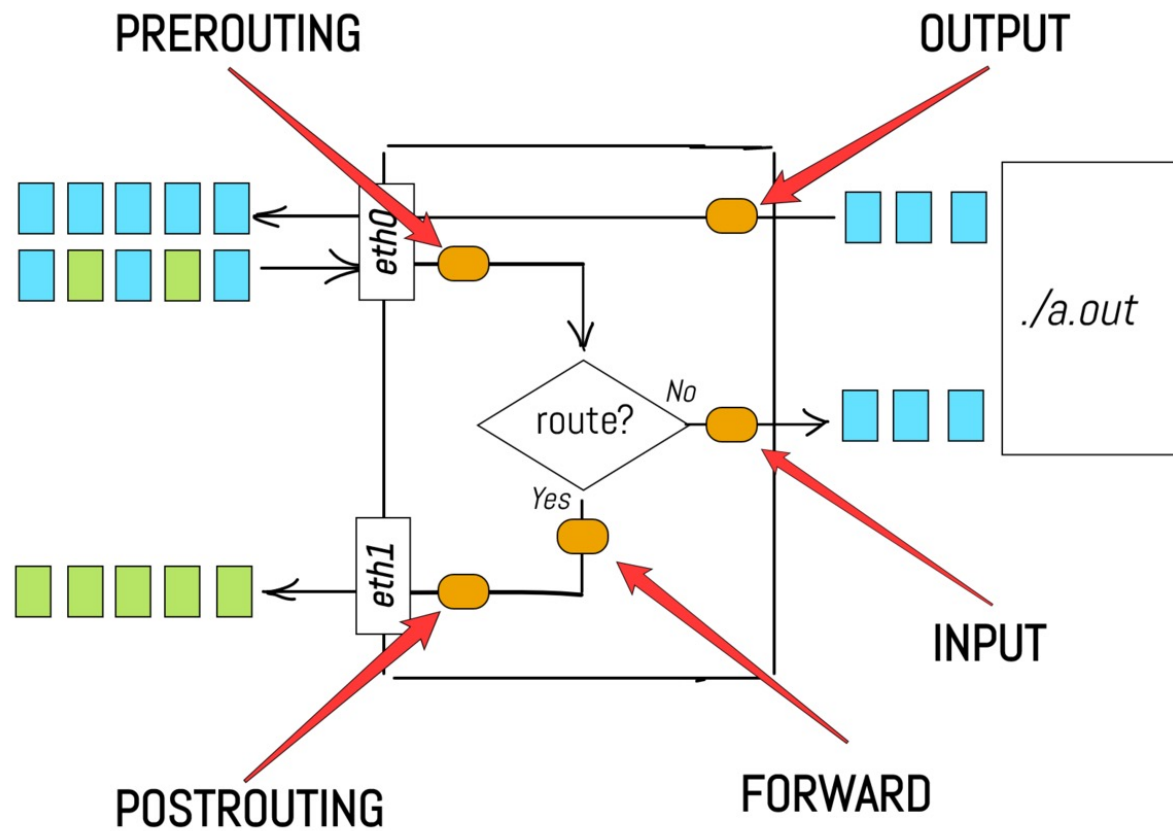
- User-space vs. kernel-space
- Routing
- Virtual interfaces and bridges
- Network namespaces



Linux kernel vs. user space networking



Linux routing (and iptables)

