



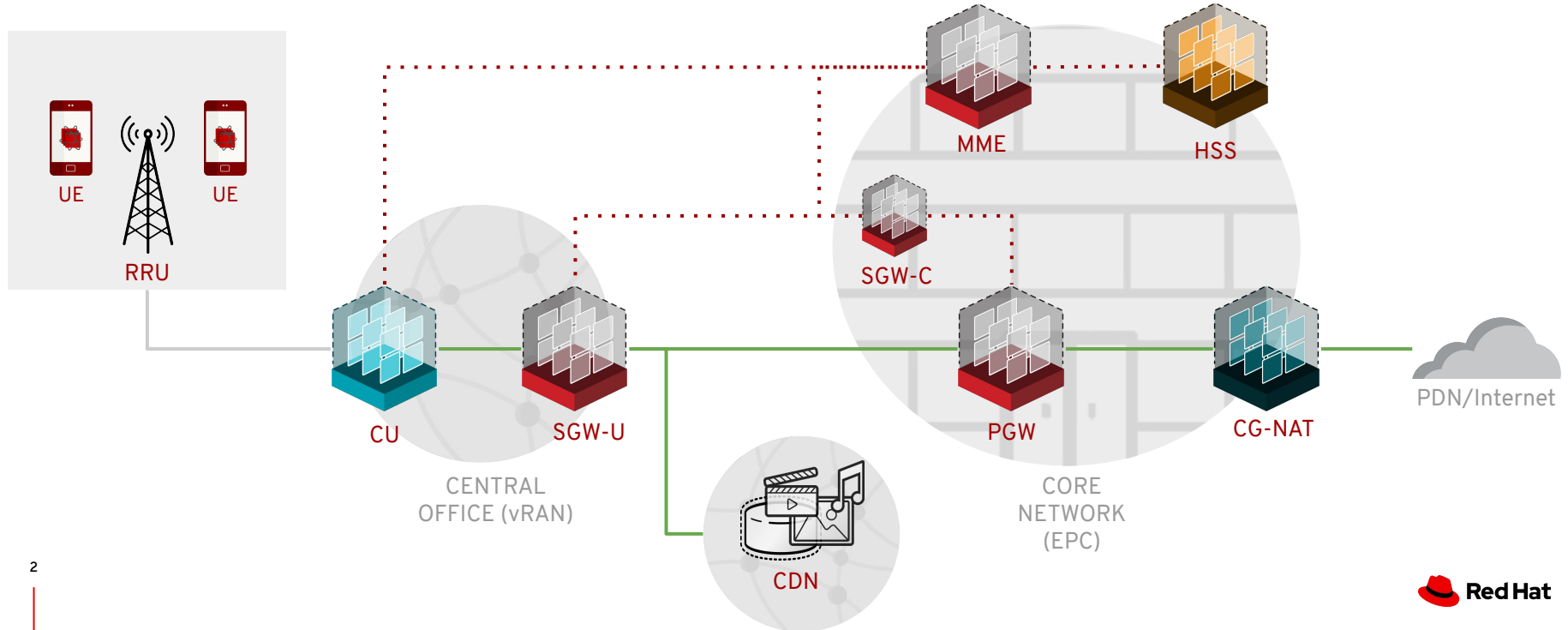
OpenStack at the vRan Edge

PTP Live demo, up to Kubernetes

Franck Baudin, Sr Principal Product Manager - NFV

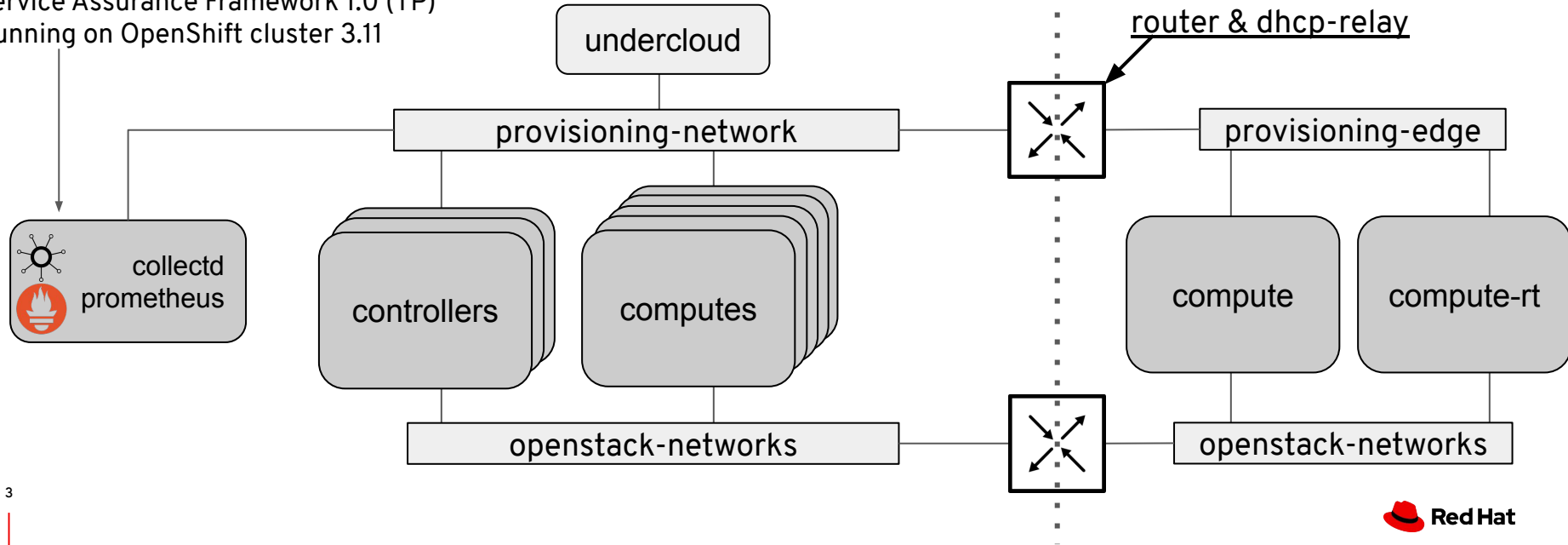


NFV is at every hop of Mobile networks

