Data Center Interconnect (MDNOG 1)

Matvei Aleksandrov, SE
matvey.alexandrov@hpe.com

v02, October 2026

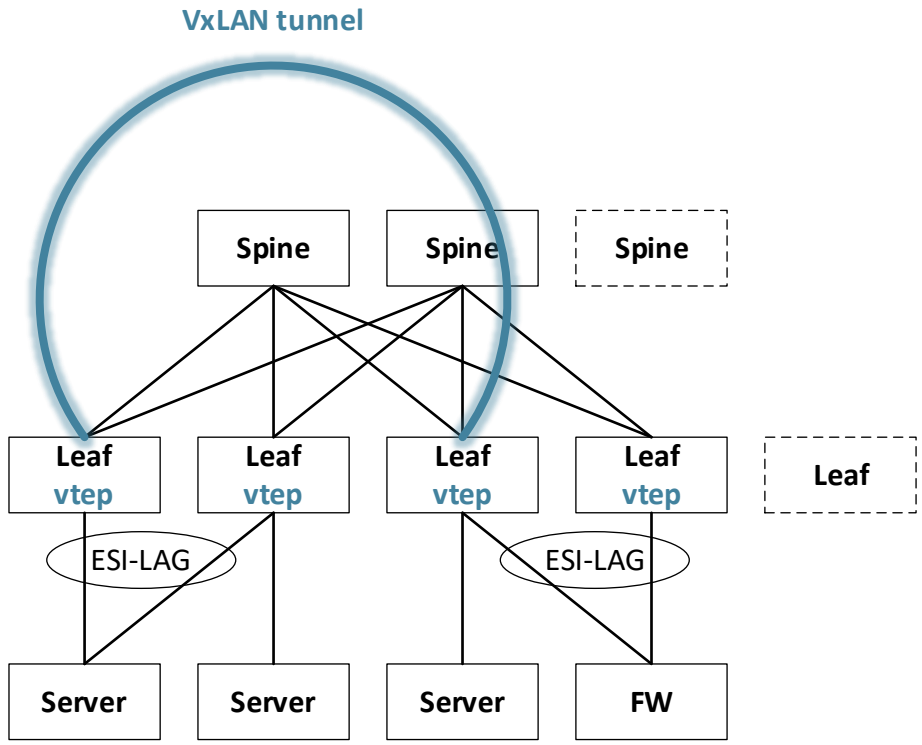
Agenda

01	Scope (and “Option A”)
02	Over The Top (OTT)
03	Integrated Interconnect
04	Additional options

Scope (and “Option A”)

What are we talking about?

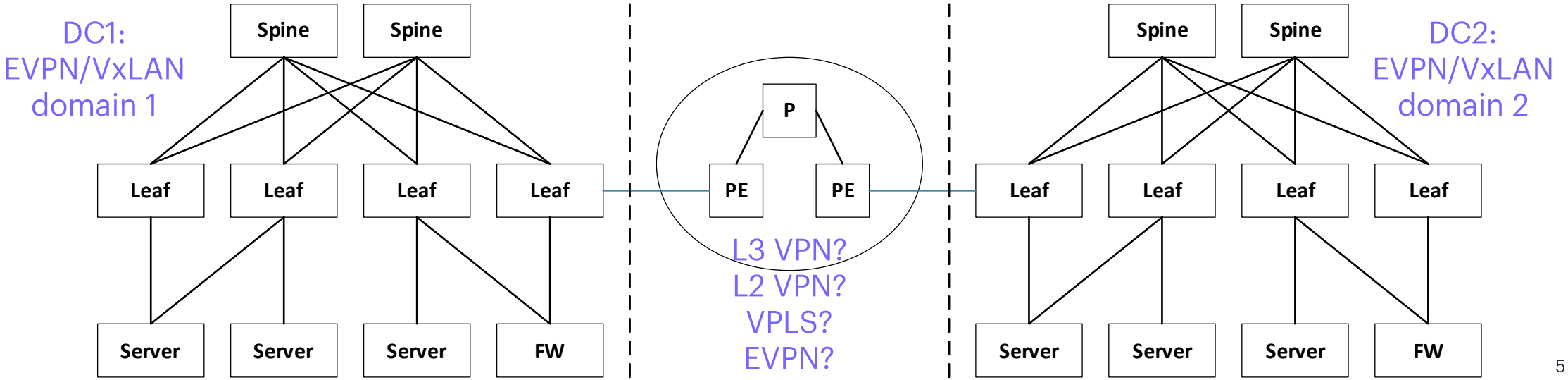
- Enterprise DC (aka “Private Cloud”)
  - Clos topology
  - IP fabric
  - Multi-tenancy requirement: **VxLAN** overlay
  - L2 stretch between racks: VxLAN overlay
  - BGP (IPv4 or IPv6) in Underlay
  - BGP (**EVPN**) in Overlay
  - ESI-LAG for multi-homing
  - Overlay routing on leaves (ERB) or spines (CRB). ERB is preferred.



How can we interconnect 2 (or more) DCs?

- Please note:
  - sometimes we have “L2 stretch between DCs” requirement
  - sometimes we don’t have that req. L3 routing is enough.

- “Option A”
  - I call it this way because I have ISP background
  - Interconnect links just L2 trunks w/ vlans
  - From leaf POV, PE is “just another server”
  - From PE POV, Leaf is “just another CE”
  - Easy understandable
  - High level of control
  - Many points of provision/coordination → slow



Do we have alternatives?

Yes, we are.
Alternative 1: **OTT**
Alternative 2: **Integrated Interconnect (IIC)**
Let’s discuss.