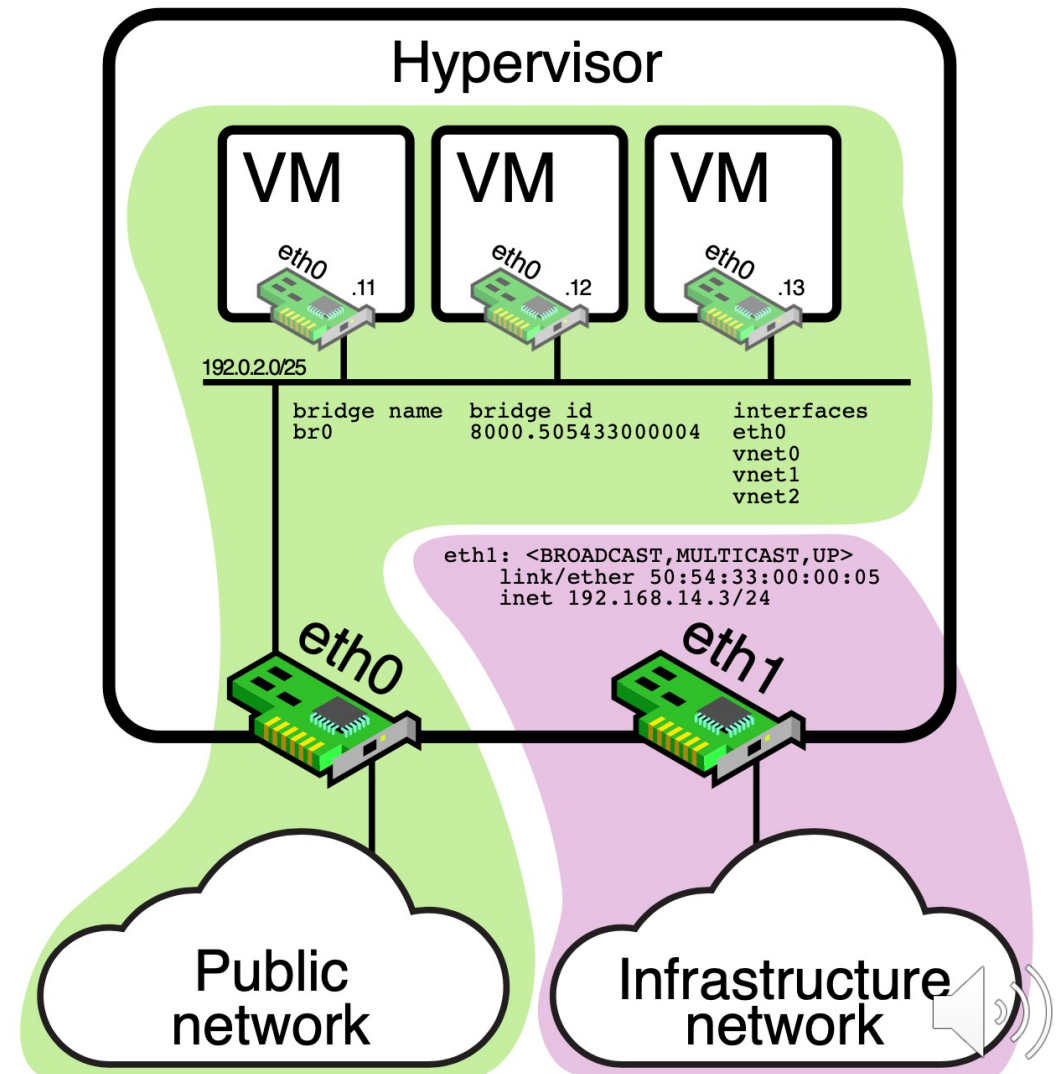


Virtual interface Virtual bridge

Hypervisor eth0/eth1 – physical interfaces

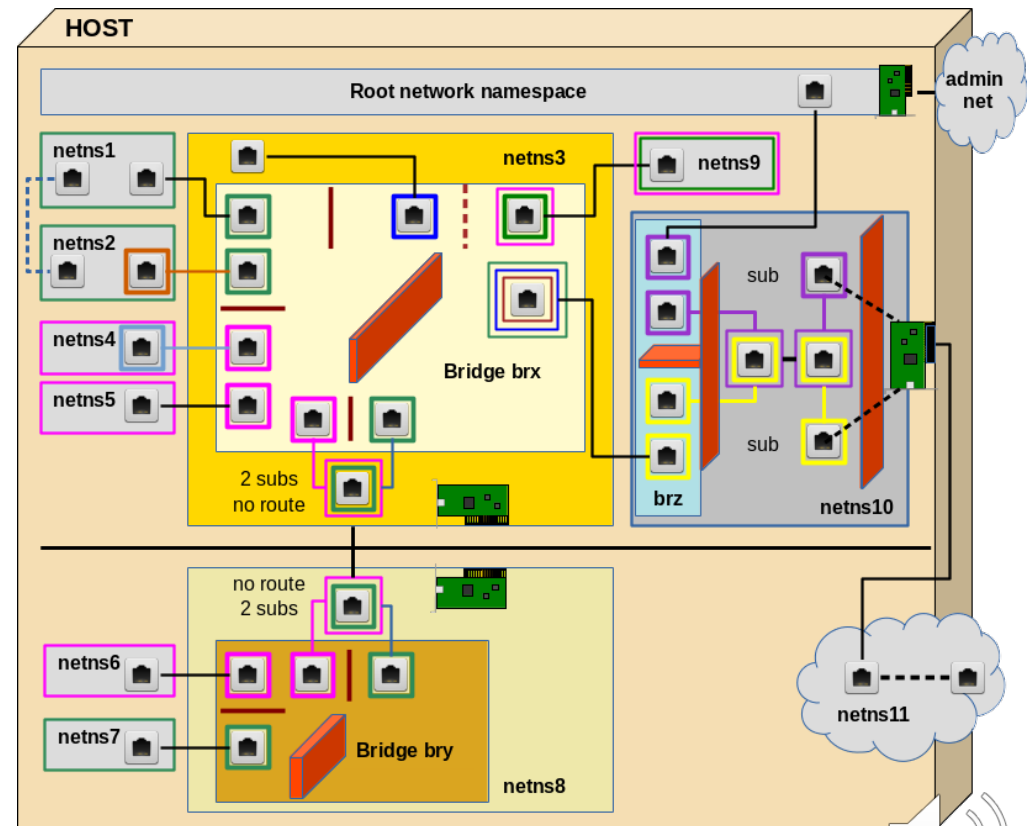
Hypervisor bridge br0 – virtual switch

VM's eth0 – virtual interfaces



Namespace isolation, virtual cables

- veth pairs are virtual cables to connect virtual interfaces
- Arbitrarily complex virtual network topologies
- One giant router if no virtualization
- Network namespaces netns solve this problem



Python programming for network automation

- Network source of truth
- Typical provisioning steps
 - Prepare configurations
 - Define management access to target devices
 - Push commands
- Templates



Network source of truth

- Reference of network assets and intended configurations
 - Used as input for configuration etc.

- Manually ok for small deployments

```
router['interfaces'] = ['eth0', 'eth1']  
router['IP'] = {'eth0': '10.1.1.254', 'eth1': '10.1.2.254'}
```

- Or code your way around big deployments

```
hosts = []  
for i in range(0,20):  
    Host_i = {'hostname':'ws' + str(i) + '.xpto.pt',  
              'IP':'10.1.2.' + str(i) + '/24',  
              'if':'eth1', 'gw': '10.1.2.254', 'host_i':i}  
    hosts.append(host_i)
```



What's wrong with this code – network wise?



- Support:
 - Helper library nsot <https://nsot.readthedocs.io>
 - full blown NSoT application netbox



Prepare configurations

- Define function that sets ip address on interface

```
def set_ip_addr (if, ip):  
    cmds = [  
        'ip link set ' + if + ' up',  
        'ip address add ' + ip + ' dev ' + if  
    ]  
    return cmds
```

- Host interfaces

```
host_cmds = {}  
for host in hosts:  
    cmds = set_ip_addr (host['if'], host['IP'])  
    host_cmds[host['hostname']] = cmds
```

- Router interfaces

```
router_cmds = [  
    'sysctl -w net.ipv4.ip_forward=1',  
]  
for if in router['IP'].keys():  
    router_cmds.extend(set_ip_addr(if, router['IP'][if]))
```

- Sanity check

```
print (host_cmds, router_cmds)
```