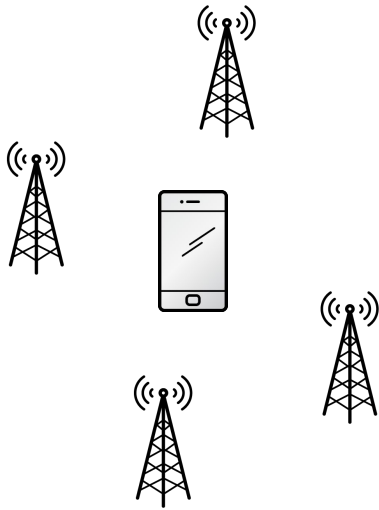


Deployment overview 1/3

Service Assurance Framework 1.0 (TP)
Running on OpenShift cluster 3.11

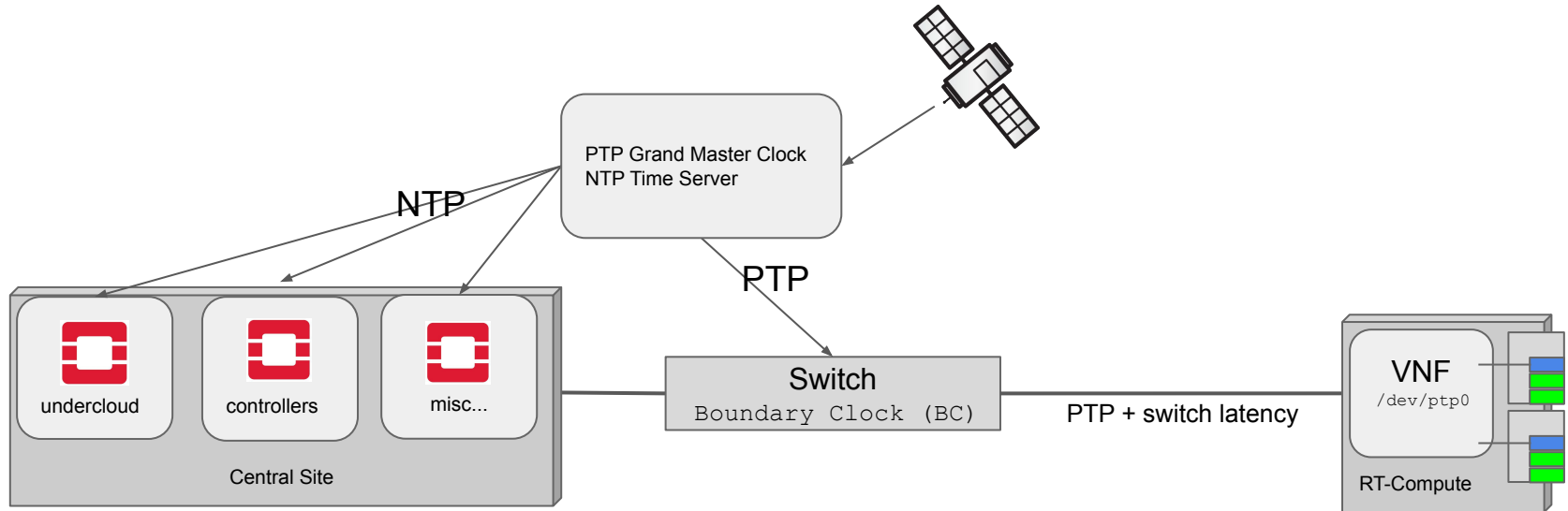


vRAN: why do we need PTP ?