Over The Top (OTT)

Over The Top (OTT)

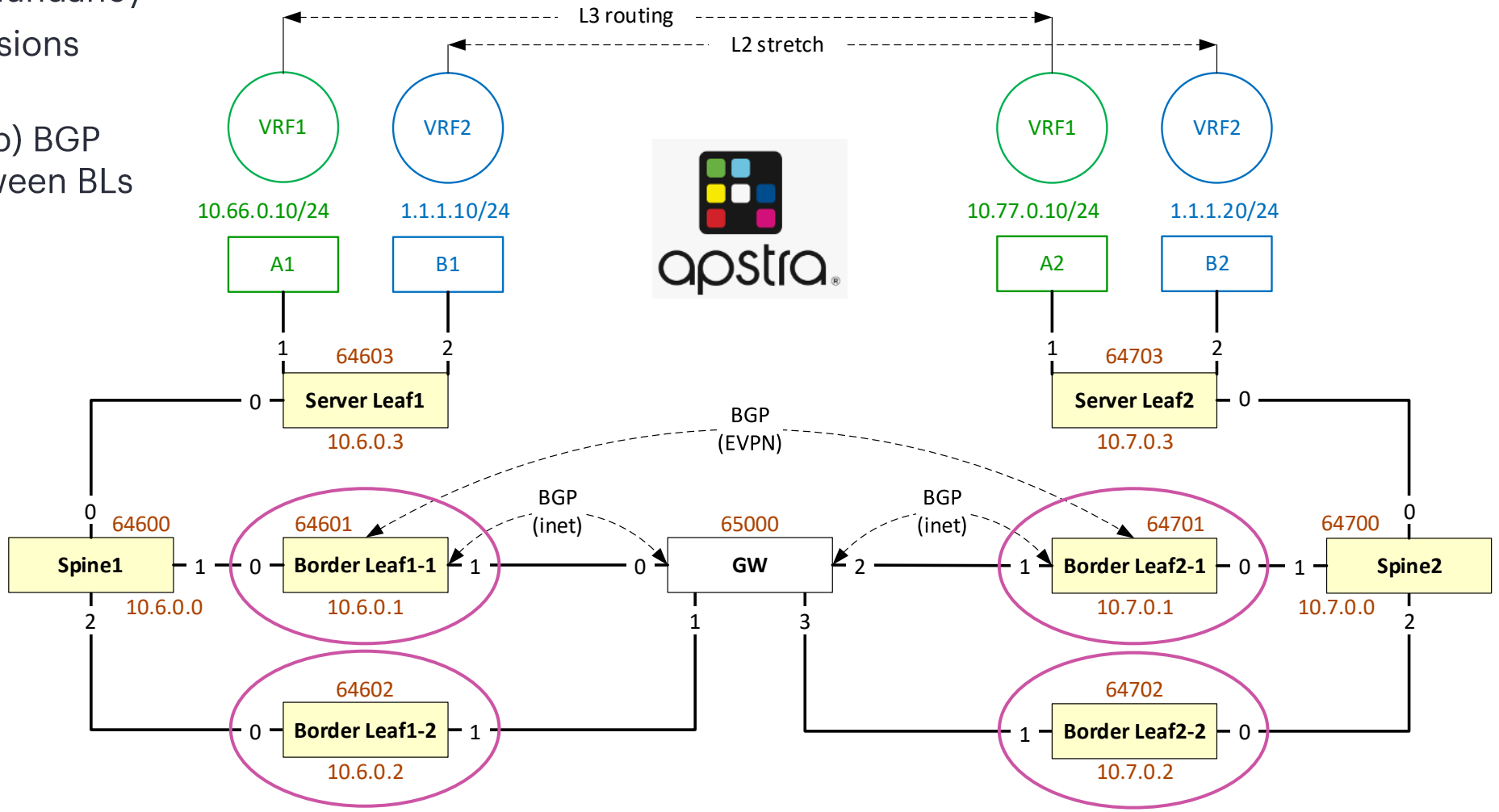
- The idea is very simple (and logical)
- VxLAN is UDP packets, right?
- For VxLAN to work, we just need IP connectivity between VTEPs (for ERB, this is leaf switches)
- Let's provide IP reachability between VTEPs in different DC …
- … And arrange BGP EVPN route exchange between VTEPs
- That should be enough, isn't it?



Demonstration (1) - Topology

- Welcome: “Border leaf” role
  - Usually two for redundancy
  - **Underlay** BGP sessions with External GW
  - **Overlay** (multi-hop) BGP EVPN session between BLs in different DCs

Does it remember Inter-AS Option-C?



Demonstration (2) – Underlay network

```
warrior@dc1-leaf1# run show route table inet.0 10.6.0.0/15 brief

10.6.0.0/32        *[BGP/170] 3d 04:21:11, localpref 100
                      AS path: 64600 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.6.0.1/32        *[BGP/170] 3d 04:21:11, localpref 100
                      AS path: 64600 64601 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.6.0.2/32        *[BGP/170] 3d 04:21:11, localpref 100
                      AS path: 64600 64602 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.6.0.3/32        *[Direct/0] 3d 04:23:27
                    >  via lo0.0
10.7.0.0/32        *[BGP/170] 2d 02:58:41, localpref 100
                      AS path: 64600 64601 65000 64701 64700 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.7.0.1/32        *[BGP/170] 2d 02:58:41, localpref 100
                      AS path: 64600 64601 65000 64701 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.7.0.2/32        *[BGP/170] 00:03:57, localpref 100
                      AS path: 64600 64601 65000 64701 64700 64702 I
                    >  to 10.16.0.4 via ge-0/0/0.0
10.7.0.3/32        *[BGP/170] 2d 02:58:41, localpref 100
                      AS path: 64600 64601 65000 64701 64700 64703 I
                    >  to 10.16.0.4 via ge-0/0/0.0
```

```
warrior@GW# run show bgp summary
...
inet.0
...
Peer                     AS  InPkt  OutPkt  OutQ  Flaps Last Up/Dwn State|#
10.99.1.0             64601  20496  20297   0     0     2d 3:15:35 Establ
  inet.0: 4/4/4/0
10.99.3.0             64701  20364  20168   0     0     2d 2:56:08 Establ
  inet.0: 4/4/4/0
```

- 10.6.0.x – DC1 loopbacks
- 10.7.0.x – DC2 loopbacks
- Now dc1-leaf1 (10.6.0.3) can ping dc2-leaf1 (10.7.0.3)



