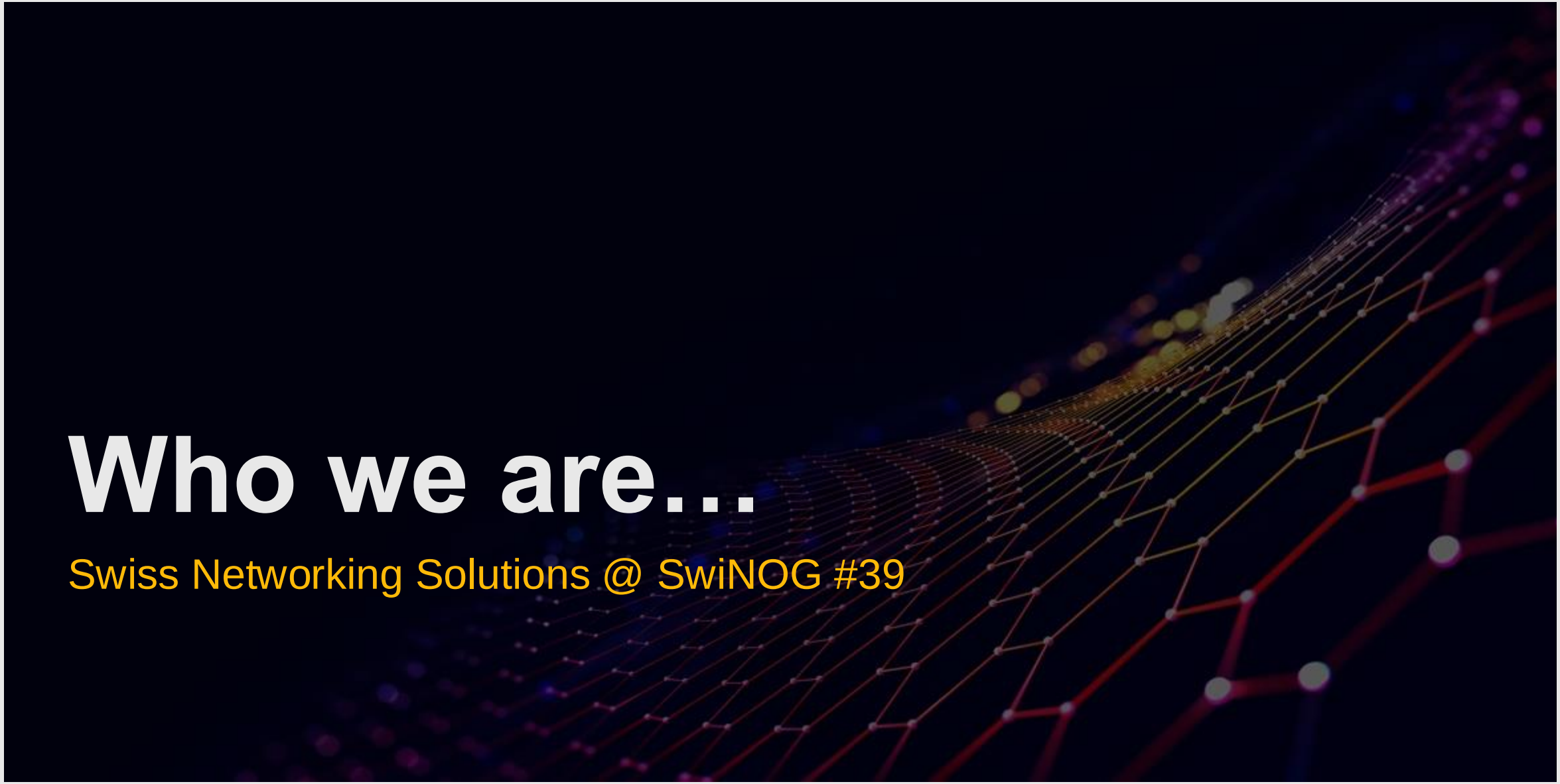
Our Path to Multi-Node High Availability