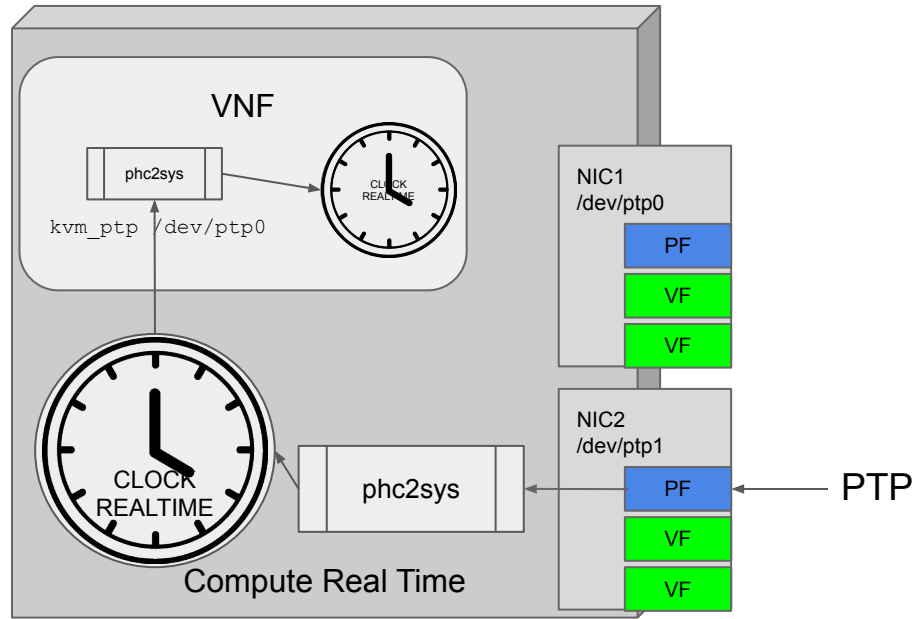


- Connection to multiple antennas
 - (+) Higher throughput
 - (-) Risk of interferences
 - Timing spec for radio fronthaul
 - CDMA2000: $\pm 3\mu\text{s} < \text{phase} < \pm 10\mu\text{s}$
 - LTE: $\pm 1.5\mu\text{s} < \text{phase} < \pm 5\mu\text{s}$
 - 5G: $\pm 130\text{ns}$
- GNSS receivers for each compute node?
 - Satellites not visible
 - High buildings “shadow”
 - In-buildings
 - Underground
 - Expensive

Deployment overview 2/3



Deployment overview 3/3



Deploy and configure the modules you need

- Modular & extensible platform
 - SDN
 - Storage
 - Monitoring
- Feature enablement:
 - 1 TripleO environment file
 - 1 parameter file

```
$ ls ./templates/*.yaml
global-config.yaml
collectd.yaml
dpdk-config.yaml
sriov-config.yaml
hci-dpdk-config.yaml
compute-rt-edge-config.yaml
ptp.yaml
ssl-certificates.yaml
```

```
$ openstack overcloud deploy \
-e $TRIPLEO/environments/metrics-collectd-qdr.yaml
-e templates/service-assurance-framework.yaml \
...
```

```
resource_registry:
  OS::TripleO::Services::MetricsQdr: ../docker/services/metrics/qdr.yaml
  OS::TripleO::Services::Collectd: ../docker/services/metrics/collectd.yaml
```

```
parameter_defaults:
  MetricsQdrConnectors:
    - host: qdr-white-port-5671-sa-telemetry.redhat.local
      port: 443
      role: edge
      sslProfile: sslProfile
      verifyHostname: false
```

NFV (auto) tuning: Mistral workflow

```
"cpu": { "count": 48 },
"memory": {"physical_mb": 131072 },
"numa_topology": {"cpus": [ {"cpu": 0,
    "thread_siblings": [ 0, 24],
    "numa_node": 0
  }, ... ] }
"nics": [{"name": "p3p1", "numa_node": 1 },
  ... ]
```



```
- name: ComputeOvsDpdkRTEdge0
  ServicesDefault:
    - OS::TripleO::Services::ComputeNeutronOvsDpdk
```