Demonstration (3) – Overlay network

```
warrior@dc1-border1# run show route advertising-protocol bgp 10.7.0.1 table bgp.evpn.0

bgp.evpn.0: 19 destinations, 19 routes (19 active, 0 holddown, 0 hidden)
Restart Complete
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:10.6.0.3:65534::6000::52:54:00:28:93:91/304 MAC/IP
*                         10.6.0.3                             64600 64603 I
  2:10.6.0.3:65534::9000::4e:82:33:23:a3:2a/304 MAC/IP
*                         10.6.0.3                             64600 64603 I
  2:10.6.0.3:65534::9000::52:54:00:cc:52:4f/304 MAC/IP
*                         10.6.0.3                             64600 64603 I
  2:10.6.0.3:65534::6000::52:54:00:28:93:91::10.66.0.10/304 MAC/IP
*                         10.6.0.3                             64600 64603 I
  2:10.6.0.3:65534::9000::52:54:00:cc:52:4f::1.1.1.10/304 MAC/IP
*                         10.6.0.3                             64600 64603 I
  3:10.6.0.3:65534::6000::10.6.0.3/248 IM
*                         10.6.0.3                             64600 64603 I
  3:10.6.0.3:65534::9000::10.6.0.3/248 IM
*                         10.6.0.3                             64600 64603 I
  5:10.6.0.3:2::0::10.66.0.0::24/248
*                         10.6.0.3                             64600 64603 I
  5:10.6.0.3:2::0::10.26.0.0::32/248
*                         10.6.0.3                             64600 64603 I
  5:10.6.0.3:4::0::10.26.0.1::32/248
*                         10.6.0.3                             64600 64603 I
```

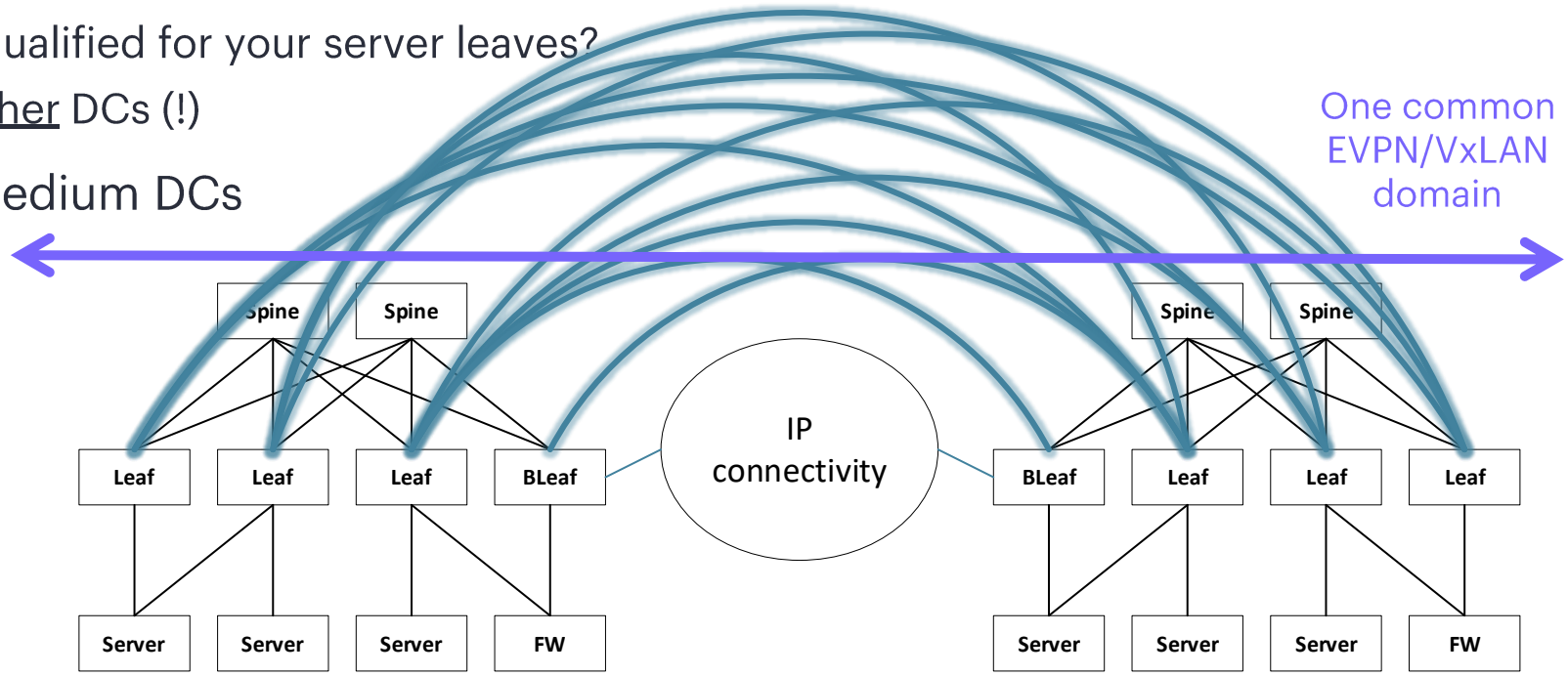
That could be also on spines, despite of ERB architecture (!)

```
warrior@dc1-border1# show protocols bgp group evpn-gw
type external;
multihop {
    ttl 30;
    no-nexthop-change;
}
multipath { multiple-as; }
neighbor 10.7.0.1 {
    description facing_evpn-gw1-evpn-gateway;
    local-address 10.6.0.1;
    import ( EVPN_GW_IN );
    family evpn { signaling; }
    export ( EVPN_GW_OUT );
    peer-as 64701;
}
vpn-apply-export;
```

- For L3 routing, we should adjust VRF import policies
- For L2 stretch, the easiest way is to use the same VxLAN VNI

Easy? Yes, but …

- The OTT solution is easy to understand, maintain, and troubleshoot
- But this is just extended (shared) EVPN control plane and that's all
- We need to coordinate RTs, RDs, VNIs between DC
- Actually, we have full mesh of VxLAN tunnels between server leaves – scaling issue (!)
  - What is max.number of VTEPs qualified for your server leaves?
  - If one DC grows – it affect all other DCs (!)
- OTT is suitable only for small/medium DCs