Overcoming Challenges with SRX Clusters


Who we are…
Swiss Networking Solutions @ SwiNOG #39

Swiss Networking Solutions

Based in Zug

System-, Network- and Security-Engineering Teams

“sister”-company of Telecom26

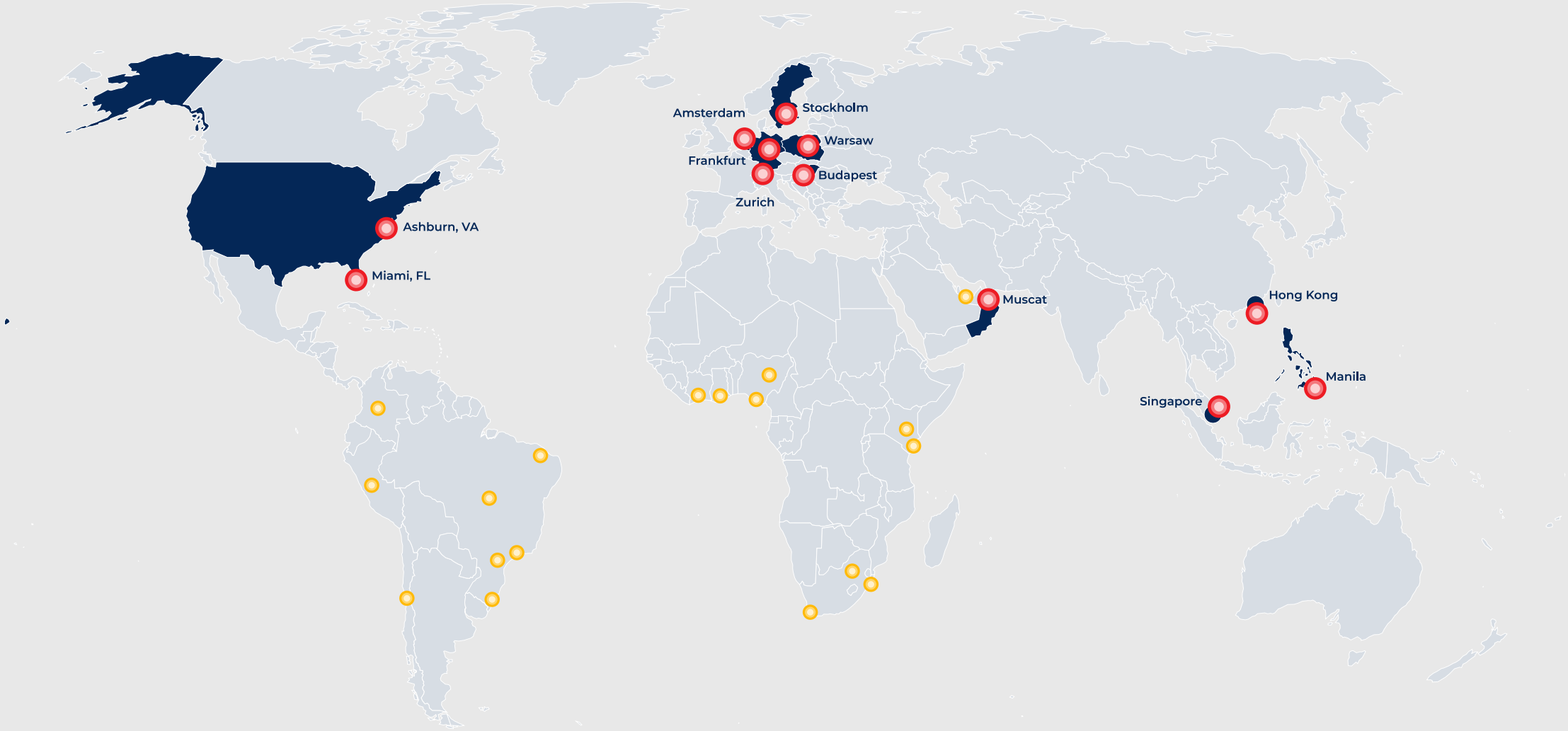
www.sns.ag

Use Cases

- Direct connectivity
- Data backhaul
- Content partners
- SIM testing
- IMSI partners
- Network Design
- Remote testing