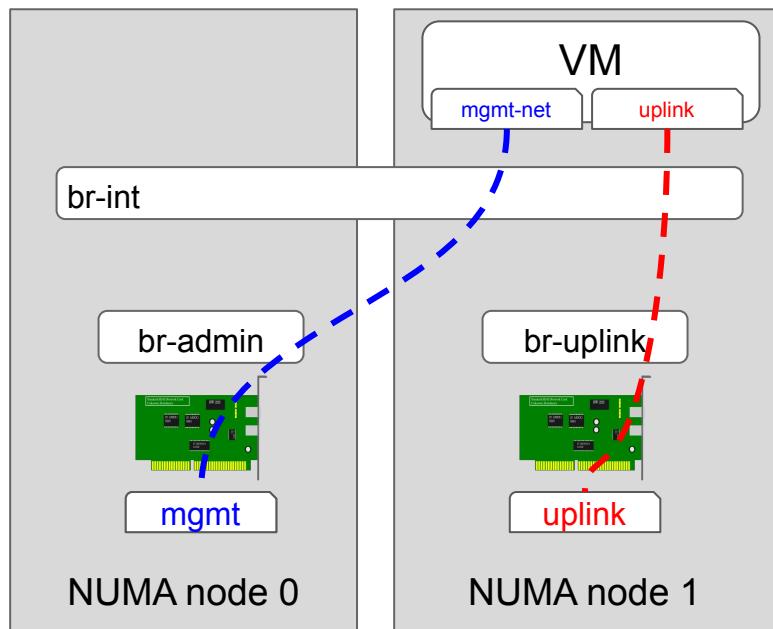


```
workflow_parameters:
  tripleo.derive_params.v1.derive_parameters:
    num_phy_cores_per_numa_node_for_pmd: 1
    huge_page_allocation_percentage: 95
```



```
ComputeOvsDpdkRTEdge0Parameters:
  IsolCpusList: 2-23,26-47
  KernelArgs: default_hugepagesz=1GB hugepagesz=1G
  hugepages=120 intel_iommu=on iommu=pt isolcpus=2-23,26-47
  NovaReservedHostMemory: 8192
  NovaVcpuPinSet: 2-6,8-15,17-23,26-30,32-39,41-47
  OvsDpdkCoreList: 0-1,24-25
  OvsDpdkSocketMemory: 2048,1024
  OvsPmdCoreList: 7,16,31,40
```


Numa Aware vSwitch



```
# TripleO
```

```
NeutronBridgeMappings:  
['mgmt:br-admin','uplink:br-uplink']
```

```
NeutronPhysnetNUMANodesMapping:  
{'mgmt': [0, 1], 'uplink': [1]}
```

```
# nova.conf
```

```
[neutron]  
physnets=mgmt,uplink  
  
[neutron_physnet_mgmt]  
numa_nodes=0,1  
  
[neutron_physnet_uplink]  
numa_nodes=1
```

Enabling vRAN use case

- Generic NFV characteristics

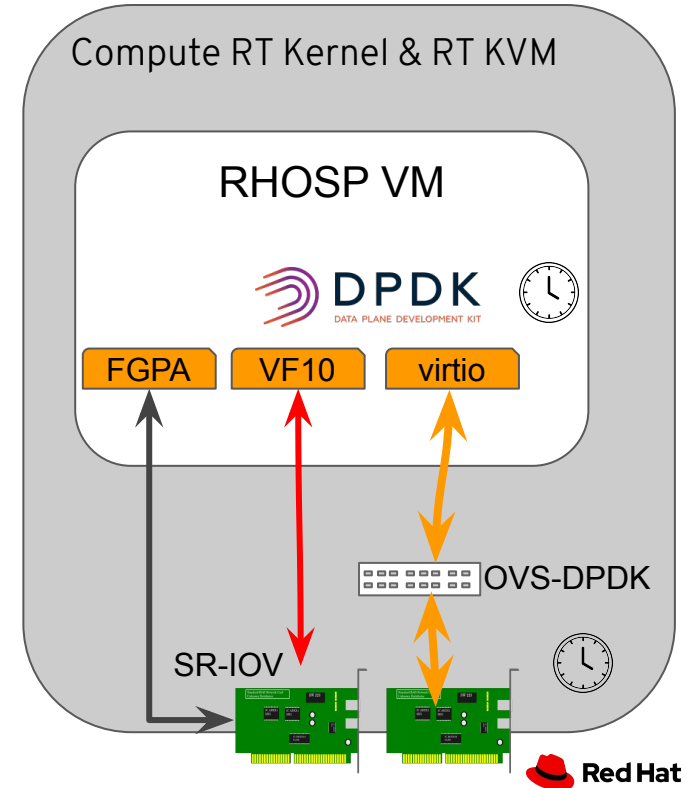
- Mix virtio + SRIOV VF
- Device role tagging

```
(overcloud)$ nova boot --nic net-id=$UPLINK_ID,tag=uplink  
--nic port-id=$RADIO_PORT_ID,tag=radio
```

```
(vm)$ jq '.devices[]|"\(.address) \(.mac) \(.tags[0])"' meta_data.json  
"0000:00:04.0 fa:16:3e:fa:89:0f uplink"  
"0000:00:06.0 fa:16:3e:6f:dd:e8 radio"
```