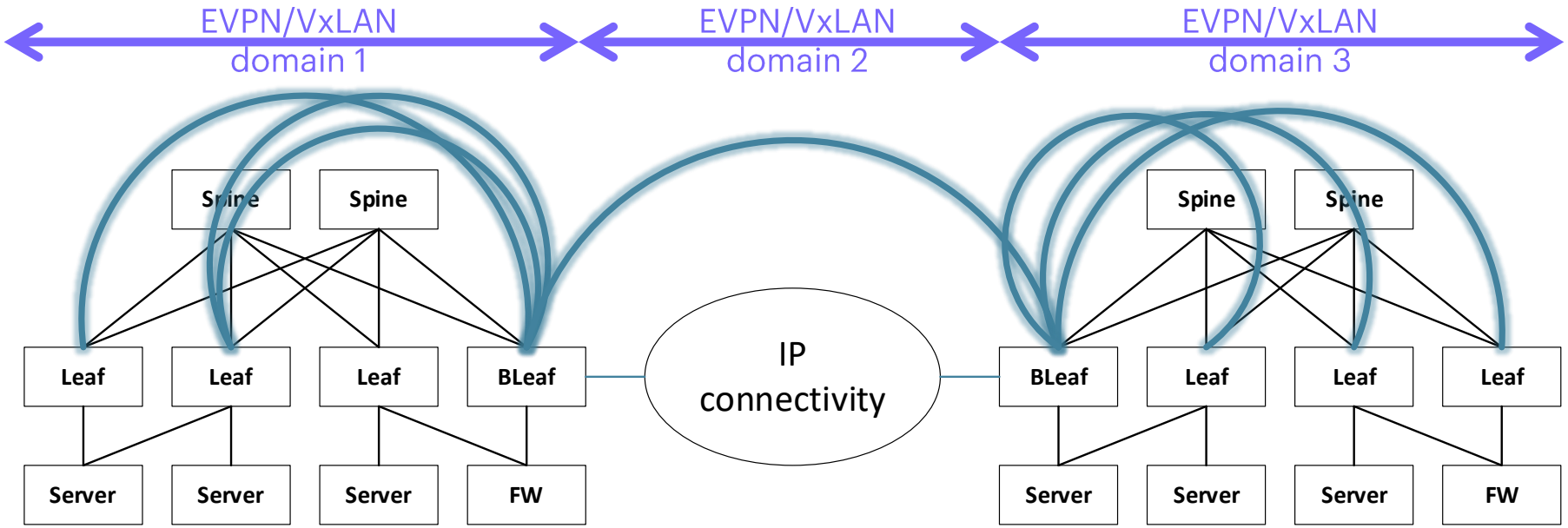
Integrated Interconnect

Integrated Inter-Connect (IIC) or VxLAN Stitching

- RFC 9014
- Border Leaves re-originates EVPN prefixes between DCs
  - Changing RT, RD, ESI
- Full isoltation of EVPN/VxLAN domains – no “full mesh” anymore
- Better control – we explicitly define which VNI and VRF/L3 subnets should be inter-connected

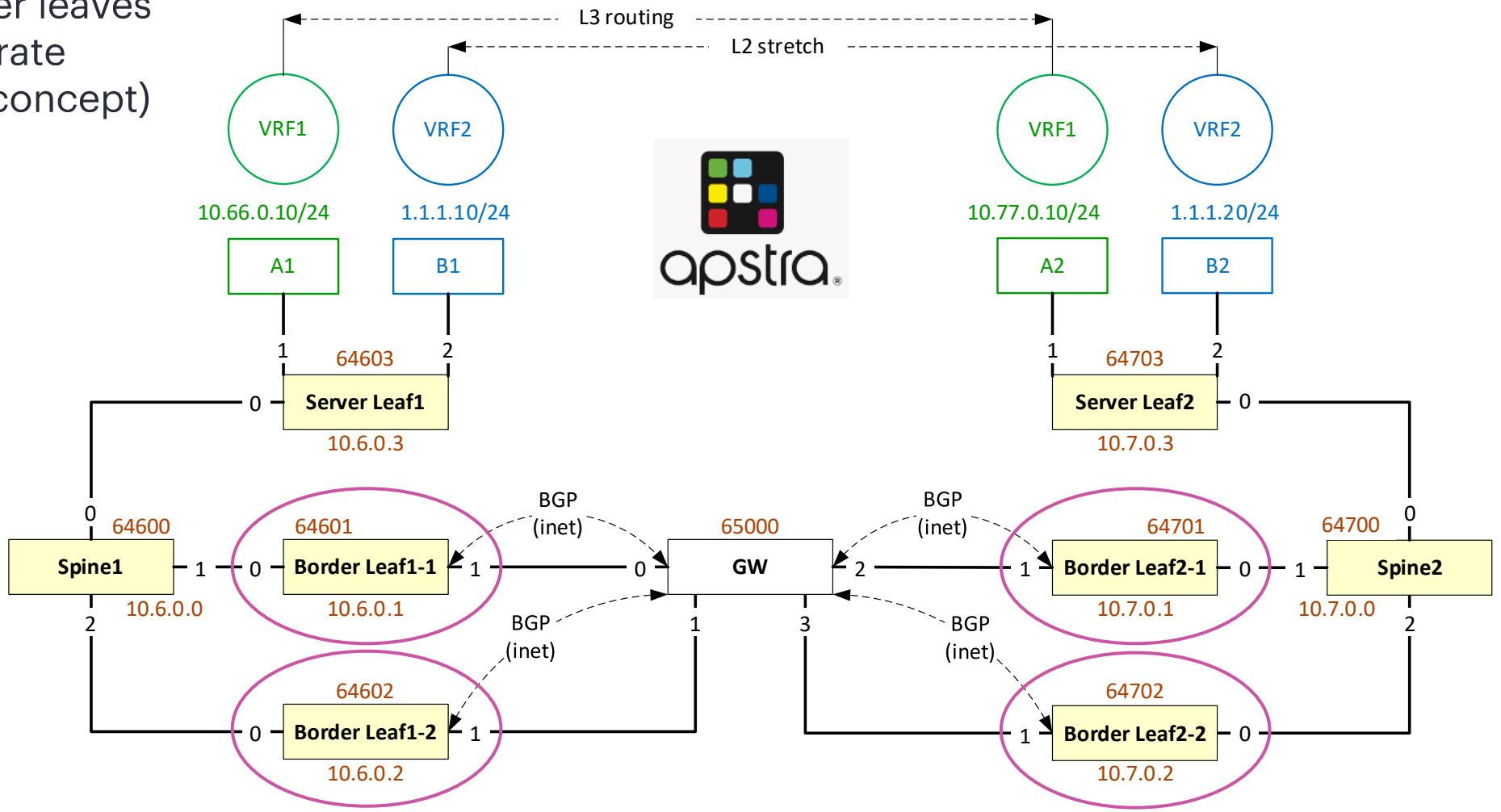
RFC 9014
Interconnect Solution for Ethernet VPN (EVPN) Overlay Networks