Global core network

Global IP Services Provider



Our Network Challenges

Focus on high availability and resiliency

Fully encrypted MPLS core

- IPSec
- Transition to MACSec

High availability

- Full path and device redundancy
- Fast convergence (preferably sub second)
- Session consistency

High diversity

- Various solutions to connect to customers and partners
- Cloud connectivity

Used Vendors and Technologies

Juniper focused but open to other vendors

About 140 Juniper devices

- MX104/204 – mainly as MPLS P-routers
- SRX 34x/1500 – almost all in cluster/MNHA deployments, many integrated as PE-routers
- EX 4000 Series – smaller switch fabrics (EVPN-VXLAN, MC-LAG) or office switches
- QFX 5120 – MACSec, MC-LAG
- ACX – planned replacement for MX104s

Still over 110 Cisco devices

- Nexus 9K – datacenter switch fabric
- ASR 1K – BGP routers

And many others…

Micro PoP

Example of a small PoP deployment

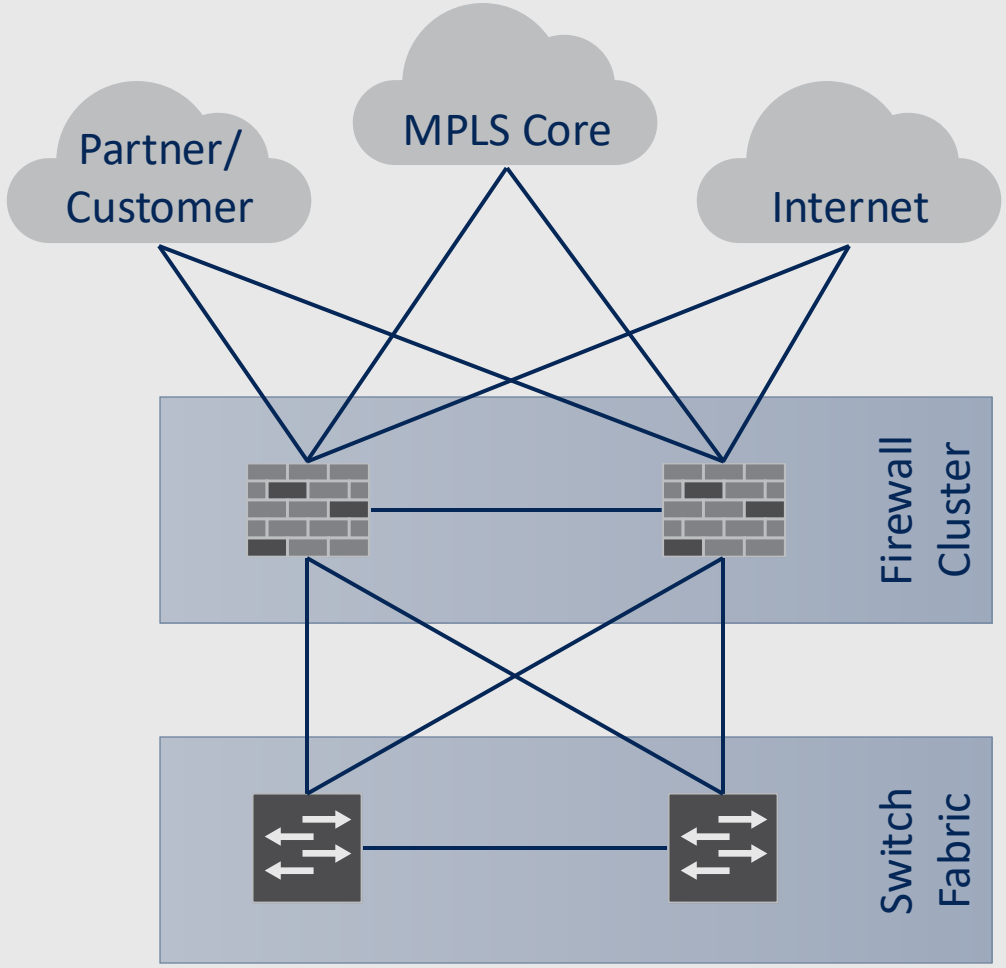
Minimal Setup

Juniper EX switches (EVPN-VXLAN)