- vRAN Specific

- PTP
- FPGA (PCI passthrough)
- Real time

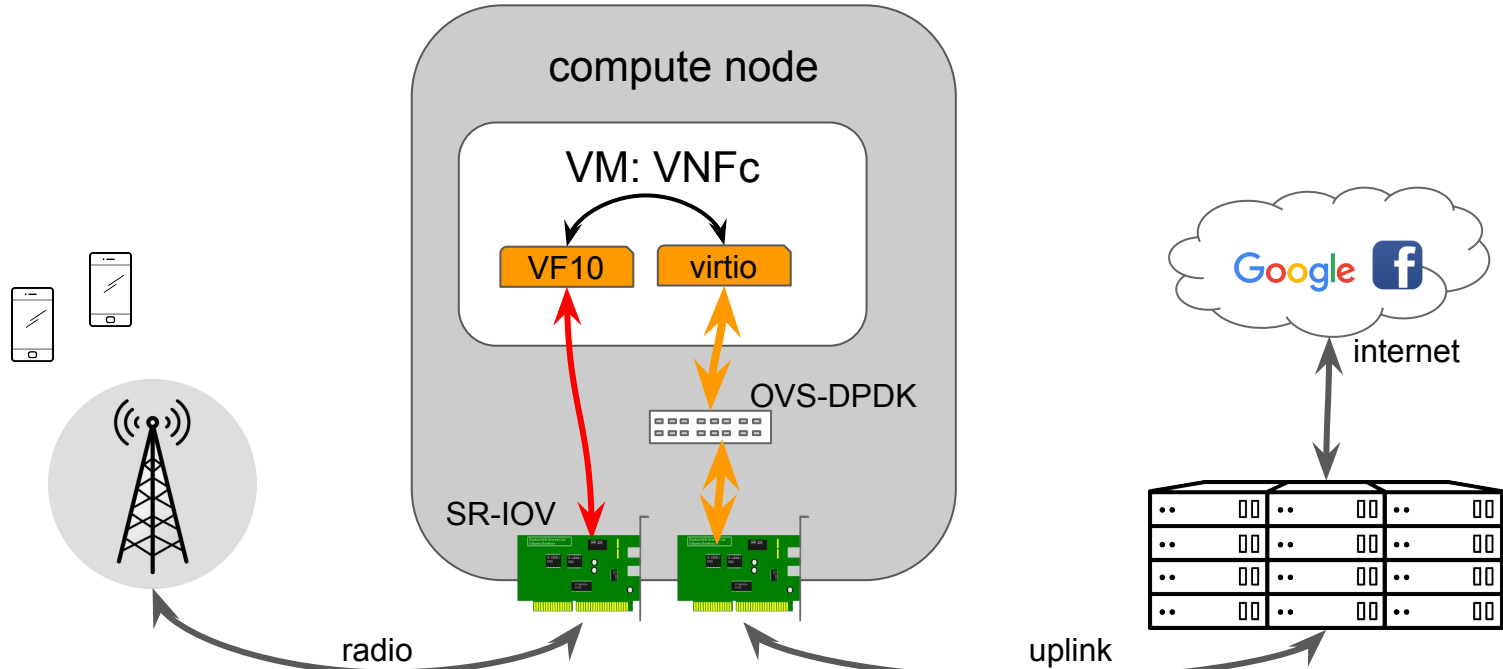


Let's have a look at the deployment



Post-Deployment validation

How to validate the NFVI?



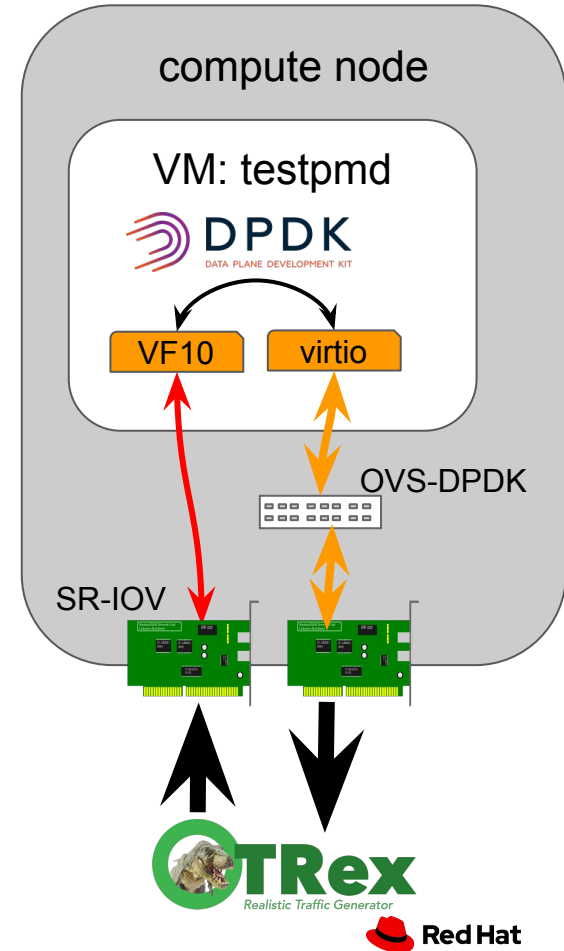
Simpler catch-all test

This is not a benchmark!

