A VIRTUAL FABRIC SWITCHING

JIE ZHENG@NSBU.VMWARE

BACKGROUND

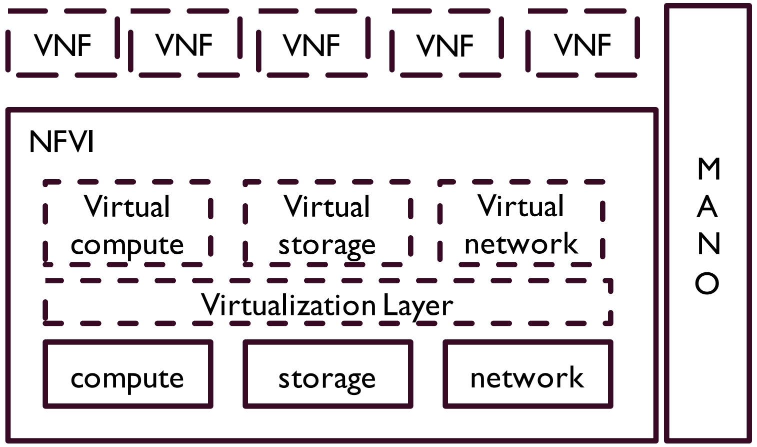
- Network Function Virtualization(NFV) prevails.