SRX chassis cluster or MNHA

Connectivity to partner/customer, MPLS-core and internet according requirements

PDU’s and SIM enabled terminal servers (not shown)




SRX Chassis Cluster
Features and Limitations

Quick poll

Please raise your hand…

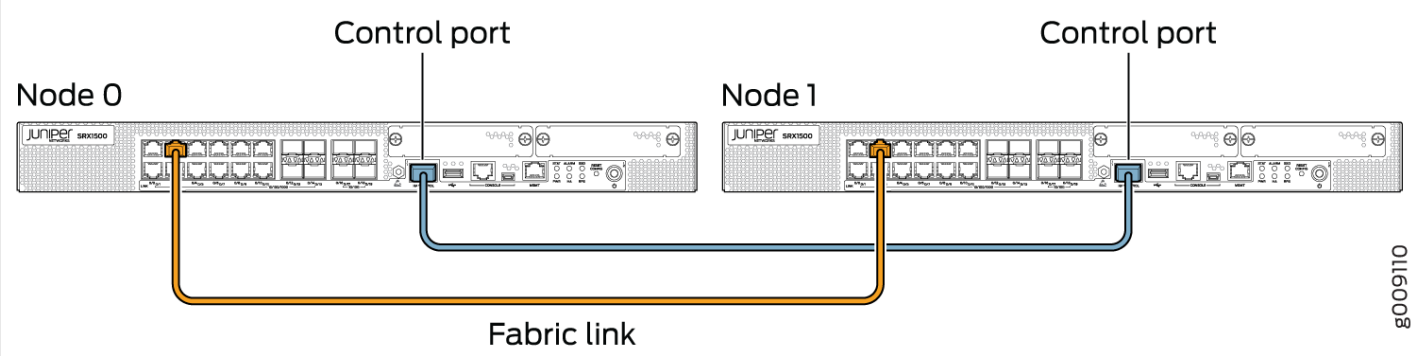
Have you:

- Ever used a Juniper SRX firewall?
- Deployed a SRX chassis cluster?
- Upgraded a chassis cluster?
- Ever encountered issues upgrading a chassis cluster?

Terminology

… used with chassis clusters.

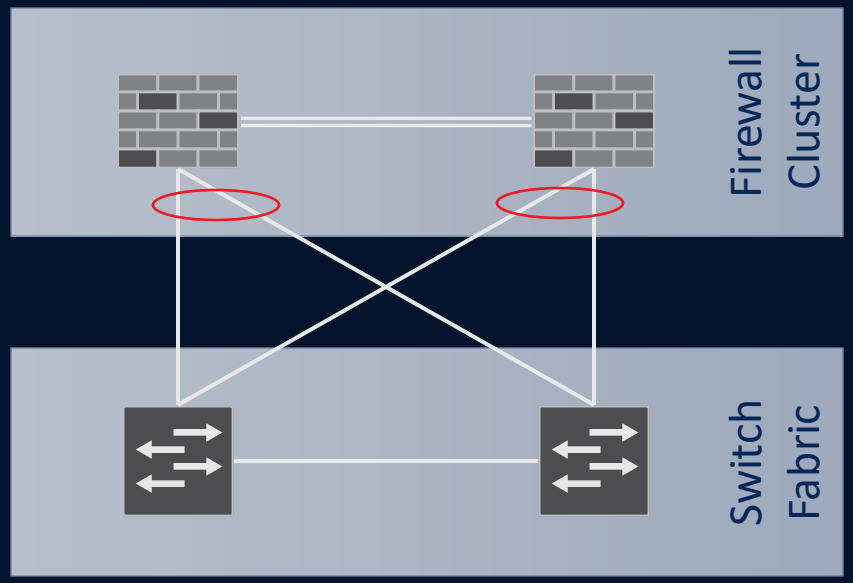
- Node0 and Node1
- Cluster control port
- Fabric links
- Redundancy Groups
- reth interfaces



https://www.juniper.net/documentation/us/en/software/junos/chassis-cluster-security-devices/topics/task/chassis-cluster-srx-series-hardware-connecting.html