Define management access to each host

- Management interface already configured

```
mngt['router.xpto.pt'] = {  
    'address': 10.255.255.254, 'username': 'admin58234',  
    'priv_key': '~/.ssh/id_rsa_xpto'  
}  
for host in hosts:  
    mngt[host['hostname']] = {  
        'address': 10.255.255.' + host['host_i'],  
        'username': 'admin58234', 'priv_key': '~/.ssh/id_rsa_xpto' }
```

- For others may need to redirect via host (docker, mininet, ...)



Push commands via ssh

- Import ssh/network library

```
import netmiko
```

- Install if necessary

```
$pip install netmiko
```

- Run

```
for host in hosts:
    m = mngt[host['hostname']]
    device = ConnectHandler(
        host=m['address'], username=m['username'],
        device_type='linux', use_keys=True, key_file=m['priv_key'])
    cmds = host_cmds[host['hostname']]
    for cmd in cmds:
        device.send_command(cmd)
```



Templates

- Large configuration files with only a few parameters
- template + parameter db
=> config for each device
- Deploy config file

1. jinja2 template for cisco switch

```
{% for name, desc in if_descr.items() %}  
interface {{ name }}  
    description {{ desc }}  
{% end for %}
```

2. jinja2 mapping

```
if_descr = {  
    'GigabitEthernet0/1': 'client port',  
    'GigabitEthernet0/2': 'server port'  
}
```

3. jinja2 output

```
interface GigabitEthernet0/1  
    description client port  
interface GigabitEthernet0/2  
    description server port
```



Linux – a network in a box

- Span a new linux virtual machine
- Move half of the hosts to the network that only has the router
- Try to automatically create and provision the host and router interfaces
 - How are you going to connect to the management network? Do you need to adapt the commands to id the namespace?
- Make sure you test the connectivity between the two networks



Management and Operations of Networks, Services, and Systems

Network Provisioning with Python/Linux

Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC