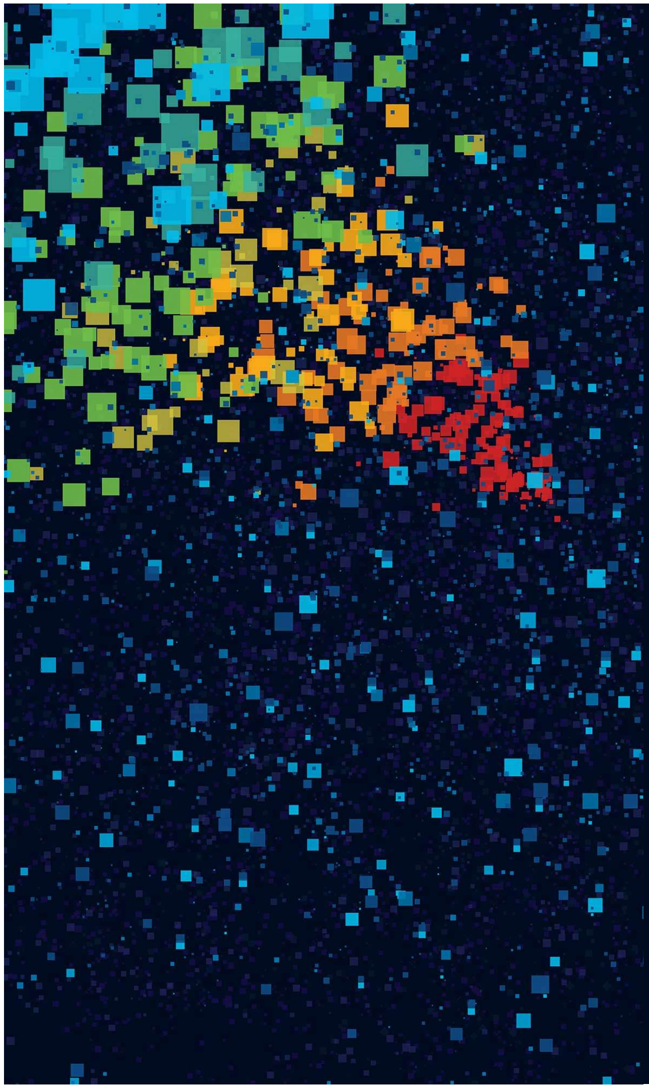
Simplifying Network Service Chaining and Load-balancing with Nexus Elastic Services Redirection

Krithika Krishna Moorthy, Technical Marketing Engineer
Rahul Parameswaran, Leader Technical Marketing
Sreenivas Rao, Product Manager

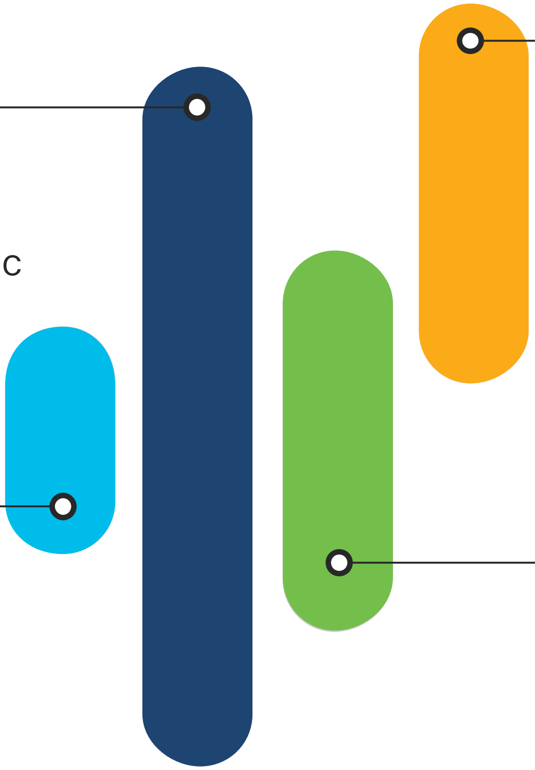


Agenda

- Introduction to ESR
- Intelligent Traffic Director(ITD) Overview
- ITD Use cases
- Enhanced Policy-based Redirect(ePBR) Overview
- ePBR Use cases
- Hardware and Licensing Requirements
- Summary

Elastic Service Redirection

Current Industry Trends



Expanding network functionality

Cloud, multi-tenancy, NFV, Fabric

Solutions

Global growth of internet traffic

More internet users, more devices, more video, faster speed

Service nodes are becoming bottle necks

Not scalable

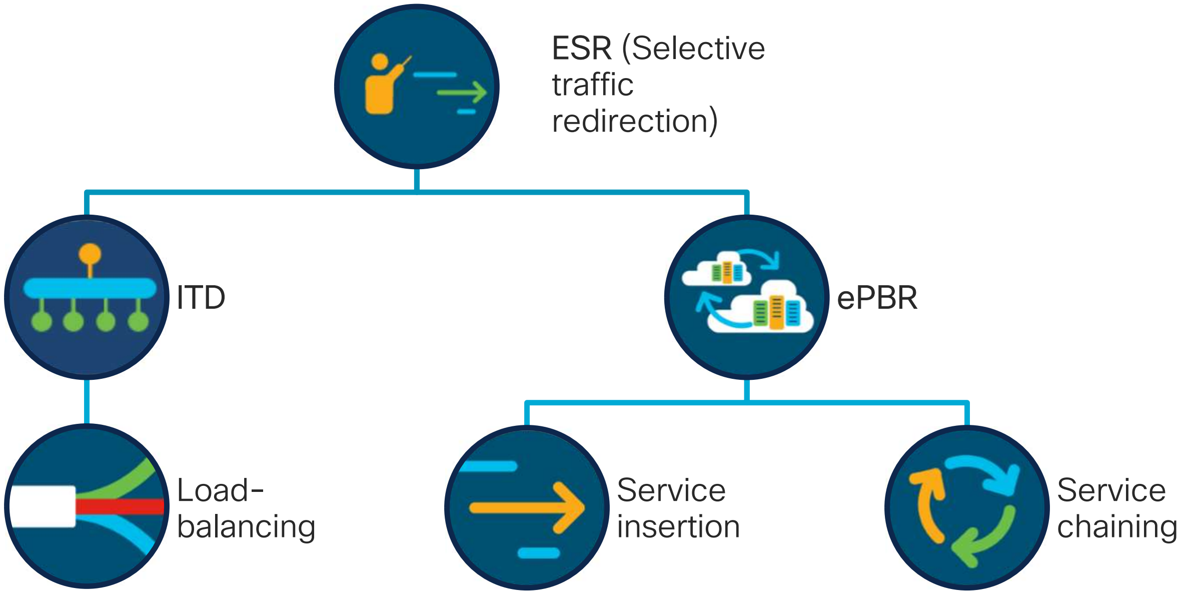
Need extra capacity

Not easy to troubleshoot.

Services deployment

FW, IPS, IDS, Video Caches, WAE, Proxies, TCP optimizers

What is Elastic Services Redirection (ESR) ?



ESR (Selective traffic redirection)

ITD

ePBR

Load-balancing

Service insertion

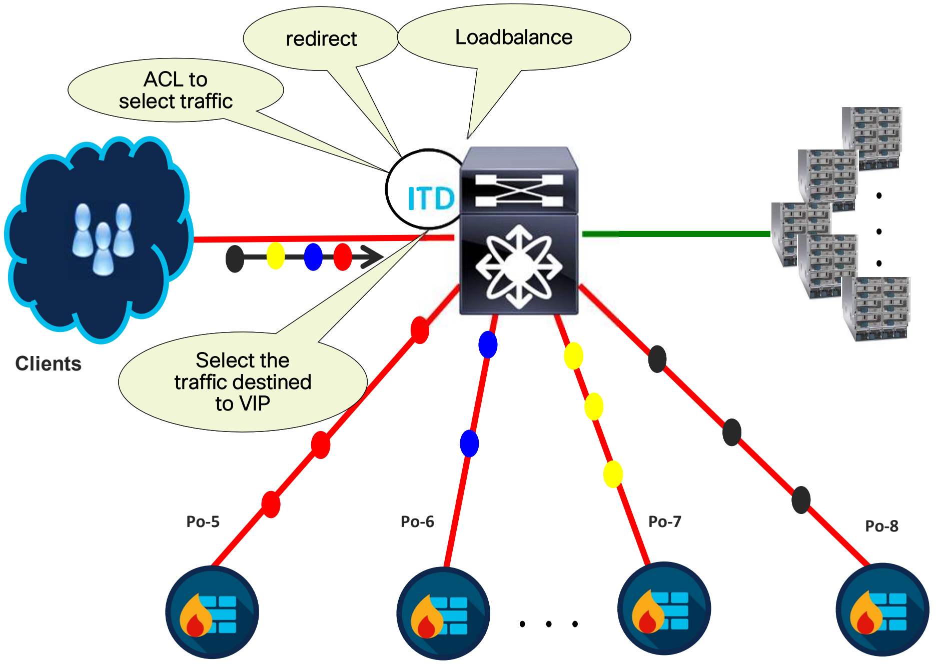
Service chaining

Intelligent Traffic Director (ITD) Overview

ITD ?

- Hardware based Multi-terabit L2/L3/L4 network load-balancing solution at wire-speed
- Addresses growing demand for High-Capacity Traffic Distribution
- ITD eliminates the need to provision and manage another external expensive load-balancer (L2/L3/L4)

ITD: Intelligent Traffic Director



- Maintains IP stickiness & Flow symmetry
- IPv4,IPV6 and VRF Aware
- Health Monitoring
- Resilient and supports failure handling
- High availability, standby support
- Flexible deployment options
- Appliance agnostic

ITD Capabilities

✓ Supports Selective traffic load-balancing
  - ACL based
  - VIP based SLB – (VIP/Protocol/Port)

✓ Supports NAT & PAT (non-DSR mode)

✓ Supports load-balancing using Src/Dst ip and L4 port

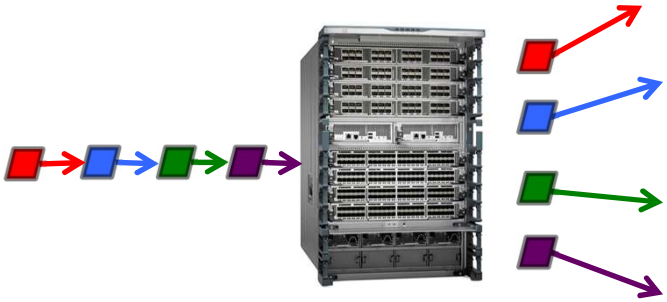
✓ Weighted load-balancing

✓ Flexible Probe options - (ICMP/TCP/UDP/HTTP/DNS/Custom)

✓ Non-disruptively add and delete service nodes + ACL selection

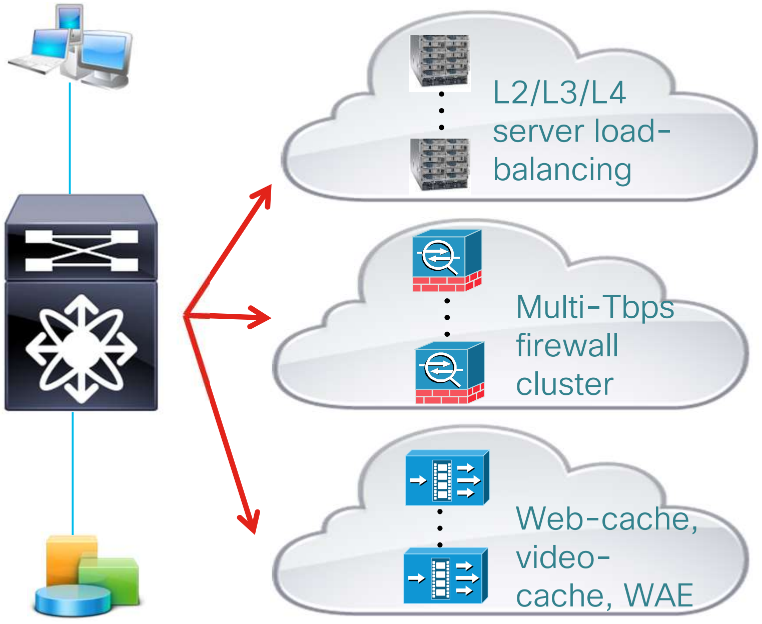
✓ ITD for traffic distributions across inline/bump in the wire deployments

✓ Sub second Convergence



ITD Deployments

Server LB	Server farms, Application servers, Web Servers
Network Function LB	Firewall, IDS, IPS, WAF, L4-L7 Load balancers, Sandwich Deployment
Traffic Steering & Redirection	Web Acceleration Engine (WAE), Web Caches , WCCP replacement


L2/L3/L4 server load-balancing
Multi-Tbps firewall cluster
Web-cache, video-cache, WAE

ITD Solution details

Device group Definition

- Service Nodes / Service Appliances IP
- Probes
- Weights
- Standby (backup nodes)

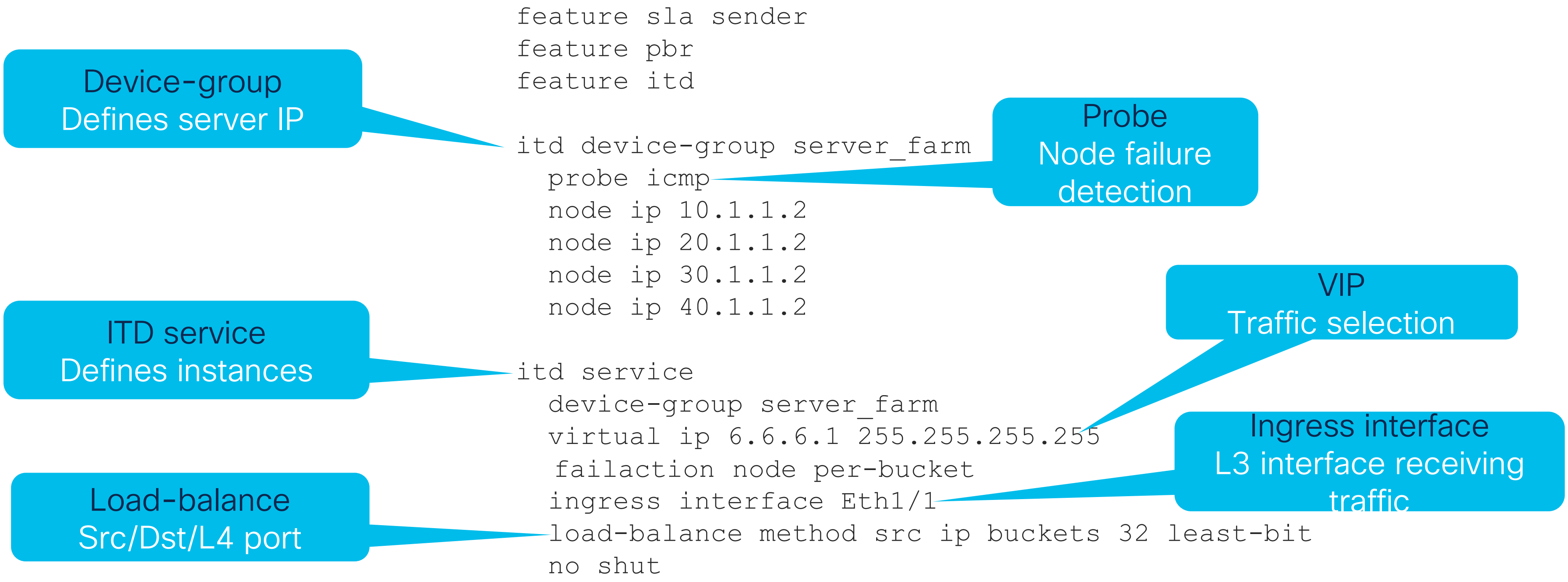
Service Definition

- Attach device-group
- Ingress-interface
- Virtual IP Address
- Traffic Filtering / selection ACL
- Load-balancing options
- Failover options

Service Bring up

- Bring up the ITD service with ‘no shut’ for the policy to be applied on the interface

ITD config example



```
feature sla sender
feature pbr
feature itd

itd device-group server_farm
  probe icmp
  node ip 10.1.1.2
  node ip 20.1.1.2
  node ip 30.1.1.2
  node ip 40.1.1.2

itd service
  device-group server_farm
  virtual ip 6.6.6.1 255.255.255.255
  failaction node per-bucket
  ingress interface Eth1/1
  load-balance method src ip buckets 32 least-bit
  no shut
```

Device-group
Defines server IP

Probe
Node failure detection

ITD service
Defines instances

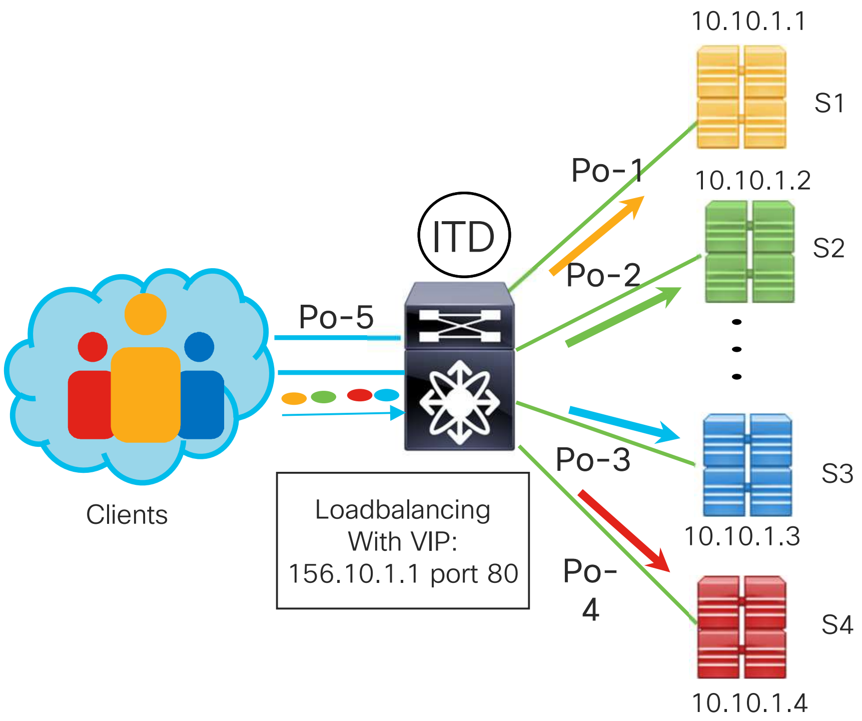
VIP
Traffic selection

Ingress interface
L3 interface receiving traffic

Load-balance
Src/Dst/L4 port

ITD Use cases

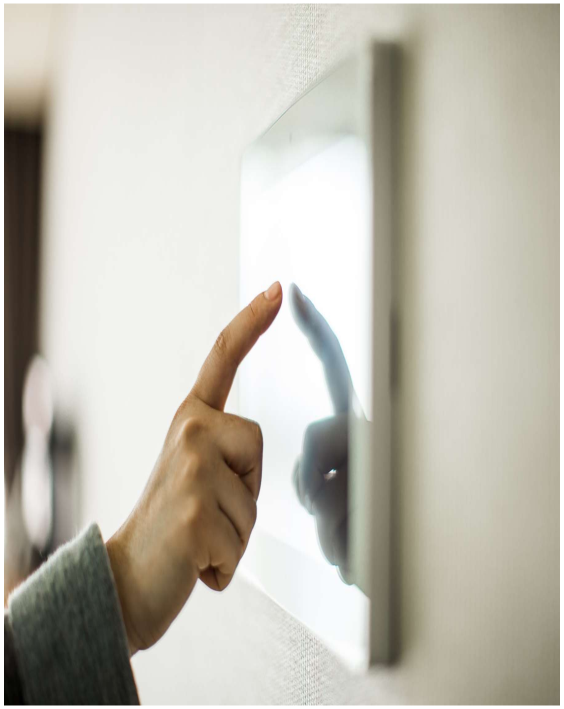
Server Load Balancing(SLB)



- Packets from client redirected and load-balanced across servers using ITD
- All servers configured with a VIP as loopback address
- Server returns the packet to client using Direct Server Return(DSR)
- One-Arm Mode deployment

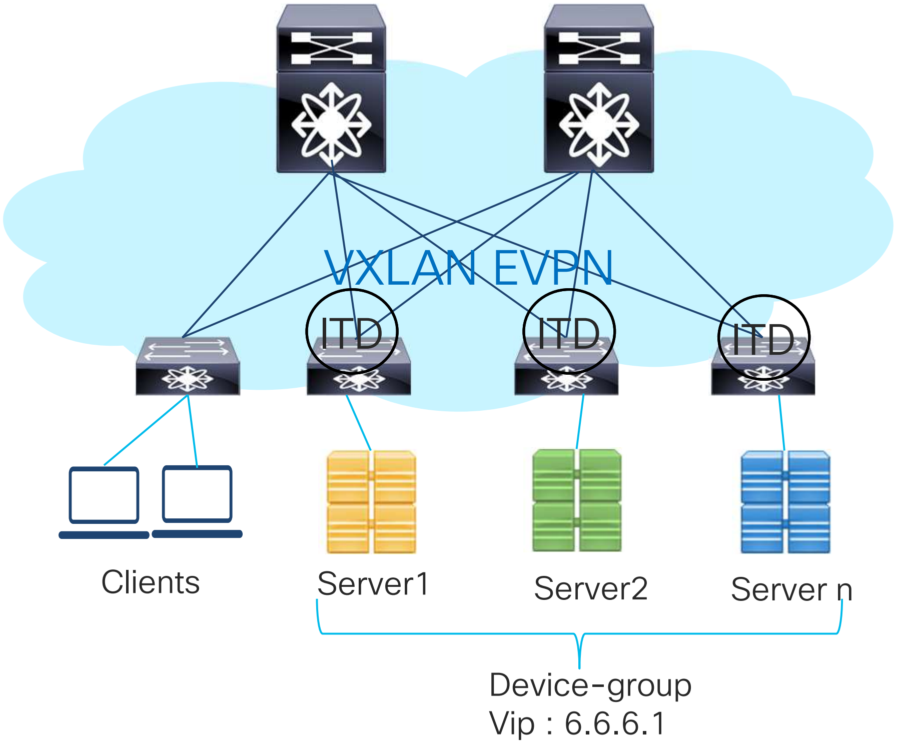
Server Load Balancing(SLB)

Configuration Example



```
feature itd
feature pbr
feature sla sender

itd device-group server_farm
probe tcp port 80
  node ip 10.10.1.1
  node ip 10.10.1.2
  node ip 10.10.1.3
  node ip 10.10.1.4

itd Service-1-IPv4
  device-group server_farm
  virtual ip 156.10.1.1 255.255.255.255 tcp 80
  ingress interface po5
  ingress interface po6
  failaction node per-bucket
  load-balance method src ip buckets 16
  no shut
```

Fabric as a Load-balancer



- ITD over VXLAN enables server load-balancing based on VIP in a VXLAN EVPN fabric

- All servers are configured with a VIP as loopback address

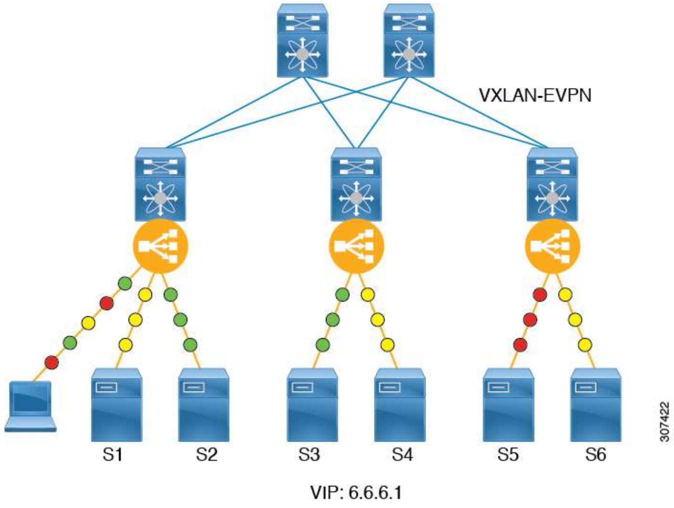
- Server returns the packet to client using Direct Server Return(DSR)

Fabric as a Load-balancer

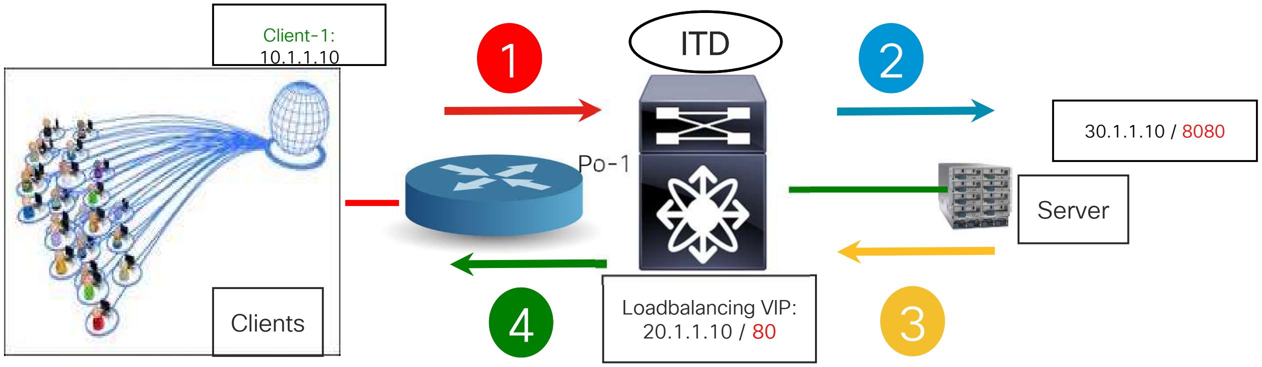
Configuration Example

```
itd device-group DG1
node ip 10.200.1.2
node ip 10.200.2.2
mode hot-standby
node ip 10.200.3.2
node ip 10.200.4.2
node ip 10.200.5.2
mode hot-standby
node ip 10.200.6.2

itd SER1
vrf Org1:vrf1
source-interface loopback9
device-group DG1
virtual ip 6.6.6.1 255.255.255.255
ingress interface Vlan100
load-balance method src ip
no shut
```

L3vni interface



SLB with Destination NAT & PAT



Step	dst-mac	src-mac	src-ip	dst-ip
1	N9K MAC	Router MAC	10.1.1.10	20.1.1.10:80
2	Server MAC	N9K MAC	10.1.1.10	30.1.1.10:8080
3	N9K MAC	Server MAC	30.1.1.10:8080	10.1.1.10
4	Router MAC	N9K MAC	20.1.1.10:80	10.1.1.10

- ITD NAT eliminates the need to configure a loopback on server for DSR
- In forward flow from Client to server, the N9k translates the DIP and port from VIP to real IP/Port of the server
- In reverse flow from Server back to client, N9k translates the SIP from server IP/Port to VIP/Port

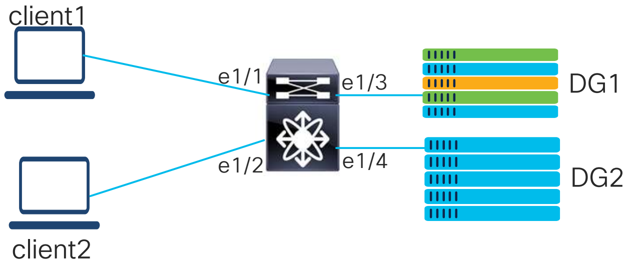
SLB with Destination NAT & PAT

Configuration Example

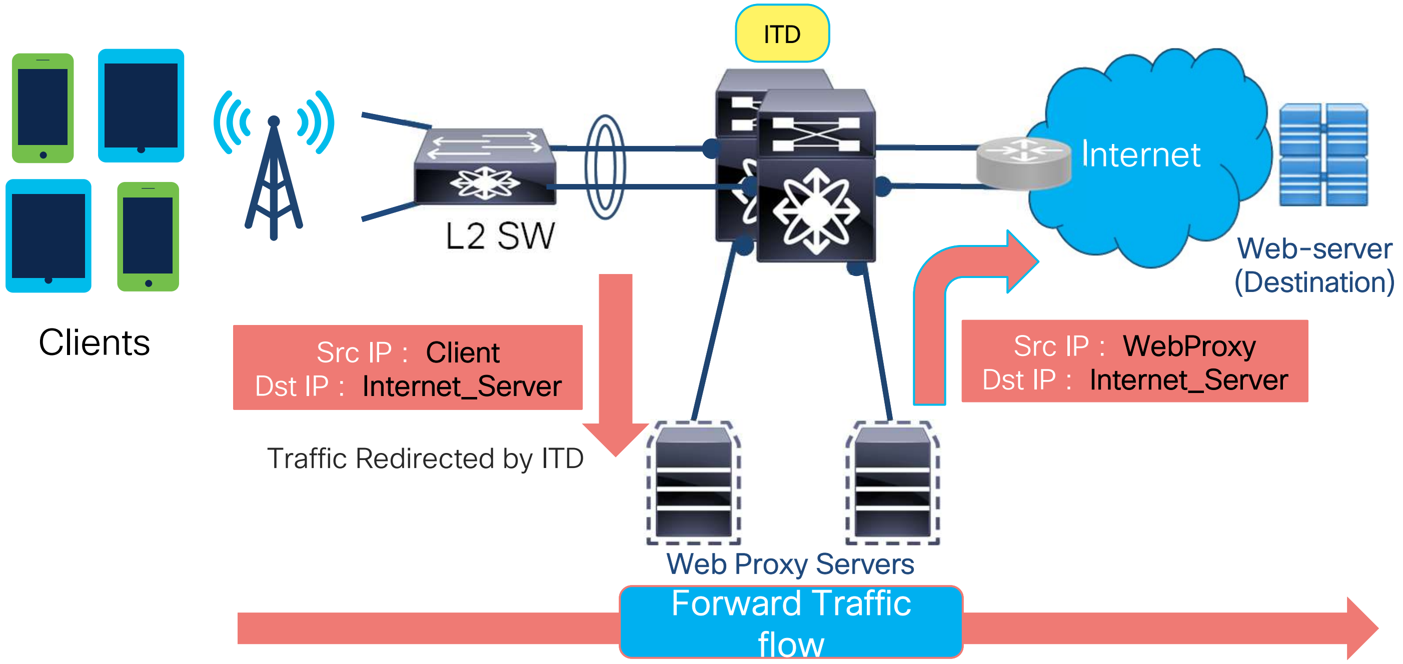
ITD NAT requires NAT tcam and feature ”NAT” to be enabled

```
hardware access-list tcam region nat 2048
feature nat
feature itd
feature sla sender

interface eth1/1-2
ip nat outside
interface e1/3-4
ip nat inside

itd device-group DG1
 probe icmp frequency 2 timeout 1
node ip 8.8.1.2
node ip 9.9.1.2
port 1000

itd device-group DG2
probe icmp
node ip 10.10.1.2
port 1000
node ip 11.11.1.3
port 2000
```

```
itd SER1
virtual ip 6.6.1.1 255.255.255.255 tcp 80 advertise enable device-group DG1
virtual ip 6.6.1.2 255.255.255.255 tcp 90 advertise enable device-group DG2
ingress interface e1/1
Ingress interface e1/2
nat destination
failaction node per-bucket
load-balance method src ip buckets 32
no shut
```

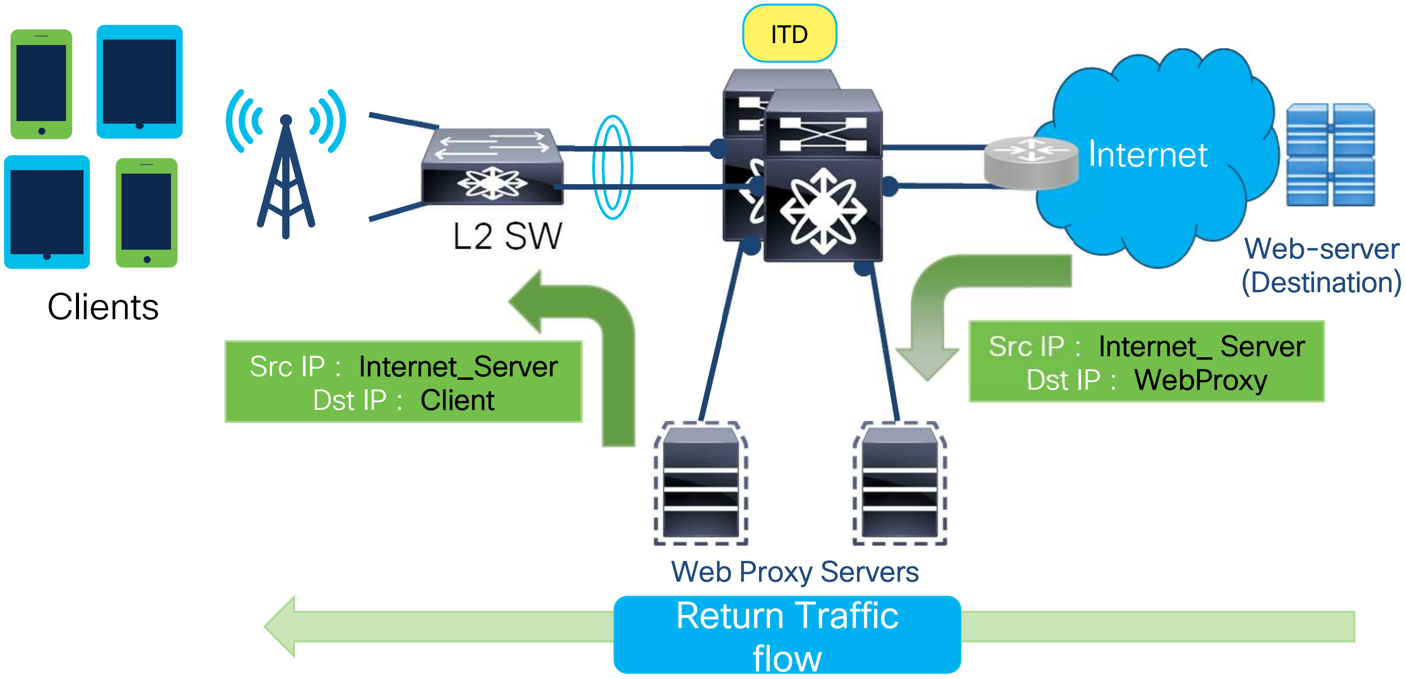


Web-Proxy Deployment



ITD redirects using Include-ACL and load-balances the packets across the Web-Proxy servers

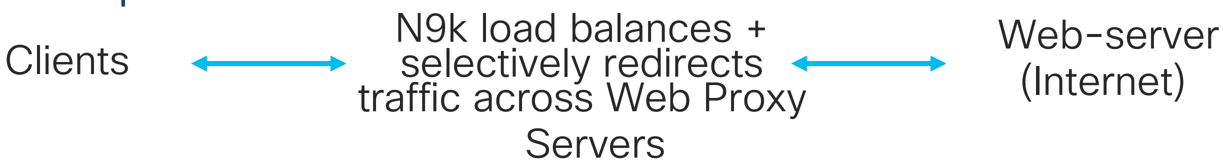
Web-Proxy Deployment (cont.)



Packets are being forwarded normally(no redirection) on the Nexus Switches.

Web-Proxy Deployment

Configuration Example



Clients

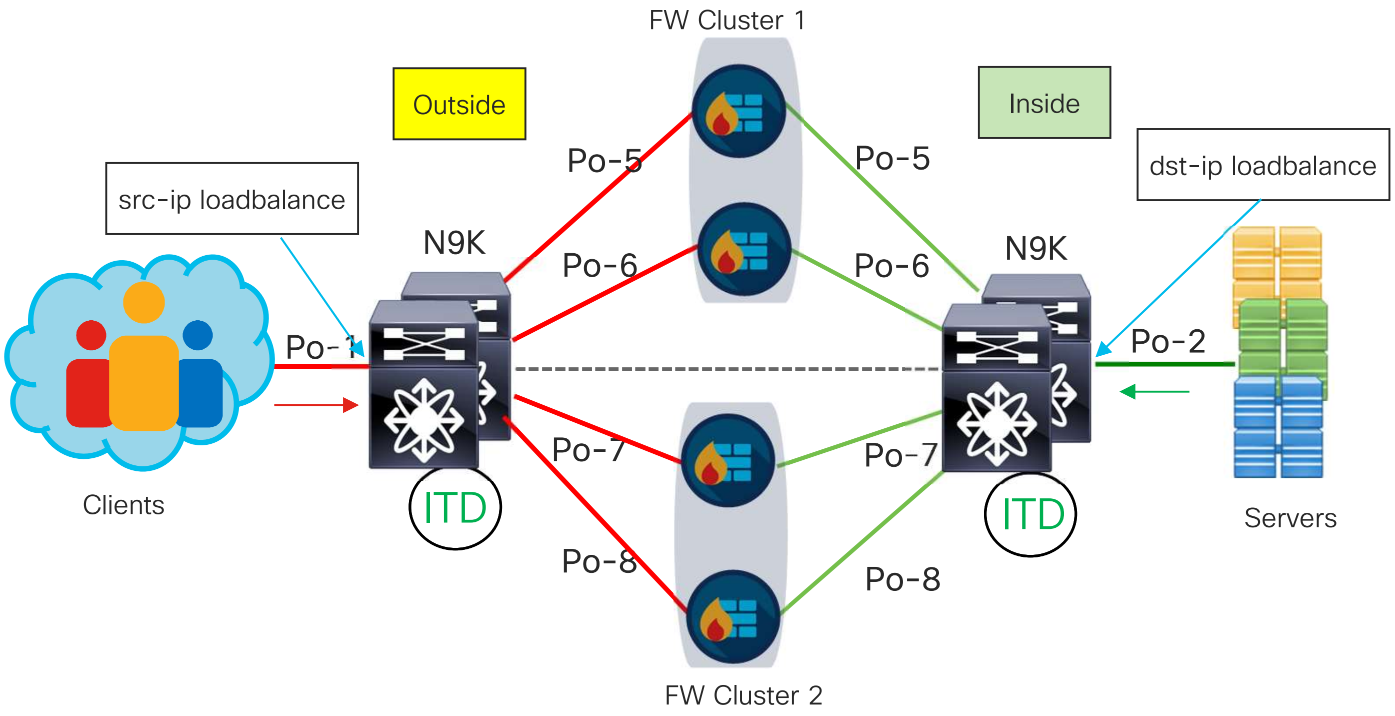
N9k load balances + selectively redirects traffic across Web Proxy Servers

Web-server (Internet)

```
feature itd
feature pbr
feature sla sender

ip access-list itd_exclude_ACL
  ! Exclude private IP address
  10 permit ip any 10.0.0.0/8
  20 permit ip any 192.168.0.0/16
  30 permit ip any 172.16.0.0/12

ip access-list internet-acl
  10 permit ip any any tcp 80
  20 permit ip any any tcp 443
```

```
itd device-group Web_Proxy_Servers
  probe icmp
  node ip 10.1.50.1
  node ip 10.1.50.2

itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_ACL
  access-list internet-acl
  ingress interface Vlan 10
  failaction bucket distribute
  load-balance method src ip
  no shutdown
```

Sandwich Mode Deployment

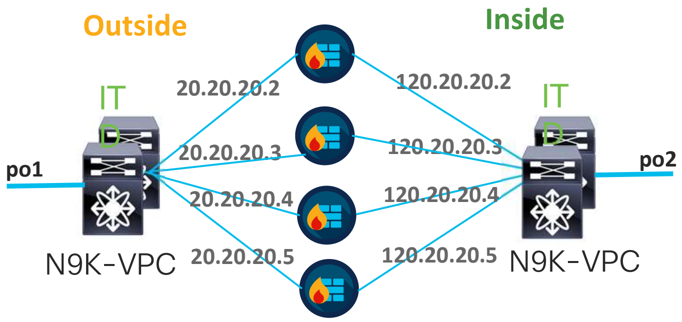


- The sandwich deployment mode provides Symmetric handling of traffic.
- Forward and reverse traffic between the client and the server flows through the same appliance.
- Appliance clustering capability

Sandwich Deployment

Configuration Example

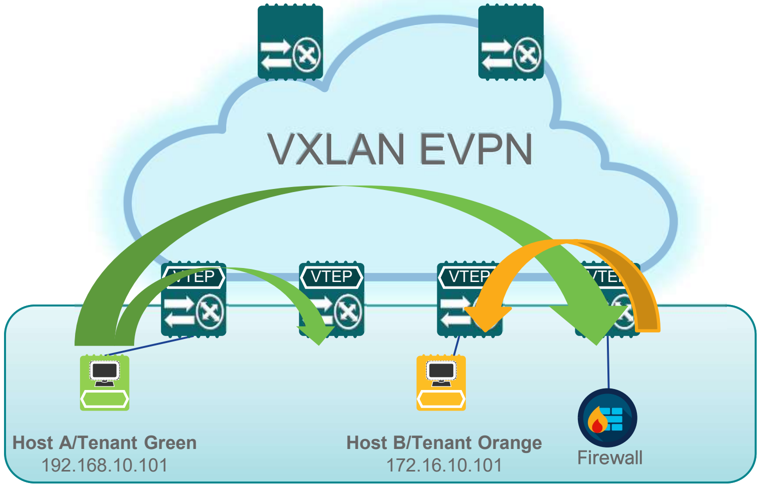
```
itd device-group FW-INSPECT
probe icmp
node ip 20.20.20.2
cluster 1
node ip 20.20.20.3
cluster 1
node ip 20.20.20.4
cluster 2
node ip 20.20.20.5
cluster 2

itd WebTraffic
Device-group FW-INSPECT
ingress interface po1
failaction bucket distribute
load-balance method src ip buckets
64
no shut
```



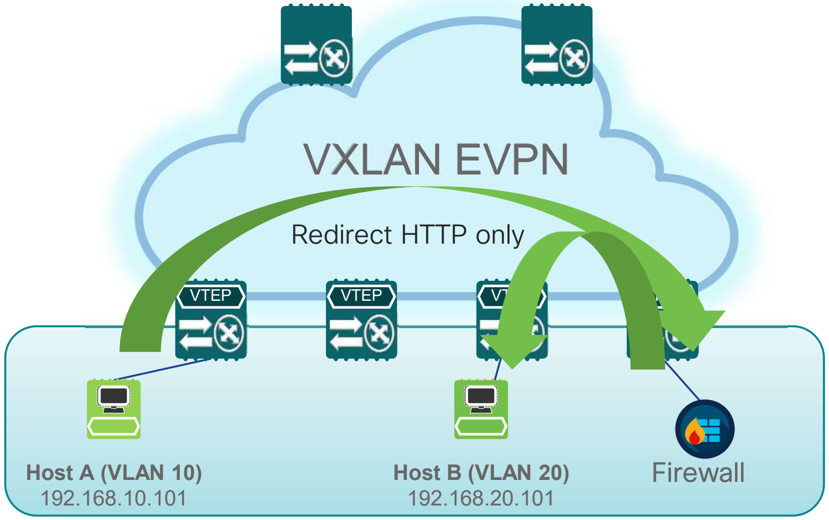
```
itd device-group FW-INSPECT
probe icmp
node ip  20.20.20.2
cluster 1
node ip 20.20.20.3
cluster 1
node ip 20.20.20.4
cluster 2
node ip 20.20.20.5
cluster 2

itd WebTraffic
Device-group FW-INSPECT
ingress interface po2
failaction bucket distribute
load-balance method dst ip buckets
64
no shut
```

Enhanced Policy-based Redirect (ePBR) Overview

How is Service Chaining Done today?



Routing rules reflect path via service devices

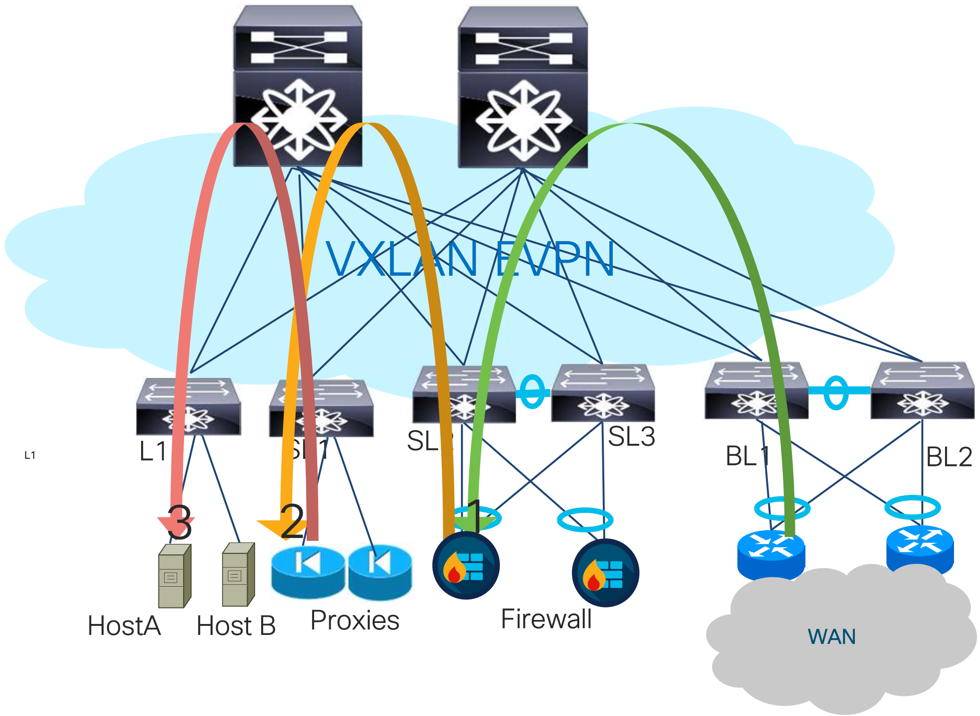


Selective Traffic Redirect using Policy Based Routing

What are the challenges with existing options?

- Service nodes becoming bottlenecks
- Static PBR policies complex to maintain
- Service redirection across multiple service nodes is complex to configure
- Options to load-balance and redirect missing
- Limited ability to monitor device health and configuring fail action based on device availability is missing
- Need to account for forward and return traffic to maintain symmetry

Enhanced Policy-based Redirect(ePBR)



- Hardware based Multi-terabit service-chaining solution
- EPBR provides the ability to
  - ✓ Onboard services
  - ✓ Create service-chains
  - ✓ Load-share with selective redirection
- Health monitoring and node failover
- Flexible deployments options
  - ✓ Fabric based with VXLAN EVPN
  - ✓ Traditional centralized gateway deployment

ePBR Capabilities

Simplified service chain creation

- Simplified device onboarding
- Granular multi-level service policy creation

Optimized utilization of service node

- Selective traffic redirection
- L3/L4 redirect , exclude and drop options

Ability to scale

- Through symmetric load-balancing along with chaining

Health monitoring & flexible failover

- Probes - ICMP/TCP/UDP/HTTP/DNS/Custom
- Failover - Forward / Bypass / Drop

Non-disruptive in-service config updates

- Add/delete/modify service, policy and match ACL selection

Line Rate traffic forwarding

- No impact to throughput & performance
- No increased latency

ePBR Solution details

Onboard Service Appliance

- Service IP address
- Forward and reverse attached interface (single/dual arm)
- Probes
- VRF membership
- Additional service end-points for creating appliance cluster

Define traffic redirect Policy

- Traffic Filtering or selection ACL
- Service-chain creation
- Load-balancing options(src/dst and buckets)
- Failover options (forward/bypass/drop)

Apply the ePBR Policy on relevant interfaces

- Apply policy on ingress interface where chaining needs to start
- VXLAN – Apply on L3 VNI interfaces on service leaf
- Apply policy with “reverse” keyword to maintain flow symmetry

ePBR config example

```
epbr service FIREWALL_CLUSTER_A
   probe icmp source-interface loopback10
   vrf TENANT_A
   service-endpoint ip 172.16.1.200 interface VLAN100
           reverse ip 172.16.2.200 interface VLAN101
   service-endpoint ip 172.16.1.201 interface VLAN100
           reverse ip 172.16.2.201 interface VLAN101

epbr policy Tenant_A-Redirect
   match ip address WEB
   load-balance method src-ip
    10 set service FIREWALL fail-action drop
    20 set service TCP_Optimizer fail-action bypass
   match ip address APP
   load-balance method src-ip
    10 set service FIREWALL fail-action drop

interface vlan 2010
  !L3 VNI SVI
  epbr ip policy Tenant_A-Redirect
  epbr ip policy Tenant_A-Redirect reverse
```