Cluster pitfalls and drawbacks

Why we want something else…



- Single control plane (RG0) → high convergence times
- GRE tunnels required for MPLS
- QoS not properly working on GRE tunnels
- ISSU Issues
- active/active would double the links required

Multi-Node High Availability

A new approach to HA-Firewalls

Terminology

… used with MNHA.

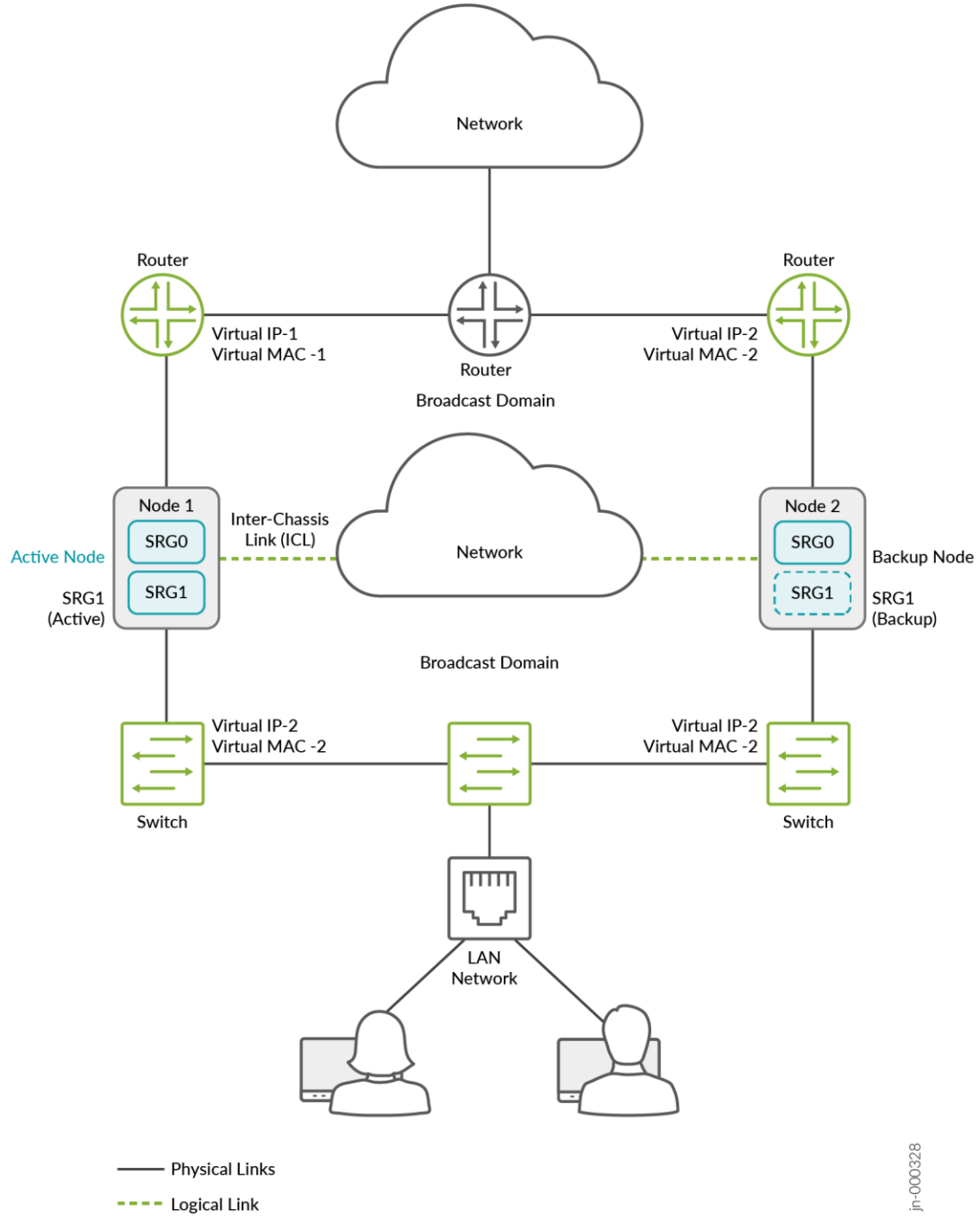
- Node1 and Node2
- Inter-Chassis Link (ICL): can be routed and encrypted
- Service Redundancy Groups (SRGs)
- Ae interfaces (active on both nodes)
- Signal Routes