Does it remember Inter-AS Option-B?



Demonstration (1) - Topology

- The same topology
- Now we use 2 border leaves in each DC (to illustrate “interconnect ESI” concept)



Demonstration (2) – Underlay network

- Exactly the same Underlay connectivity – BGP “family inet” sessions with external GW
- Border leaves need to ping each other – that’s all; we don’t need all the loopbacks to exchange

```
warrior@dc1-border1# run ping 10.7.0.1 source 10.6.0.1          // 10.7.0.1 is Border-Leaf-2-1 loopback
PING 10.7.0.1 (10.7.0.1): 56 data bytes
64 bytes from 10.7.0.1: icmp_seq=0 ttl=63 time=2.402 ms
^C
--- 10.7.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.402/2.402/2.402/0.000 ms

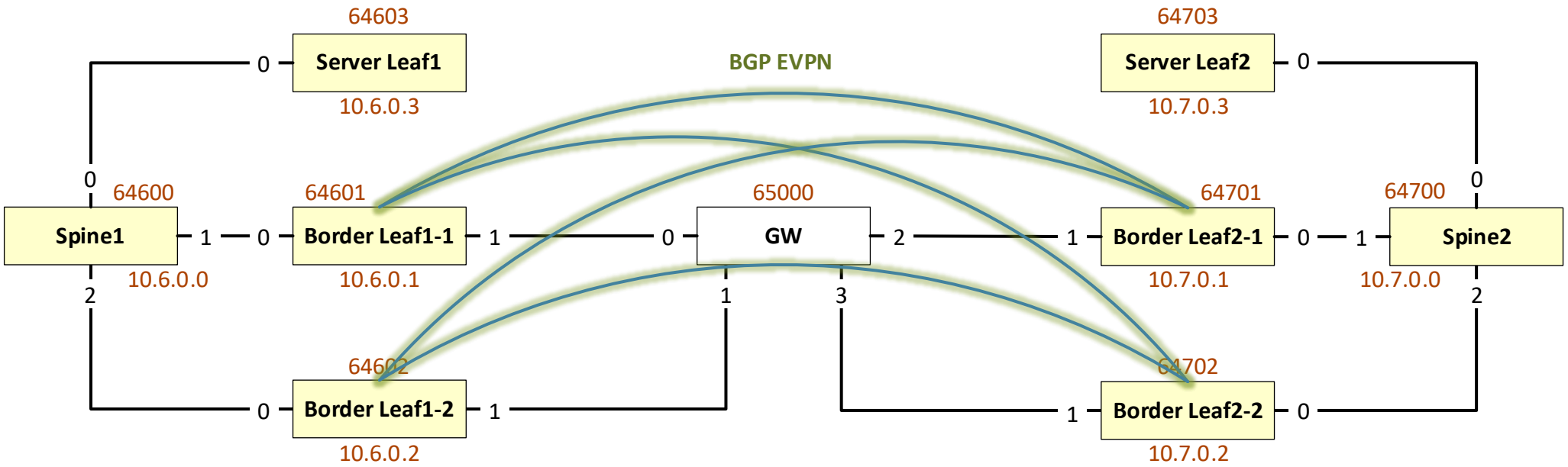
warrior@dc1-border1# run ping 10.7.0.2 source 10.6.0.1          // 10.7.0.2 is Border-Leaf-2-2 loopback
PING 10.7.0.2 (10.7.0.2): 56 data bytes
64 bytes from 10.7.0.2: icmp_seq=0 ttl=63 time=2.810 ms
^C
--- 10.7.0.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.810/2.810/2.810/0.000 ms
```

Border Leaf in DC1



Demonstration (3) – Overlay network

- Border Leaves establish (multi-hop) BGP EVPN sessions with each other – very similar to OTT case



Let’s consider Type 5 stitching (1)

- DC1, VRF1, 10.66.0.0/24 (RT target:16000:1)
- DC2, VRF1, 10.77.0.0/24 (RT target:17000:1)
- 61:61 is a new “Interconnected RT” (the same on all BLs)

```
warrior@dc1-leaf1# show routing-instances VRF1
instance-type vrf;
routing-options {…}
protocols {
    evpn {
        irb-symmetric-routing {
            vni 16000;
        }
        ip-prefix-routes {
            advertise direct-nexthop;
            encapsulation vxlan;
            vni 16000;
            export BGP-AOS-Policy-VRF1;
        }
    }
}
interface irb.3;
interface lo0.2;
route-distinguisher 10.6.0.3:2;
vrf-target target:16000:1;
```

Server Leaf in DC1

```
warrior@dc1-border1# show routing-instances VRF1
instance-type vrf;
routing-options {…}
protocols {
    evpn {
        irb-symmetric-routing {
            vni 16000;
        }
        interconnect {
            vrf-target target:61:61;
            route-distinguisher 10.6.0.1:65531;
        }
        ip-prefix-routes {
            advertise direct-nexthop;
            encapsulation vxlan;
            vni 16000;
            export BGP-AOS-Policy-VRF1;
        }
    }
}
interface lo0.2;
route-distinguisher 10.6.0.1:2;
vrf-target target:16000:1;
```

RT change (!)
RD change (!)

Border Leaf in DC1

Let’s consider Type 5 stitching (2)

- BGP announcement from BLeaf1-1 to BLeaf2-1 (again: change of RT, RD, Next-Hop, and Router-MAC (!))
- BLeaf2-1 to Server-Leaf-2: the same thing. Re-origination (!)

```
warrior@dc1-border1# run show route advertising-protocol bgp 10.7.0.1 table bgp.evpn.0 match-prefix *10.66.* detail

* 5:10.6.0.1:65531::0::10.66.0.0::24/248 (1 entry, 1 announced)
 BGP group evpn-gw type External
     Route Distinguisher: 10.6.0.1:65531
     Route Label: 16000
     Overlay gateway address: 0.0.0.0
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [64601] 64600 64603 I
     Communities: target:61:61 encapsulation:vxlan(0x8) router-mac:2c:6b:f5:35:e7:f0

* 5:10.6.0.1:65531::0::10.66.0.10::32/248 (1 entry, 1 announced)
 BGP group evpn-gw type External
     Route Distinguisher: 10.6.0.1:65531
     Route Label: 16000
     Overlay gateway address: 0.0.0.0
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [64601] I
     Communities: target:61:61 encapsulation:vxlan(0x8) router-mac:2c:6b:f5:35:e7:f0
```

```
warrior@dc1-border1# run show interfaces irb | match address
  Current address: 2c:6b:f5:35:e7:f0, Hardware address: 2c:6b:f5:35:e7:f0
```

Let’s consider Type 5 stitching (3)

- Server-Leaf-2 sees 10.66.0.0/24 Green network from DC1 as “behind local border leaves”

```
warrior@dc2-leaf1# run show route 10.66.0.0/24 exact table VRF1.inet.0

10.66.0.0/24       @[EVPN/170] 19:01:38
                    >  to 10.17.0.4 via ge-0/0/0.0
                    [EVPN/170] 19:01:38
                    >  to 10.17.0.4 via ge-0/0/0.0
                   #[Multipath/255] 19:01:38, metric2 0
                    >  to 10.17.0.4 via ge-0/0/0.0
                    >  to 10.17.0.4 via ge-0/0/0.0

warrior@dc2-leaf1# run show route 10.66.0.0/24 exact table VRF1.inet.0 detail | match "Protocol next hop"
                Protocol next hop: 10.7.0.1
                Protocol next hop: 10.7.0.2
                Protocol next hop: 10.7.0.1
                Protocol next hop: 10.7.0.2

warrior@dc2-leaf1# run show evpn ip-prefix-database l3-context VRF1 prefix 10.66.0.0/24

EVPN->IPv4 Imported Prefixes
Prefix                                  Etag
10.66.0.0/24                            0
  Route distinguisher    VNI/Label/SID          Router MAC         Nexthop/Overlay GW/ESI   Route-Status  Reject-Reason
  10.7.0.1:2             17000                  2c:6b:f5:b9:a7:f0  10.7.0.1                  Accepted      n/a
  10.7.0.2:2             17000                  2c:6b:f5:58:78:f0  10.7.0.2                  Accepted      n/a
```



Let’s consider Type 2 stitching (1)

- DC1, VRF2, VNI 9000 (RT target:9000:1)
- DC2, VRF2, VNI 9000 (RT target:9000:1)
- 51:51 is “I-RT”, 00:0a:ff::01:00:00:01 is “I-ESI”

```
warrior@dc1-leaf1# show routing-instances evpn-1
instance-type mac-vrf;
protocols {
    evpn {
        encapsulation vxlan;
        default-gateway do-not-advertise;
        extended-vni-list all;
        vni-options {
            vni 9000 {
                vrf-target target:9000:1;
            }
        }
    }
}
vtep-source-interface lo0.0;
service-type vlan-aware;
interface ge-0/0/2.0;
route-distinguisher 10.6.0.3:65534;
vrf-target target:100:100;
vlans { vn5 { vlan-id 5; vxlan { vni 9000; }}}
```

Server Leaf in DC1

```
warrior@dc1-border1# show routing-instances evpn-1
instance-type mac-vrf;
protocols {
    evpn {
        encapsulation vxlan;
        default-gateway do-not-advertise;
        extended-vni-list all;
        interconnect {
            vrf-target target:51:51;
            route-distinguisher 10.6.0.1:65533;
            esi {
                00:0a:ff:00:00:00:01:00:00:01;
                all-active;
            }
            interconnected-vni-list 9000;
        }
        vni-options {
            vni 9000 {
                vrf-target target:9000:1;
            }
        }
    }
}
vtep-source-interface lo0.0;
service-type vlan-aware;
route-distinguisher 10.6.0.1:65534;
vrf-target target:100:100;
vlans { vn3 { vlan-id 3; vxlan { vni 9000; }}}
```

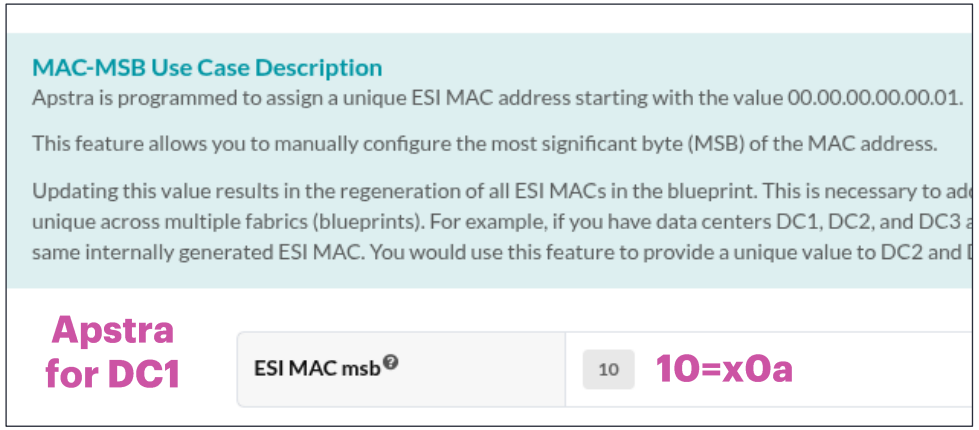
RT change (!)
RD change (!)
ESI change (!)