Forward arm

Reverse arm

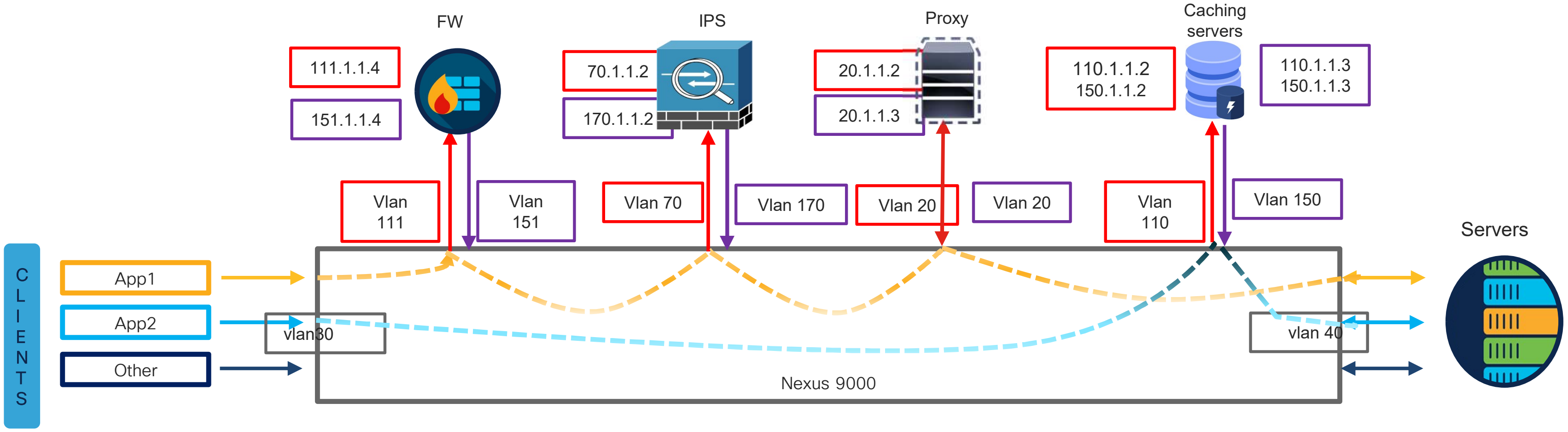
Active /Active firewall pair

ACL matches web traffic

EPBR auto generates reverse policies for return traffic

ePBR Use cases

Service chaining in Traditional deployments



- App1 traffic : firewall → IPS → Proxy
- App2 traffic : Load-balanced directly among the caching servers

Service chaining in Traditional deployments

Configuration Example

Step 1: Onboard the appliances

```
epbr service firewall
  service-end-point ip 111.1.1.4 interface Vlan111
     probe icmp source-interface loopback0
    reverse ip 151.1.1.4 interface Vlan151
       probe icmp source-interface loopback1

epbr service ips
  service-end-point ip 70.1.1.2 interface Vlan70
     probe udp 45000
    reverse ip 170.1.1.2 interface Vlan170
       probe udp 45001

epbr service proxy
  service-interface Vlan20
   probe http get index.html
  service-end-point ip 20.1.1.2
    reverse ip 20.1.1.3
```

```
epbr service caching_servers
! traffic will be load-balanced between the
servers
! server1
  service-end-point ip 110.1.1.2 interface Vlan110
     probe icmp source-interface loopback0
    reverse ip 150.1.1.2 interface Vlan150
       probe icmp source-interface loopback1
! server2
  service-end-point ip 110.1.1.3 interface Vlan110
     probe icmp source-interface loopback0
    reverse ip 150.1.1.3 interface Vlan150
       probe icmp source-interface loopback1
```

Service chaining in Traditional deployments

Configuration Example(cont.)

Step 2: Create traffic selection rules

```
ip access-list app1
        10 permit tcp 172.16.10.0/24 eq 7800 any
        20 permit tcp 192.168.20.0/24 eq 7800 any
ip access-list app2
        10 permit tcp 172.16.10.0/24 any eq www
        20 permit tcp 192.168.20.0/24 any eq www
```

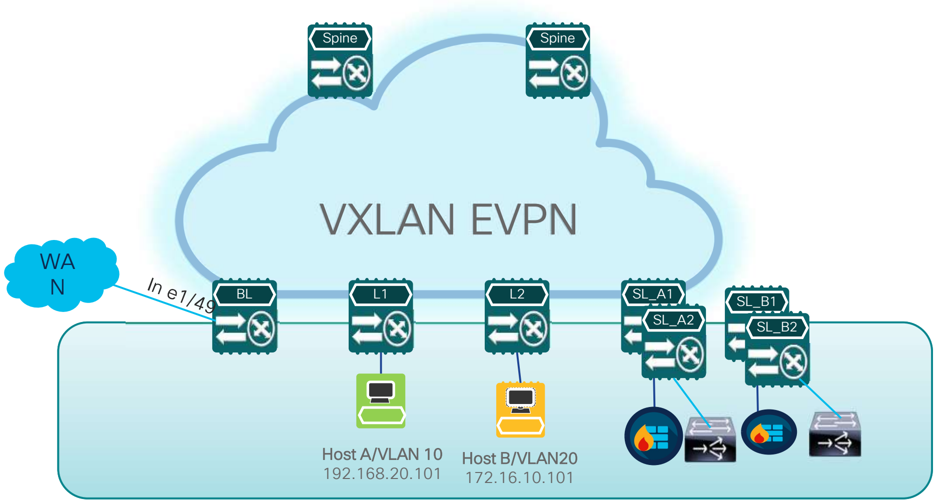
Step 3: Define ePBR traffic redirect policy

```
epbr policy redirect_and_loadbalance
  statistics
  match ip address app1
    ! Traffic matching app1 takes FW→IPS→Proxy chain
    10 set service firewall fail-action drop
    20 set service ips fail-action bypass
    30 set service proxy fail-action forward
  match ip address app2
    ! Traffic matching app2 is load-balanced across
caching servers
    load-balance buckets 8 method src-ip
    10 set service caching_servers
```

Step 4: Apply the ePBR Policy on relevant interfaces

```
interface Vlan30
  !forward policy applied to ingress interface facing
clients
  no shutdown
  ip address 30.1.1.1/24
  ipv6 address 2030::1/24
  epbr ip policy redirect_and_loadbalance

interface Vlan40
! Reverse policy applied to egress interface facing
server farm for reverse flow
  no shutdown
  ip address 40.1.1.1/24
  ipv6 address 2040::1/24
   epbr ip policy redirect_and_loadbalance reverse
```

Service chaining in VXLAN Fabric

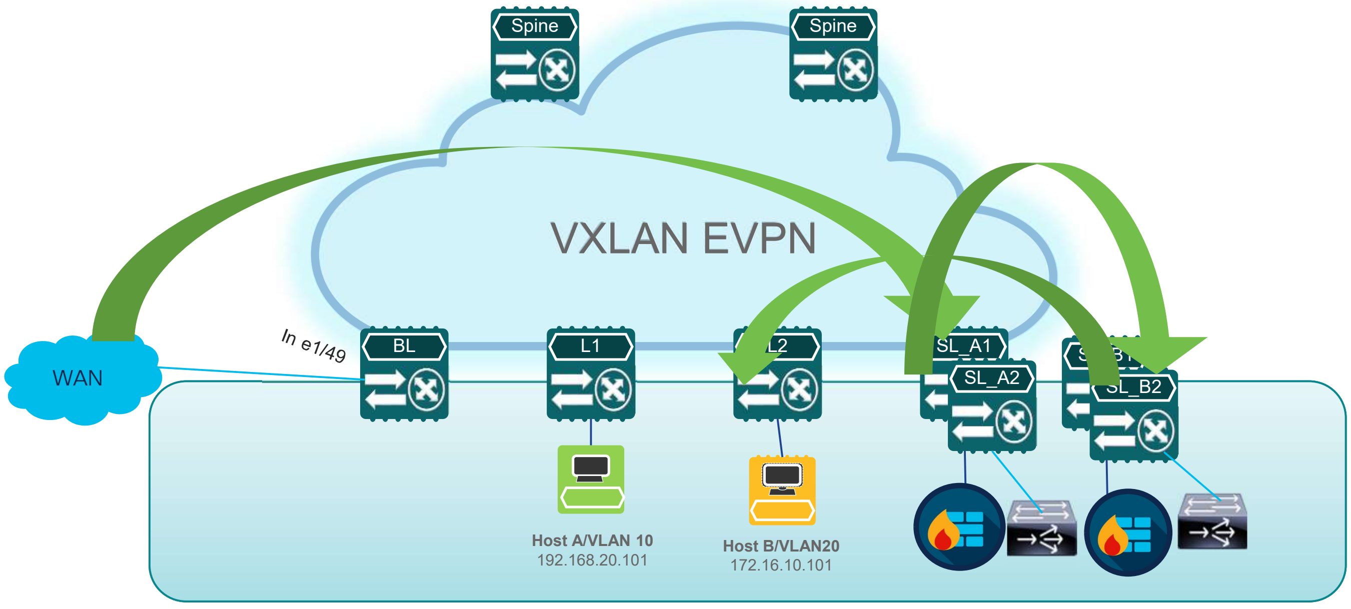


Service Nodes in Active/Standby Mode

Requirements

- Firewall and Load-balancer provisioned as High availability Pair in Active/Standby fashion
- Identify failure of active services and switchover traffic to standby on event of failure
- We need selective segregation on the traffic from WAN based on different match criteria
- All other traffic goes through via routing table rules.
- Fail-action rules
  - If FW Cluster is down , drop.
- Symmetry must be maintained for return traffic.