Make sure that the VM is not the bottleneck
=> Use DPDK testpmd to forward packets

Check expected Mpps and Latency
=> zero packet drop expected

Single flow, 64 Bytes frames



Issue detection

- Misconfiguration visible effect
 - Performance lower than expected, packet drop
 - Extra Packets
- Real example of misconfigurations caught
 - Isolation/partitioning (vCPU or OVS-DPDK PMD preemption)
 - => boot parameters, IRQ pinning, emulator thread pinning, ...
 - ToR switch misconfiguration (missing packets or extra packets)
 - BIOS misconfiguration (NUMA mode, Performance Policy, ...)
 - HW: PCIe x4 slot instead of x16, missing RAM bank (mem channel)
 - ...

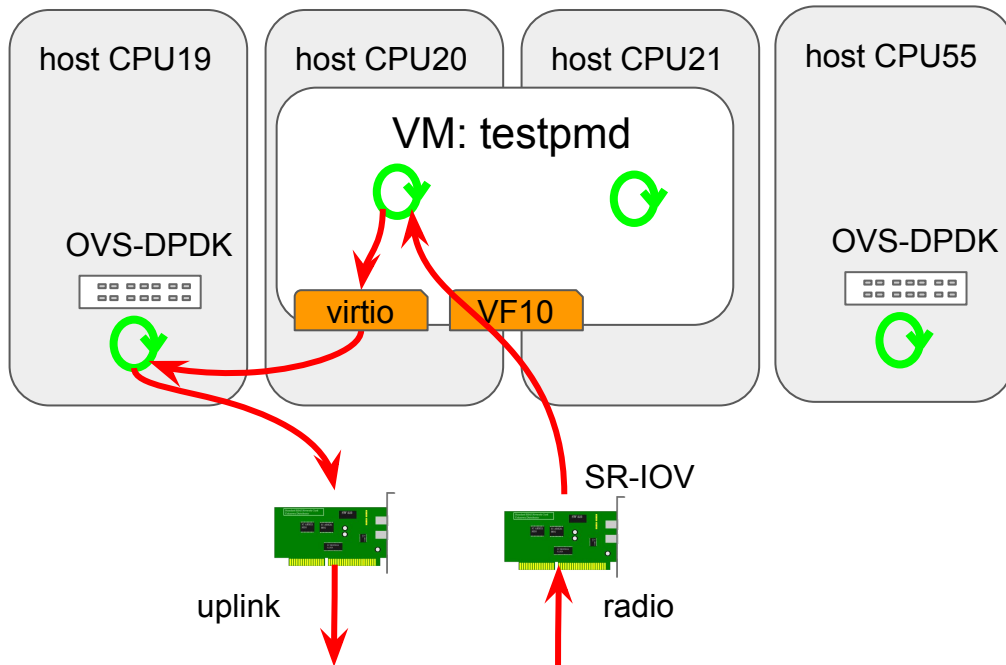
Troubleshooting

Packet journey: radio -> uplink



ACTIVE LOOP

```
while (1) {  
  RX-packet()  
  forward-packet()  
}
```

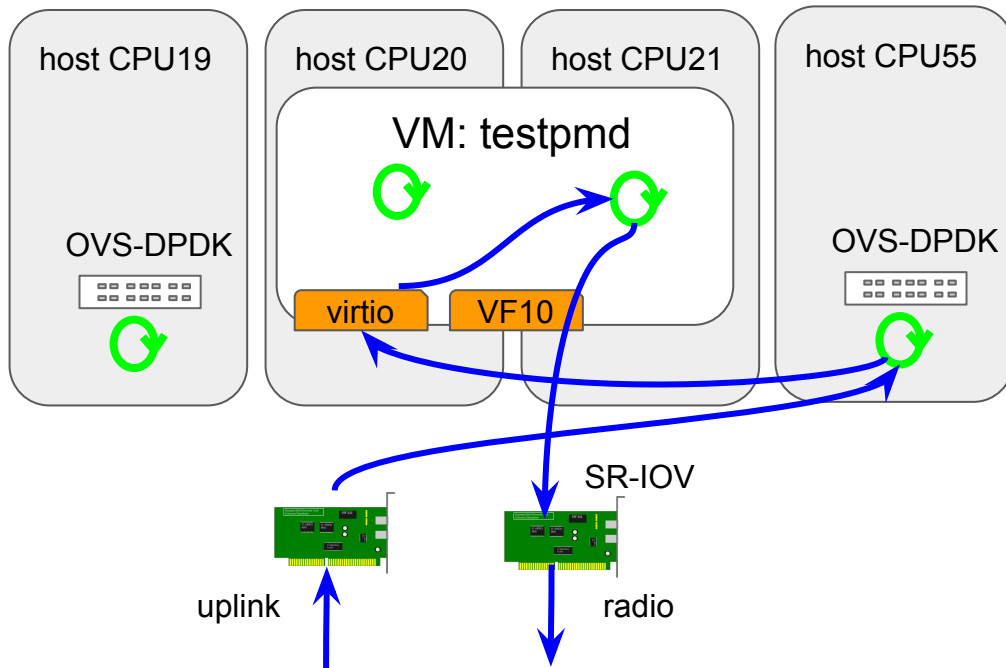


Packet journey: uplink -> radio



ACTIVE LOOP

```
while (1) {  
  RX-packet()  
  forward-packet()  
}
```



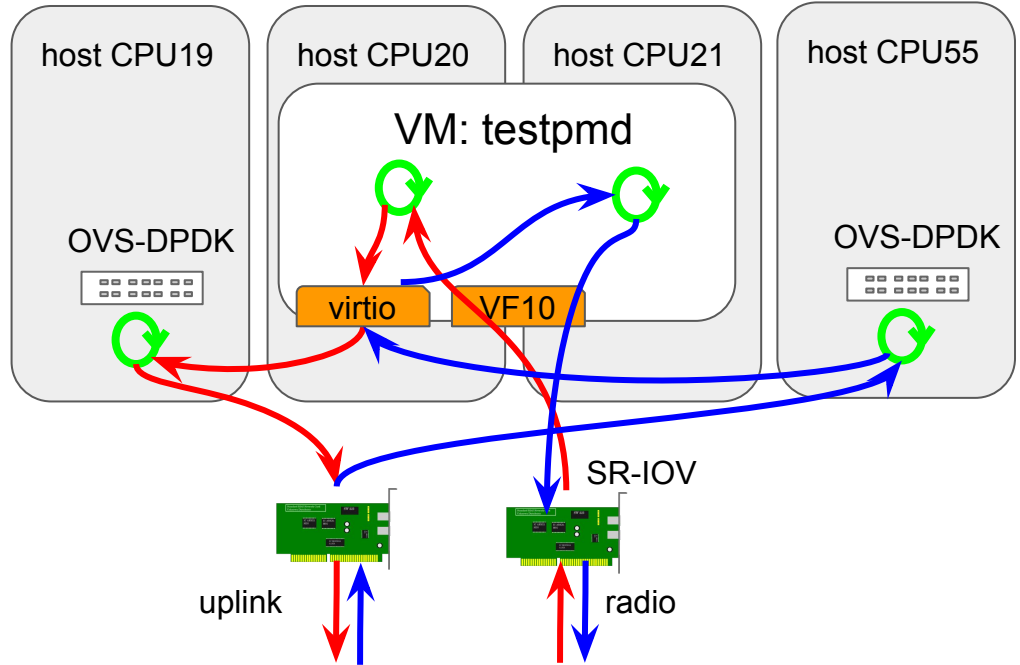
Packet journey: radio <-> uplink



ACTIVE LOOP



```
while (1) {  
  RX-packet()  
  forward-packet()  
}
```



No packets left behind!

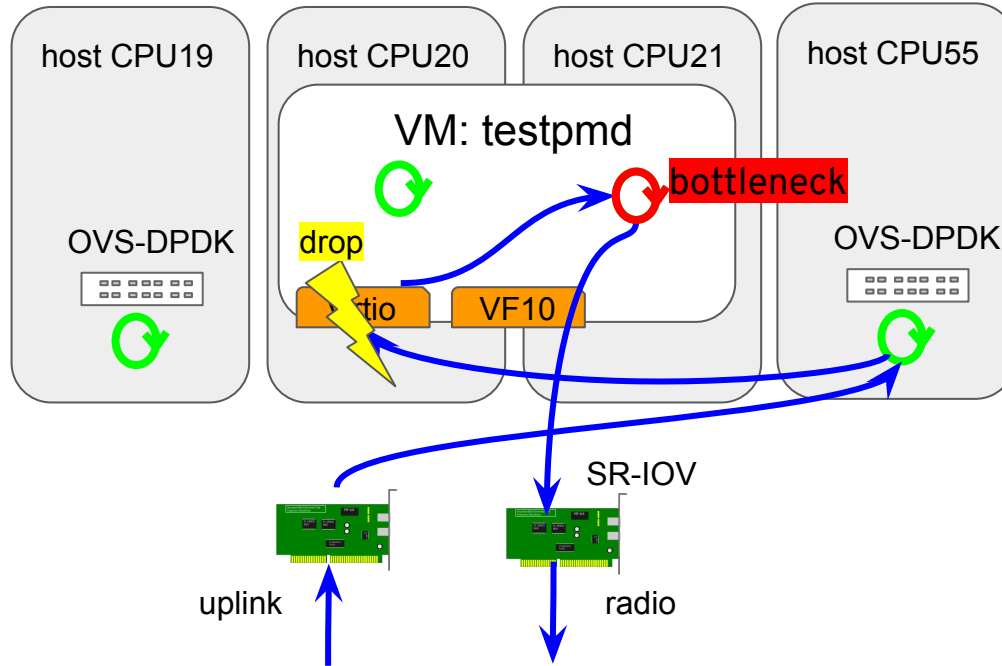
Packet are never lost, packets are dropped