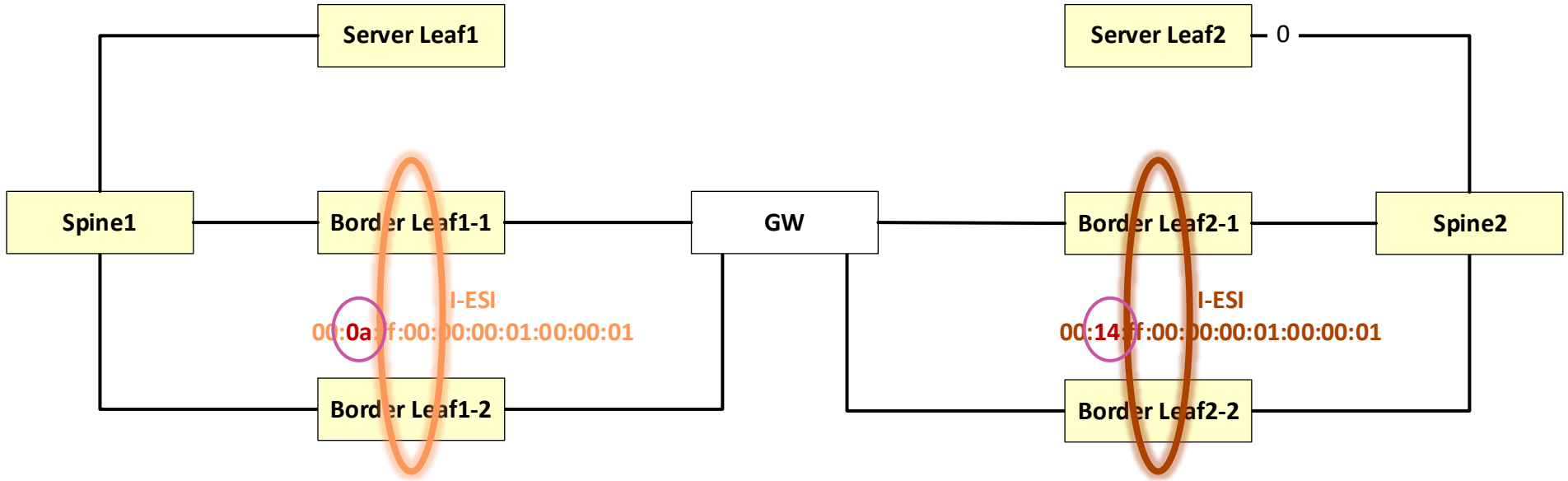
Explicitly defined VNIs

Border Leaf in DC1

Virtual ESI (Interconnect ESI) concept

- I-ESI doesn't connect with any physical LAG
- Used to maintains EVPN machinery (aliasing, redundancy) between BLs
- Of course should be unique per DC





Let’s consider Type 2 stitching (2)

- BGP announcement from BLeaf1-1 to BLeaf2-1 (again: change of RT, RD, Next-Hop, and ESI (!))
- BLeaf2-1 to Server-Leaf-2: the same thing. Re-origination (!)

```
warrior@dc1-border1# run show route advertising-protocol bgp 10.7.0.1 table bgp.evpn.0 match-prefix *9000* detail

* 2:10.6.0.1:65533::9000::52:54:00:cc:52:4f::1.1.1.10/304 MAC/IP (1 entry, 1 announced)
 BGP group evpn-gw type External
     Route Distinguisher: 10.6.0.1:65533
     Route Label: 9000
     ESI: 00:0a:ff:00:00:00:01:00:00:01
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [64601] I
     Communities: target:51:51 encapsulation:vxlan(0x8)

* 3:10.6.0.1:65533::9000::10.6.0.1/248 IM (1 entry, 1 announced)
 BGP group evpn-gw type External
     Route Distinguisher: 10.6.0.1:65533
     Route Label: 9000
     PMSI: Flags 0x0: Label 9000: Type INGRESS-REPLICATION 10.6.0.1
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [64601] I
     Communities: target:51:51 encapsulation:vxlan(0x8)
```



Let’s consider Type 5 stitching (3)

- Server-Leaf-2 sees remote MAC in VNI 9000 as “behind local border leaves”

```
warrior@dc2-leaf1# run show ethernet-switching table

Ethernet switching table : 2 entries, 2 learned
Routing instance : evpn-1
   Vlan                 MAC                 MAC       GBP    Logical                 SVLBNH/      Active
   name                 address             flags     tag    interface               VENH Index   source
   vn3                  52:54:00:50:8f:22   D                ge-0/0/1.0
   vn5                  52:54:00:cc:52:4f   DR               esi.652                              00:14:ff:00:00:00:01:00:00:01

warrior@dc2-leaf1# run show mac-vrf forwarding mac-table 52:54:00:cc:52:4f

Ethernet switching table : 4 entries, 4 learned
Routing instance : evpn-1
   Vlan                 MAC                 MAC       GBP    Logical                 SVLBNH/      Active
   name                 address             flags     tag    interface               VENH Index   source
   vn5                  52:54:00:cc:52:4f   DR               esi.652                              00:14:ff:00:00:00:01:00:00:01

warrior@dc2-leaf1# run show mac-vrf forwarding vxlan-tunnel-end-point esi
ESI                           RTT                      VLNBH INH     SVLBNH    ESI-IFL    LOC-IFL    #RVTEPs
00:14:ff:00:00:00:01:00:00:01 evpn-1                   652   1048580 0         esi.652              2      Aliasing
    RVTEP-IP             RVTEP-IFL      VENH     MASK-ID   FLAGS        MAC-COUNT
    10.7.0.2             vtep.32770     650      1         0x12         1
    10.7.0.1             vtep.32769     648      0         0x12         1
```