Selective traffic redirection across Active/Standby Service Appliances



FW_inside in VLAN 101
FW_outside in VLAN 102
LB in VLAN 200
FW and LB in same VRF as Hosts

Service Nodes in Active/Standby Mode

Selective traffic redirection across Active/Standby Service Appliances

Configuration Example on service leaf

Step 1: Onboard appliances

```
epbr service firewall
  vrf tenant_a
  service-end-point ip 10.1.1.2 interface Vlan10
     probe icmp frequency 4 timeout 2 source-
interface loopback9
    reverse ip 50.1.1.2 interface Vlan50
       probe icmp frequency 4 timeout 2 source-
interface loopback10

epbr service load-balancer
  service-interface Vlan20
  vrf tenant_a
   probe http get index.html source-interface
loopback9
  service-end-point ip 20.1.1.2
    reverse ip 20.1.1.2
```

Step 2: Create traffic selection rules

```
ip access-list custom_app
        10 permit tcp 172.16.10.0/24 eq 7800 any
        20 permit tcp 192.168.20.0/24 eq 7800 any

ip access-list web
        10 permit tcp 172.16.10.0/24 any eq www
        20 permit tcp 192.168.20.0/24 any eq www
```

Selective traffic redirection across Active/Standby Service Appliances

Configuration Example(cont.)

Step 3: Define ePBR traffic redirect policy

```
epbr policy service_chain
  statistics
  match ip address custom_app
    load-balance buckets 4 method src-ip
    10 set service firewall fail-action drop
    20 set service load-balancer fail-action bypass
  match ip address web
    load-balance buckets 2 method src-ip
    10 set service firewall fail-action drop
```

Step 4: Apply the ePBR Policy on L3vni interfaces for forward and return traffic

```
interface Vlan100
  ! L3 VNI SVI
  vrf member tenant_a
    ip forward
  ipv6 forward
  epbr ip policy service_chain
  epbr ip policy service_chain reverse
```

Verification using show command

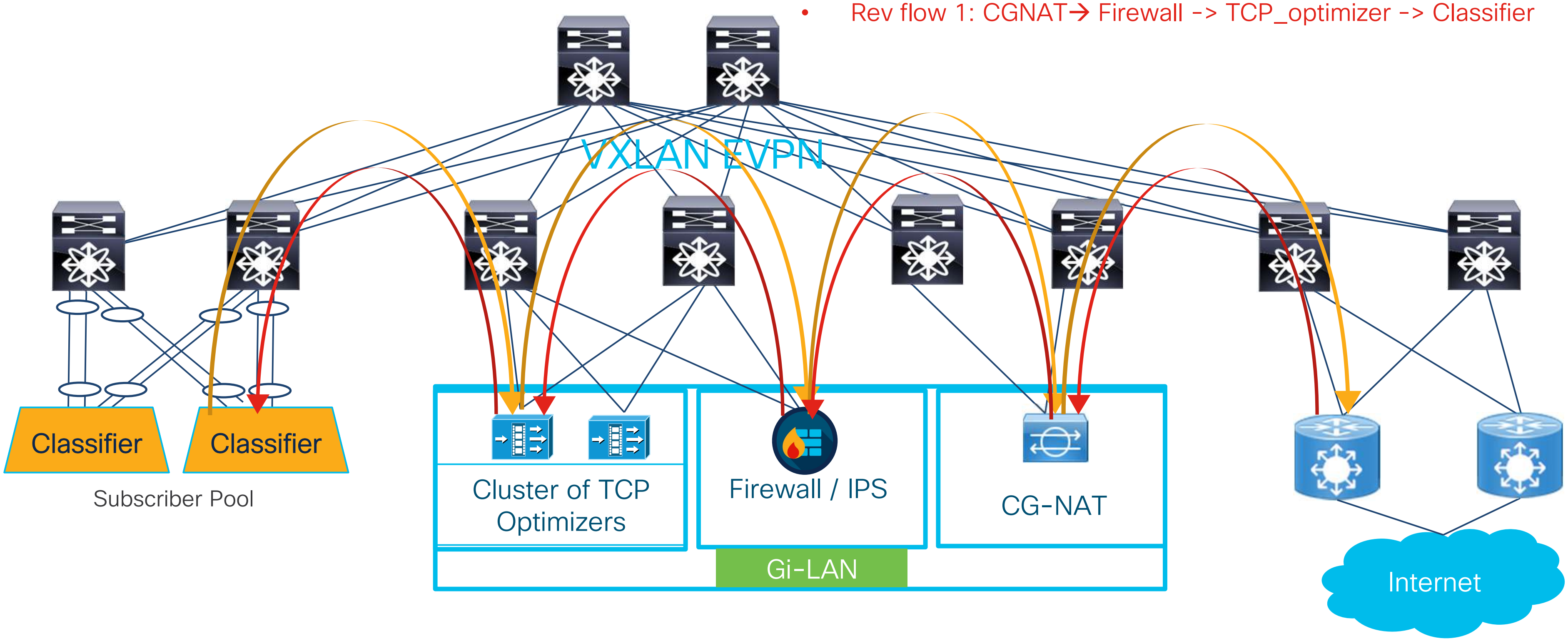
```
sh epbr policy service_chain

Policy-map : service_chain
  Match clause:
    ip address (access-lists): custom_app
  Service chain:
    service firewall, sequence 10, fail-action Drop
      IP 10.1.1.2 track 1 [UP]
    service load-balancer, sequence 20, fail-action Bypass
      IP 20.1.1.2 track 2 [UP]
  Match clause:
    ip address (access-lists): web
  Service chain:
    service firewall, sequence 10, fail-action Drop
      IP 10.1.1.2 track 1 [UP]
  Policy Interfaces:
    Eth1/49
```

Note: Use reverse keyword in show command to view reverse service chain

ePBR for 5G deployments !!

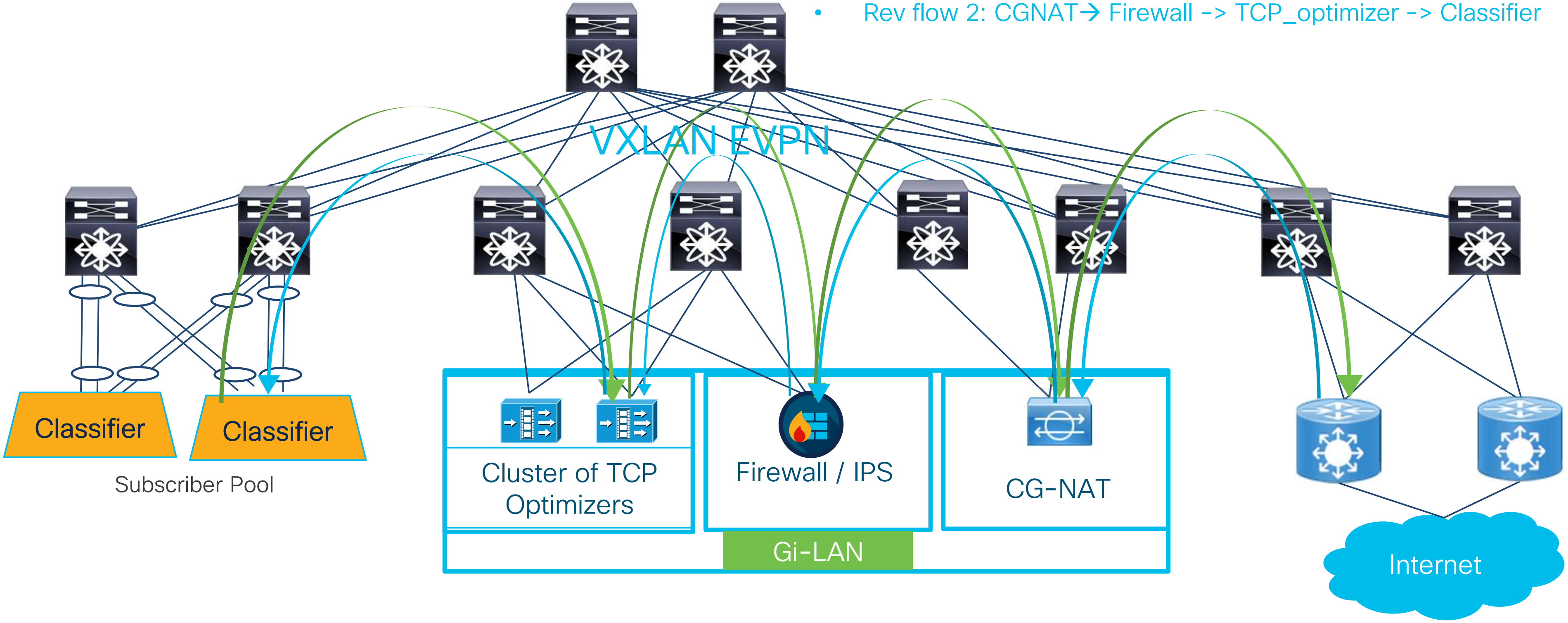
Service chaining & load-balancing across Telco DC

- Fwd flow1 : Classifier -> TCP_optimizer -> Firewall -> CG-NAT
- Rev flow 1: CGNAT→ Firewall -> TCP_optimizer -> Classifier