Vendor Lock-in

CapEx/OpEx

Time to Market

Open Telco Cloud



VNF VNF VNF VNF VNF

NFVI

Virtual compute

Virtual storage

Virtual network

Virtualization Layer

compute

storage

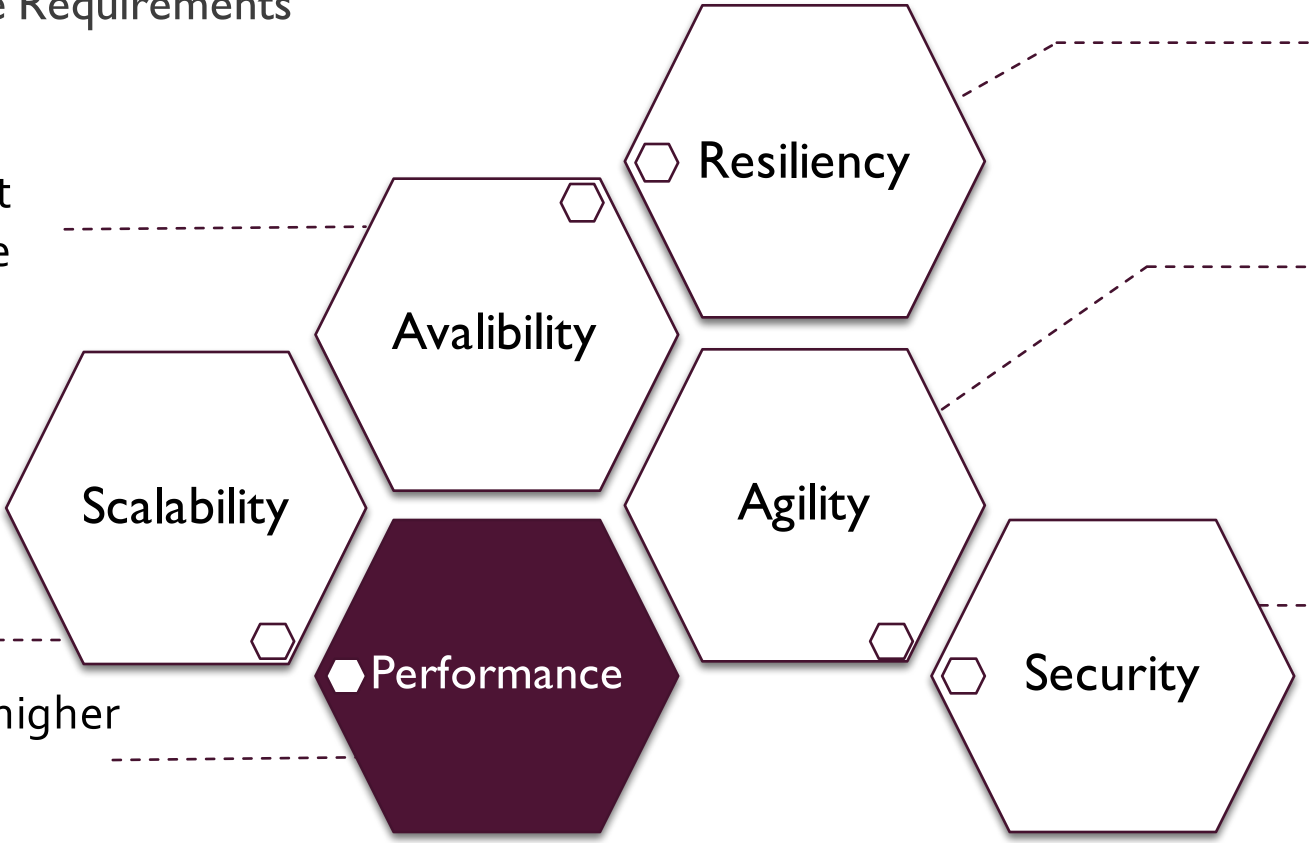
network

MANO

ETSI Reference model

CHALLENGES

- NFV Carrier-Grade Requirements



Resiliency

Self-healing,including infrastructure and VNF

Avalibility

Fault tolerant infrastructure

Agility

Elastic and rapid VNF&infra provisioning

Scalability

Infra scale with workloads

Performance

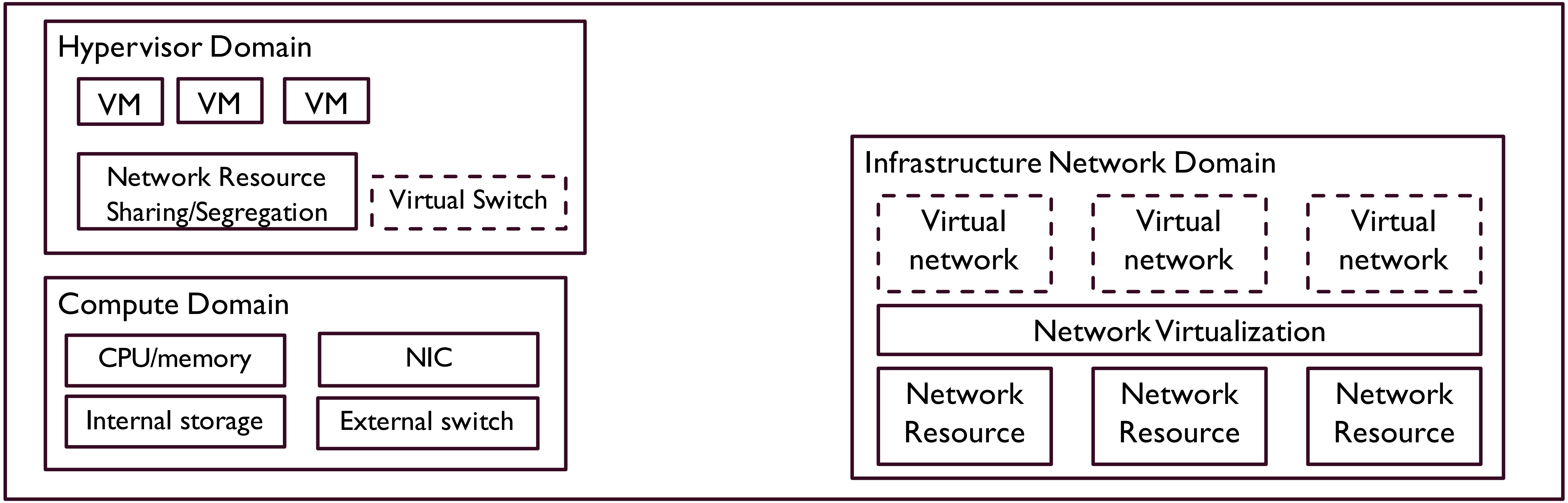
Lower latency and higher bandwidth

Security

VNF and Infrastructure segregation, intrinsic demand of multi-tenancy

CHALLENGES CONTINUED

- NFV Networking Infrastructure



1.x86/PCIe bandwidth/utilization

2.Memory/FSB bandwidth (https://dpdksummit.com/Archive/pdf/2017Asia/DPDK-China2017-Zheng-High-Speed-DPDK-PMD-LXC.pdf)