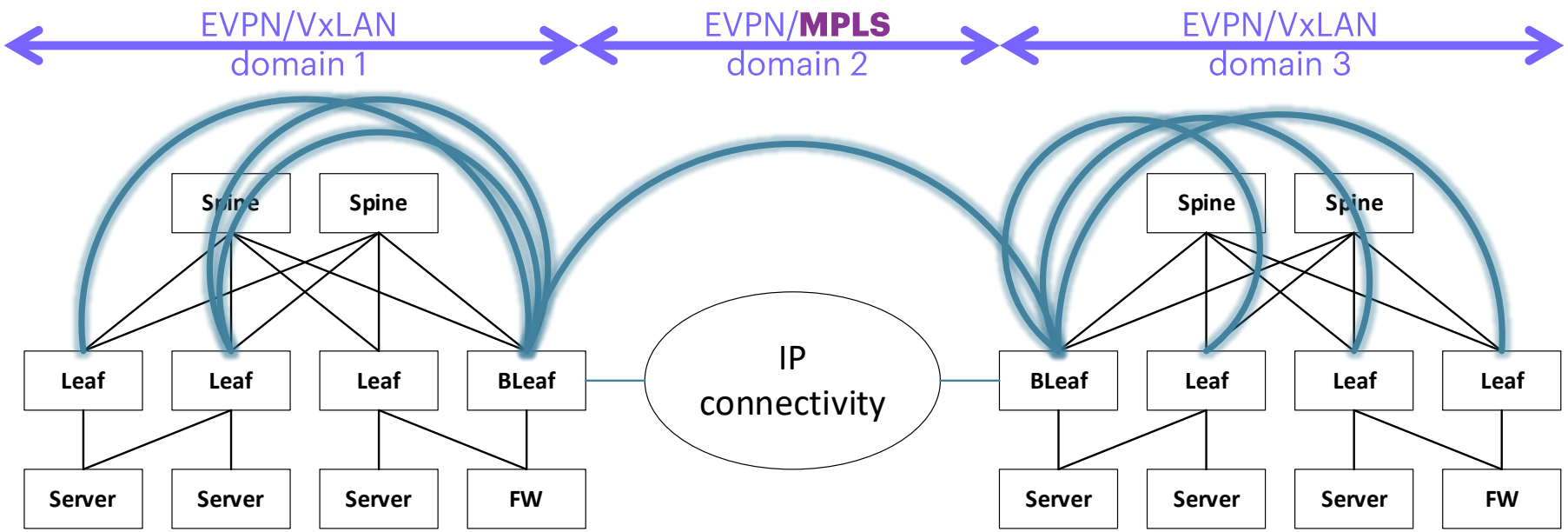


Additional options

Additional options (1)

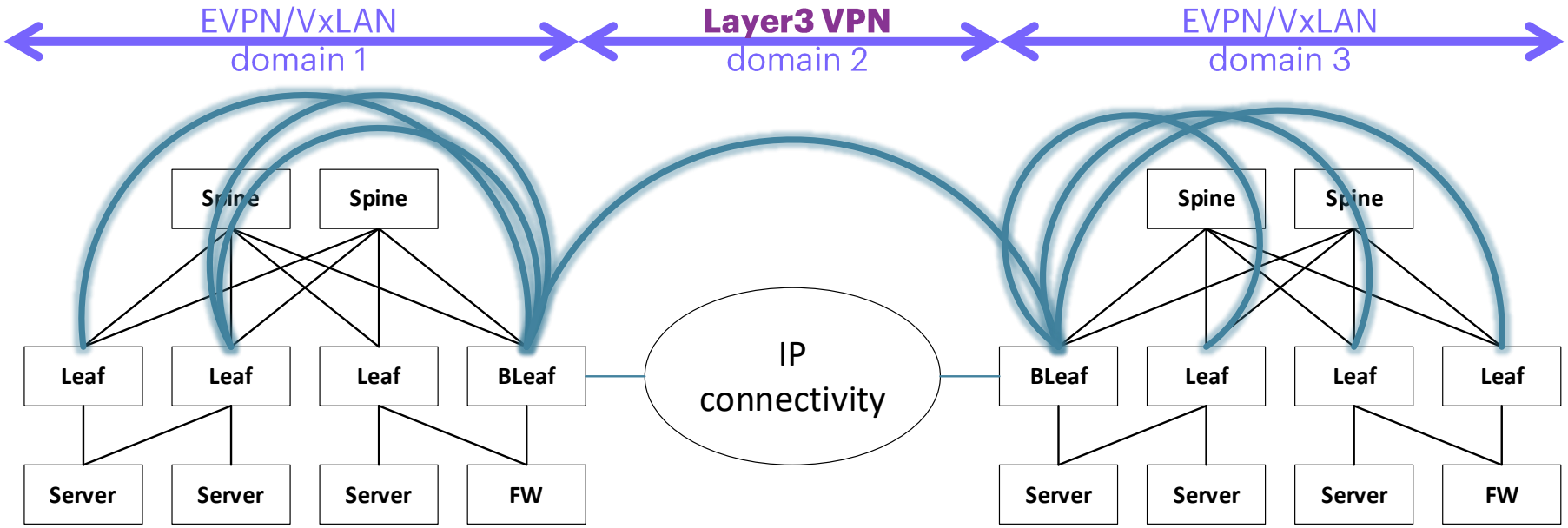
- EVPN/MPLS between DCs
- Border Leaves (also) change incapsulation

```
warrior@dc1-border1# set routing-instances evpn-1 protocols evpn interconnect encapsulation ?
Possible completions:
  mpls                 MPLS encapsulation
  vxlan                VXLAN encapsulation
```



Additional options (2)

- Layer3 VPN between DCs
- Of course, in this case we can do only L3/Type-5 interconnect, not L2/Type-2
- Demonstration TBD



Thank You

Matvei Aleksandrov, SE
matvey.alexandrov@hpe.com

My Telegram channel
https://t.me/jp2mp

Youtube playlist
https://www.youtube.com/playlist?list=PLWzebvZln3TD2Ymij92cM0h-F2jb47NS-

© 2026 Hewlett Packard Enterprise Development LP