- We always have a drop counter
- Except in case of a drop counter bug (SW, HW)

Packets are dropped when a queue is full

- A queue is full because it is not drained fast enough
- The bottleneck is the entity supposed to drain the queue

What if the VM is the bottleneck?

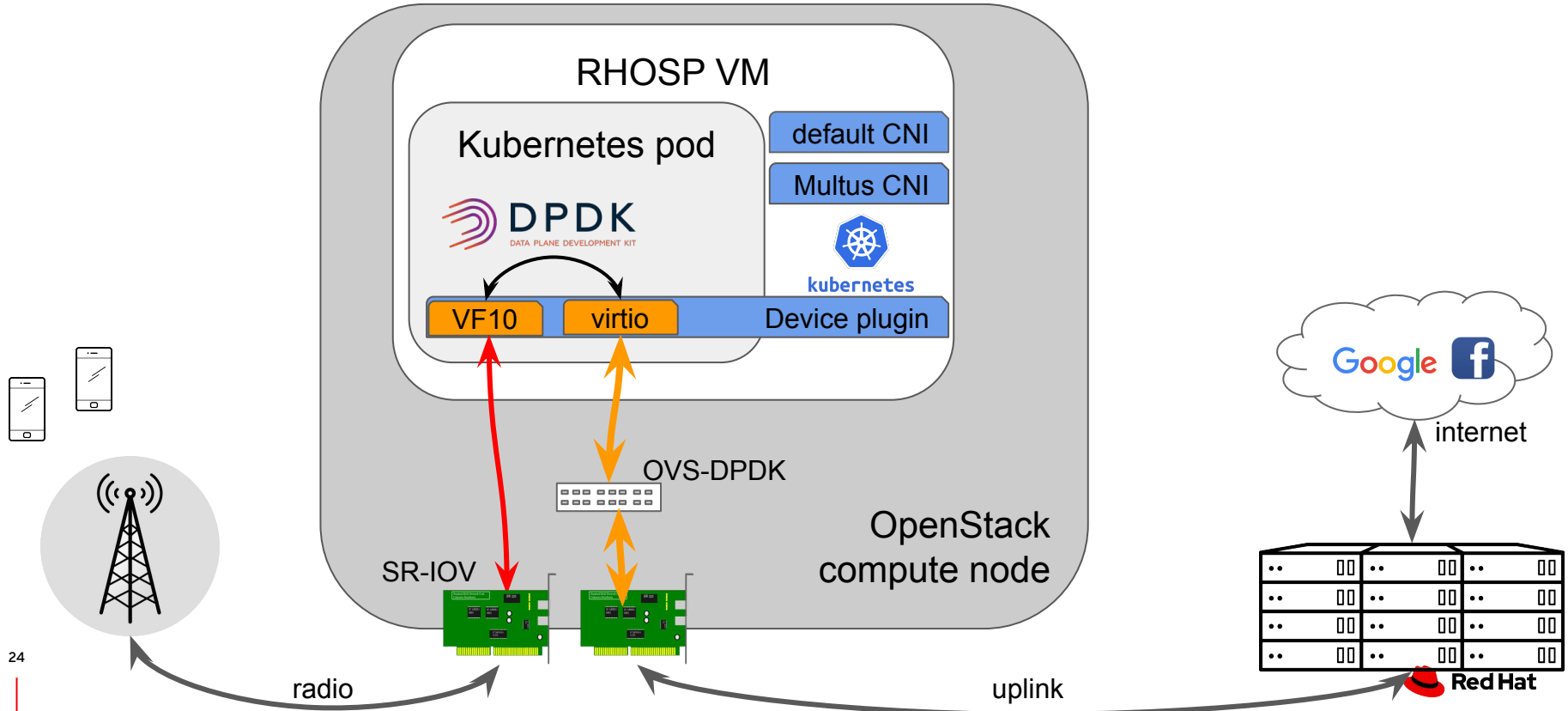


Demo



Final thoughts

Same packet flow with or w/o Kubernetes!



Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



twitter.com/RedHat