MNHA Modes

Possible deployment modes:

- Layer 3 (routed)
- Default Gateway (switched)
- Hybrid →

https://www.juniper.net/documentation/us/en/software/junos/high-availability/topics/topic-map/mnha-introduction.html



Chassis Cluster – Config Highlights

```
set chassis cluster redundancy-group 0 node 0 priority 200
set chassis cluster redundancy-group 0 node 1 priority 100
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
set chassis cluster redundancy-group 1 interface-monitor xe-0/0/18 weight 128
set chassis cluster redundancy-group 1 interface-monitor xe-0/0/19 weight 128
set chassis cluster redundancy-group 1 interface-monitor xe-7/0/18 weight 128
set chassis cluster redundancy-group 1 interface-monitor xe-7/0/19 weight 128
…
set interfaces xe-0/0/18 gigether-options redundant-parent reth1
…
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options lacp active
set interfaces reth1 unit 0 family inet address x.y.z.47/31
```

MNHA – Service Redundancy Groups

[edit chassis high-availability]

services-redundancy-group 0 peer-id 2

services-redundancy-group 1 deployment-type hybrid
services-redundancy-group 1 peer-id 2
services-redundancy-group 1 virtual-ip 1 ip x.y.z.134/29
services-redundancy-group 1 virtual-ip 1 interface ae1.38
services-redundancy-group 1 virtual-ip 1 use-virtual-mac

<monitor objects>

services-redundancy-group 1 active-signal-route 10.255.1.1
services-redundancy-group 1 backup-signal-route 10.255.1.2
services-redundancy-group 1 preemption
services-redundancy-group 1 activeness-priority 200

MNHA – Monitor Objects

[edit chassis high-availability]

services-redundancy-group 1 monitor monitor-object TGT1 object-threshold 100
services-redundancy-group 1 monitor monitor-object TGT1 bfd-liveliness threshold 100
services-redundancy-group 1 monitor monitor-object TGT1 bfd-liveliness destination-ip x.y.z.38 src-ip x.y.z.39
services-redundancy-group 1 monitor monitor-object TGT1 bfd-liveliness destination-ip x.y.z.38 session-type singlehop
services-redundancy-group 1 monitor monitor-object TGT1 bfd-liveliness destination-ip x.y.z.38 interface gr-0/0/0.11
services-redundancy-group 1 monitor monitor-object TGT1 bfd-liveliness destination-ip x.y.z.38 weight 100

services-redundancy-group 1 monitor srg-threshold 100

→ https://www.juniper.net/documentation/us/en/software/junos/high-availability/topics/topic-map/mnha-monitoring-options.html