SOLUTION

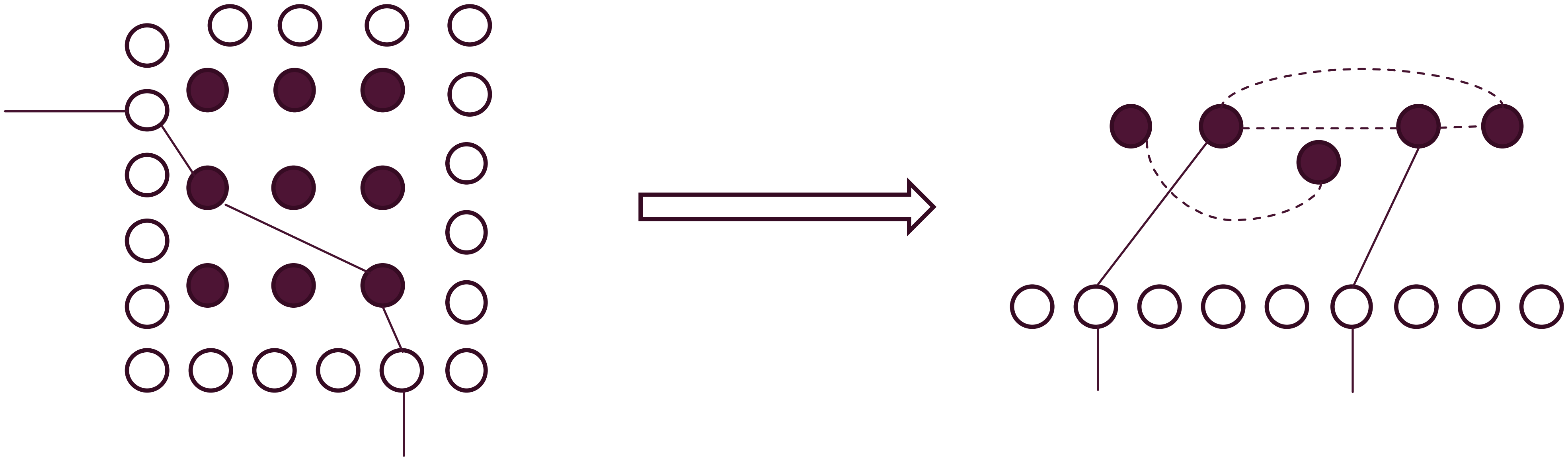
- Migrate virtual switching function to external switch(SR-IOV capable NIC built-in switches)
- Preserve more CPU/Memory resources to VNF computation
- Eliminate the overhead of software virtual switching, thus enabling scaling out.

- Infrastructure network should be virtualized instead to support datacenter-wide virtual network interconnection
- Avoid involving proprietary hardware.
- Not even with SDN switch with its controller

- X86 COTS hardware only
- **A new approach of virtual fabric switching** is introduced

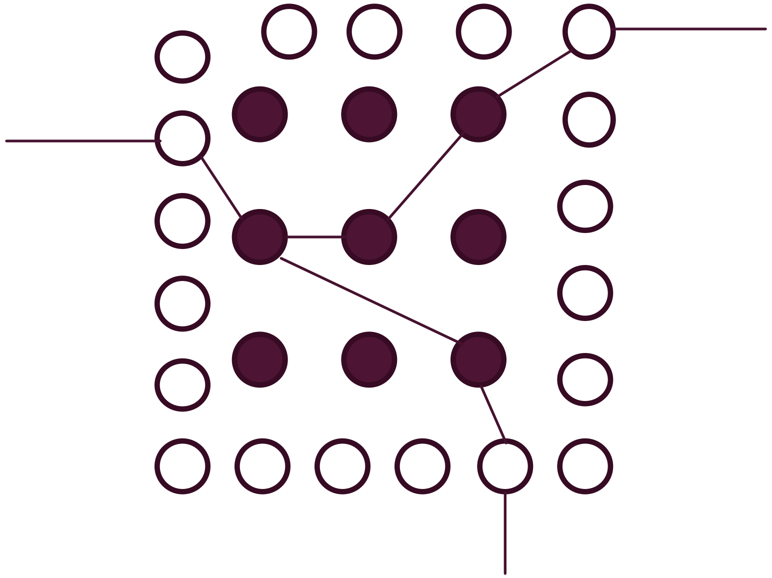
FABRIC STRUCTURE

- Undirected graph with edge nodes(attachable by customer) and core nodes (no customer access)
- Dynamically optimized path
- Can be bipartite graph if core nodes interconnection are ignored
- Aliased to Leaf node and spine node

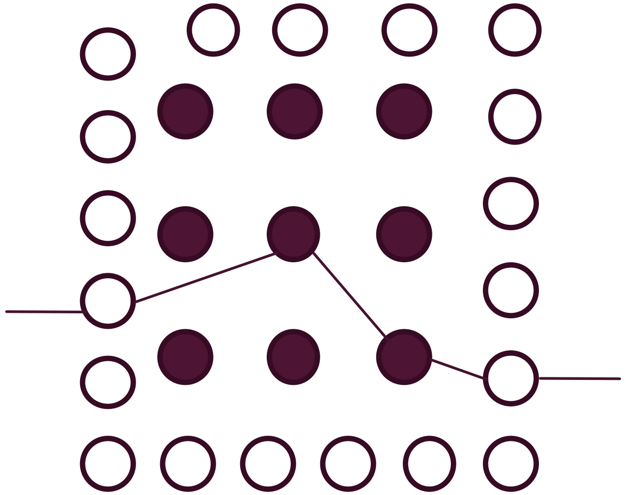


SERVICE MODEL

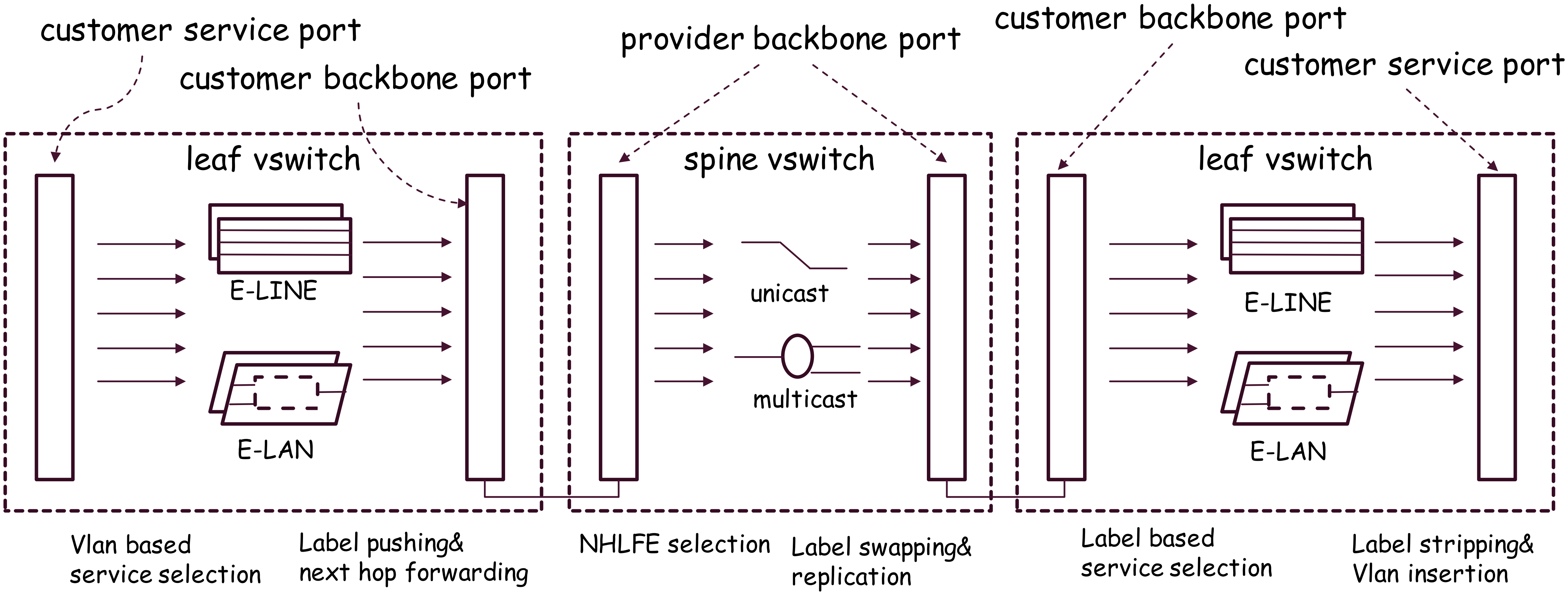
- Ether-LAN service.
- MAC based forwarding
- Infrastructure built-in multicast tree
- Replication on demand
- Optimized (minimum spanning tree)



- Ether-LINE service.
- Path determined forwarding
- Optimized (shortest path)