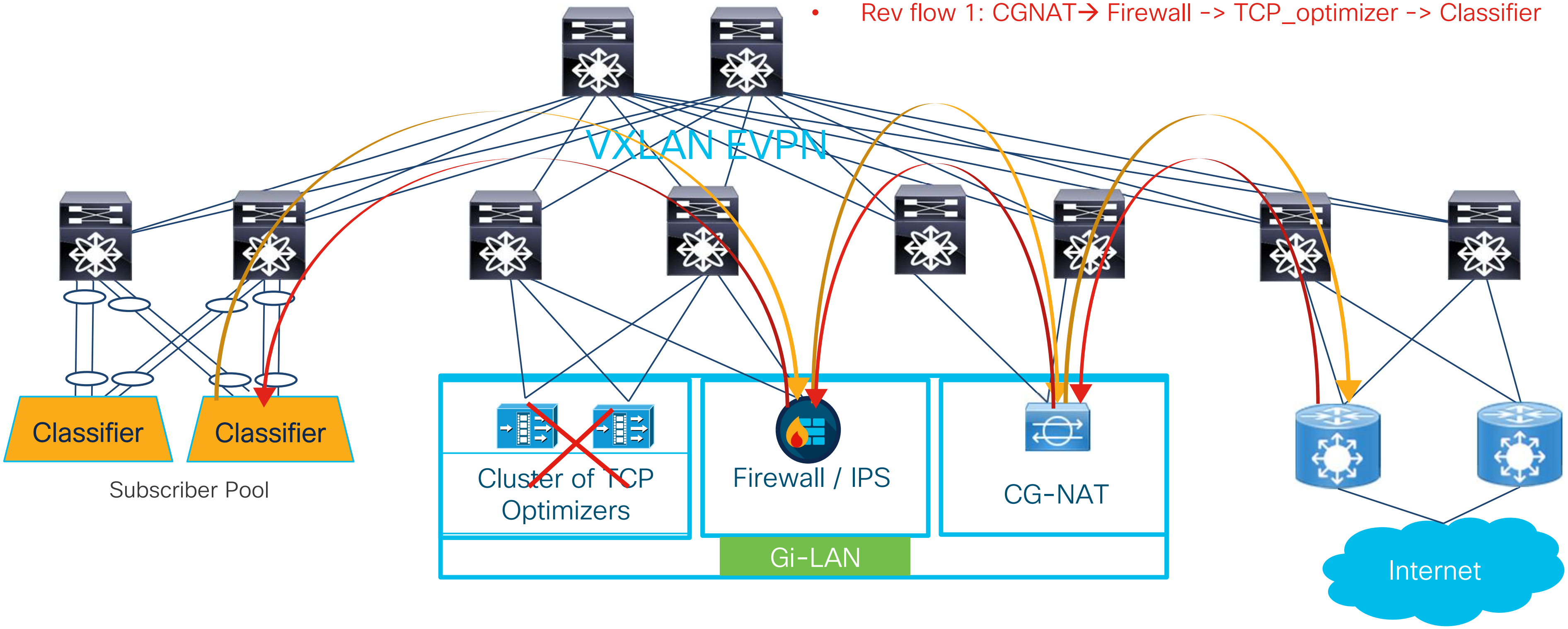
Symmetric PBR across different flows

- Fwd flow2 : Classifier -> TCP_optimizer -> Firewall -> CG-NAT
- Rev flow 2: CGNAT→ Firewall -> TCP_optimizer -> Classifier

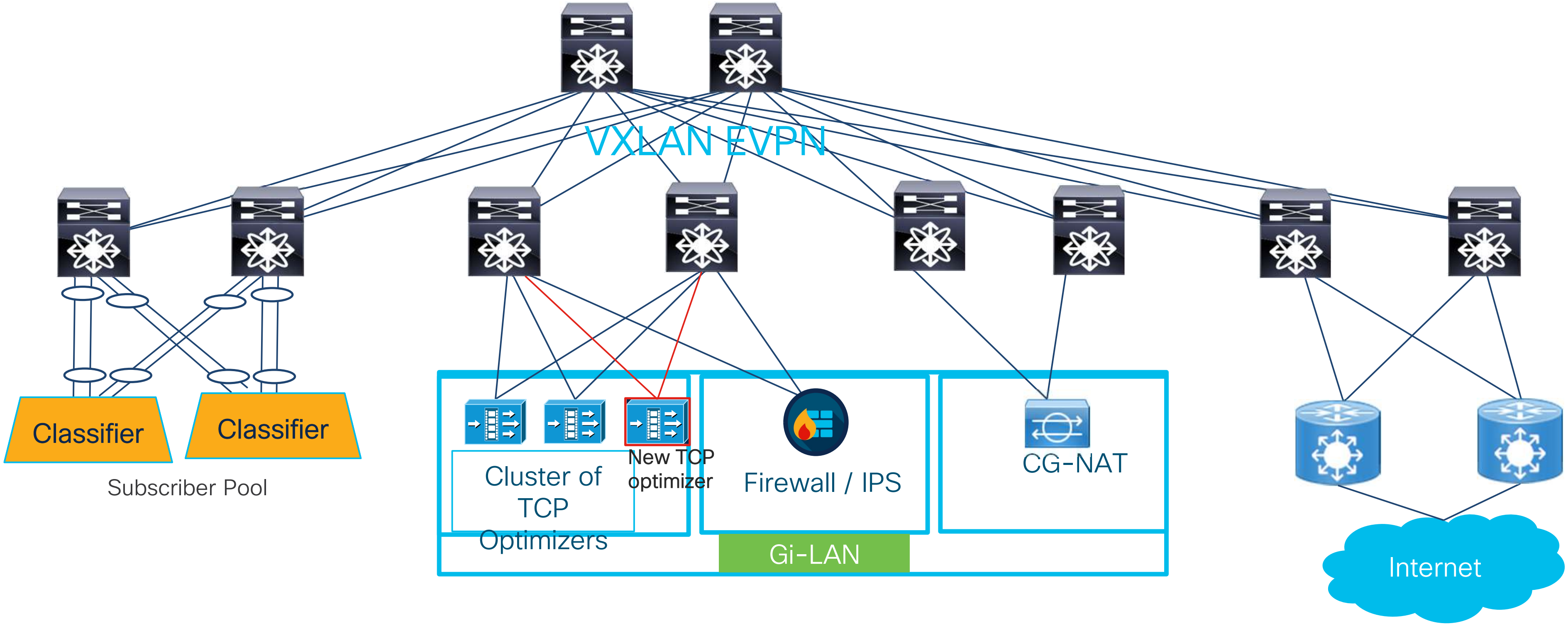


Bypass failed service node

- Fwd flow1 : Classifier -> TCP_optimizer -> Firewall -> CG-NAT
- Rev flow 1: CGNAT→ Firewall -> TCP_optimizer -> Classifier



Non-disruptive simplified expansion



ePBR for 5G deployments

Configuration Example

Step 1: Onboard the appliances

```
epbr service firewall
  service-end-point ip 111.1.1.4 interface Vlan111
     probe icmp source-interface loopback0
    reverse ip 151.1.1.4 interface Vlan151
       probe icmp source-interface loopback1

epbr service cg_nat
  service-interface Vlan20
   probe http get index.html
  service-end-point ip 20.1.1.2
    reverse ip 20.1.1.3
```

```
epbr service tcp_optimizers
! traffic will be load-balanced between the
optimizers
! optimizer1
  service-end-point ip 110.1.1.2 interface Vlan110
     probe icmp source-interface loopback0
    reverse ip 150.1.1.2 interface Vlan150
       probe icmp source-interface loopback1
! optimizer2
  service-end-point ip 110.1.1.3 interface Vlan110
     probe icmp source-interface loopback0
    reverse ip 150.1.1.3 interface Vlan150
       probe icmp source-interface loopback1
```

ePBR for 5G deployments

Configuration Example(cont.)

Step 2: Create traffic selection rules

```
ip access-list app1
        10 permit tcp 172.16.10.0/24 eq 7800 any
        20 permit tcp 192.168.20.0/24 eq 7800 any
```

Step 3: Define ePBR traffic redirect policy

```
epbr policy servicechain_and_loadbalance
  statistics
  match ip address app1
    ! TCP optimizer→firewall→cg_nat chain
    10 set service tcp_optimizers fail-action bypass
    20 set service firewall fail-action drop
    30 set service cg_nat fail-action drop
```

Step 4: Apply the ePBR Policy on relevant interfaces

```
interface Vlan30
  !forward policy applied to ingress interface facing
classifier
  no shutdown
  ip address 30.1.1.1/24
  ipv6 address 2030::1/24
  epbr ip policy servicechain_and_loadbalance

interface Vlan40
! Reverse policy applied to egress interface facing WAN
for reverse flow
  no shutdown
  ip address 40.1.1.1/24
  ipv6 address 2040::1/24
   epbr ip policy servicechain_and_loadbalance reverse

interface vlan100
! L3vni interface on service leafs
ip forward
no ip redirect
epbr ip policy servicechain_and_loadbalance
epbr ip policy servicechain_and_loadbalance reverse
```

Hardware, Software and Licensing

ESR Hardware Support

Nexus 9500 Series with EX, FX and GX line cards

Nexus 9300 EX/FX/FX2/FX3/GX Series

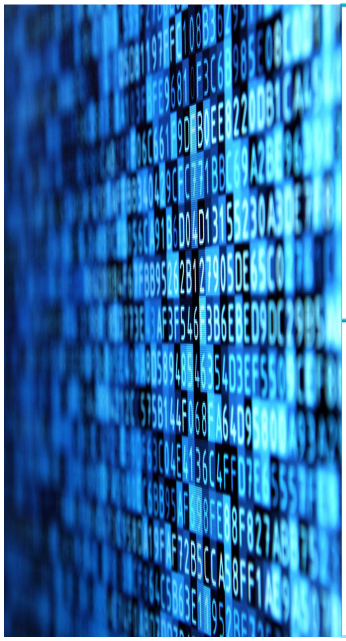
Nexus 3600 & 9500 R Series*

Cisco Nexus 9000 Series

From Cisco’s Data Center Portfolio

*ePBR support in upcoming release

ESR Software and Licensing requirements



ITD	NX-OS 7.0(3)I1(2)
	Essentials Package
ePBR	NX-OS 9.3(5)
	Advantage Package

Summary

ESR Benefits

Scalability

Multi-Terabits Line Rate solutions

No CPU overhead

Scales to large number of Service Nodes

High Availability

Health Monitoring of servers/appliances

Automatic Failure Handling

N + M redundancy

OPEX Savings

Simplified provisioning & Ease of deployment

Significant reduction of Configuration Complexity

Programmable (REST, Netconf)

CAPEX Savings

Moving away from specialized, dedicated, expensive HW

Additional Cost savings from Wiring, Power, Rackspace