MNHA – Policy Options

```
[edit policy-options]
policy-statement TEST_VRF_EXPORT term SRG1_ACTIVE from prefix-list SRG1_LANS
policy-statement TEST_VRF_EXPORT term SRG1_ACTIVE from condition SRG1_ACTIVE_ROUTE_EXISTS
policy-statement TEST_VRF_EXPORT term SRG1_ACTIVE then metric 10
policy-statement TEST_VRF_EXPORT term SRG1_ACTIVE then accept

policy-statement TEST_VRF_EXPORT term SRG1_BACKUP from prefix-list SRG1_LANS
policy-statement TEST_VRF_EXPORT term SRG1_BACKUP from condition SRG1_BACKUP_ROUTE_EXISTS
policy-statement TEST_VRF_EXPORT term SRG1_BACKUP then metric 20
policy-statement TEST_VRF_EXPORT term SRG1_BACKUP then accept

condition SRG1_ACTIVE_ROUTE_EXISTS if-route-exists address-family inet 10.255.1.1/32
condition SRG1_ACTIVE_ROUTE_EXISTS if-route-exists address-family inet table inet.0

condition SRG1_BACKUP_ROUTE_EXISTS if-route-exists address-family inet 10.255.1.2/32
condition SRG1_BACKUP_ROUTE_EXISTS if-route-exists address-family inet table inet.0
```

Sessions: Chassis Cluster

```
{primary:node0}
user@zg0testfw2a> show security flow session
node0:
--------------------------------------------------------------------------

Session ID: 223338302505, Policy name: DEMO_POLICY_HTTPS/10, HA State: Active, Timeout: 1752, Session State:
Valid
  In: x.y.z.10/55052 --> x.y.z.14/443;tcp, Conn Tag: 0x0, If: reth1.39, Pkts: 7276, Bytes: 4263612,
  Out: x.y.z.14/443 --> x.y.z.10/51020;tcp, Conn Tag: 0x0, If: gr-0/0/0.21, Pkts: 6027, Bytes: 607496,

node1:
--------------------------------------------------------------------------

Session ID: 223338302505, Policy name: DEMO_POLICY_HTTPS/10, HA State: Backup, Timeout: 1752, Session State:
Valid
  In: x.y.z.10/55052 --> x.y.z.14/443;tcp, Conn Tag: 0x0, If: reth1.39, Pkts: 0, Bytes: 0,
  Out: x.y.z.14/443 --> x.y.z.10/55052;tcp, Conn Tag: 0x0, If: gr-0/0/0.21, Pkts: 0, Bytes: 0,
```

Sessions: MNHA

user@zg0testfw1a> show security flow session

Session ID: 64424942341, Policy name: DEMO_POLICY_HTTPS/4, HA State: Active, Timeout: 1788, Session State: Valid
  In: x.y.z.128/43536 --> x.y.z.20/443;tcp, Conn Tag: 0x0, If: ae1.38, Pkts: 456, Bytes: 25303, HA Wing State: Active,
  Out: x.y.z.20/443 --> x.y.z.128/43536;tcp, Conn Tag: 0x0, If: gr-0/0/0.11, Pkts: 412, Bytes: 7343, HA Wing State: Warm,

user@zg0testfw1b> show security flow session

Session ID: 64424942341, Policy name: DEMO_POLICY_HTTPS/4, HA State: Warm, Timeout: 1788, Session State: Valid
  In: x.y.z.128/43536 --> x.y.z.20/443;tcp, Conn Tag: 0x0, If: ae1.38, Pkts: 0, Bytes: 0, HA Wing State: Active,
  Out: x.y.z.20/443 --> x.y.z.128/43536;tcp, Conn Tag: 0x0, If: gr-0/0/0.11, Pkts: 0, Bytes: 0, HA Wing State: Warm,

Challenges we faced

... and other things we tripped over!

- IPSec tunnels only up on active node:
  → fixed in recent versions
- Interface naming:
  → needs to match on both nodes
- VIP dual stack not supported yet (!)
  → workaround possible with VRRP
- Ping sessions not synced:
  → expected (testing pitfall)

Conclusions

What MNHA solved for us…

- Faster graceful failover
  (S)RG0: ~2min → ~0s
  (S)RG1+: ~1s → ~0.2s
- Faster disaster failover:
  (S)RG0: ~2min → ~7s
  (S)RG1+: ~2min → ~7.5s
- Working QoS
- MTU reassembly possible again
- Upgrade consistency to be proven further…

Questions

... if there is time left!


sns