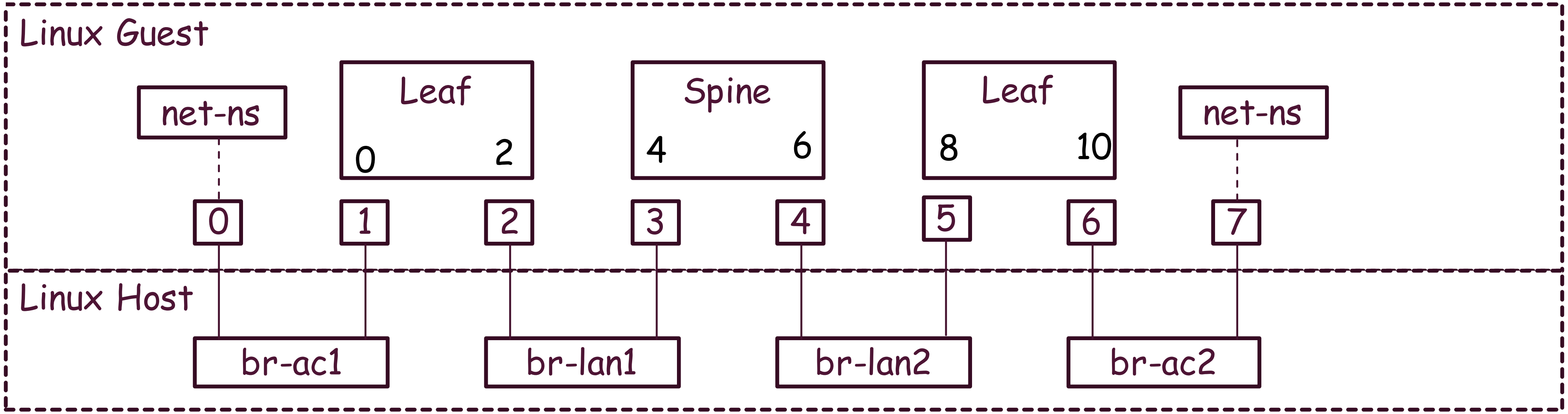


SWITCHING NODE INTERNALS



DEMO: ENVIRONMENT PRE-SETUP

- Qemu KVM emulated guest, both host and guest OS are CentOS 7
- 8 x e1000 devices with VLAN stripping and insertion offloading capability
- 4 x ovs bridges for 4 LANs emulation(2 attachment circuit lans and 2 common lans)
- Port 0 and port 7 are for customers and port 1-6 are for fabric switches
- For fabric ports, once taken over by e3datapath, re-index them [0,2,4,6,8,10]
- For customer ports, use Linux namespace and vlan sub-interfaces to segregate themselves



DEMO: E-LINE SERVICE

csp port 0	cbp port 2	pbp port 4	pbp port 6	cbp port 8	csp port 10
create two e-line services: 0 and 1					
Vlan1000 ----> e-line 0					
	E-line 0 next hop(to port 4) via port2 with label:1				
		Port 4 with label 1,knows next hop(to port 8) via port 6 with label 100			
				Port 8 with label 100 goes to e-line1	
					e-line1--->vlan 2000
					vlan 2000--->e-line1
			E-lan1 next hop(to port 6) via port 8 with label:10000		
		Port 6 with 10000,it knows next hop(to port 2) with label 10.			

DEMO: E-LAN SERVICE MULTICAST FORWARDING

csp port 0	cbp port 2	pbp port 4	pbp port 6	cbp port 8	csp port 10
create two e-lan services: 0 and 1 , and multicast next hop list:0					
Vlan3000 ----> e-lan 0					
	E-lan 0,find no fwd entry, multicast next hop(to port 4) via port2 with label:2				
		Port 4 with label 2, goes to multicast list0,perform RPF check and send replication (to port 8) via port 6 with label:101			
				Port 8 with label 101 goes to e-lan1	
					e-lan1--->vlan 4000
					vlan 4000--->e-lan1
			E-lan1 still finds no fwd entry, use multicast nexthop(to port 6)via port 8 with label:10001		
		Port 6 with 10001,does multicast forwarding, finally goes to port 2 via port 4 with label:11			

DEMO: E-LAN SERVICE UNICAST FORWARDING

- At leaf virtual switch, a fwd entry is found with deterministic <label, nhlfe>
- At spine virtual switch, single next hop is bound to input label entry, no multicast list is searched

PROGRAM FRAMEWORK

- Node encapsulation(come and go at any time)
- Network Interface encapsulation(interface hot pluggable, extendable and support slow path)
- Highly modularized data structure and base function(logging, init and config).
- Zeromq based and TLV encoded API framework(C and Python binding)
- Fast indexing technique with SSE4.2 or AVX2 instruction set
- Optimized packet delivery framework(burst processing and cached forwarding credentials) as a result of temporal locality

KNOWN ISSUE

- E-LAN can not snoop multicast packet from fabric port for mac-learning(manually register them)
- Unit test does not cover all C code, none for Python API
- No performance benchmark
- Simple data plane function as it is, complex or complicated control plane and management plane function?
  No evaluation

VISION

- Higher line rate
- Switch fabric is intrinsically highly available(fault detection and self-healing)
- Fabric visualization(Bigswitch Monitoring Fabic and SPB)
- Tuned for container networking
  - Redefine network resource
  - Inter-container communication
  - vxlan termination