Mix & Match BGP

Another way to think about architecture

03-JUN-2026

How many of us struggle with BGP?

To understand it, pretty much all of us:

- Did
- Are, or
- Will have to

(struggle, that is)

So why not try to come up with a simpler way to think about it? (And then discuss amongst ourselves)

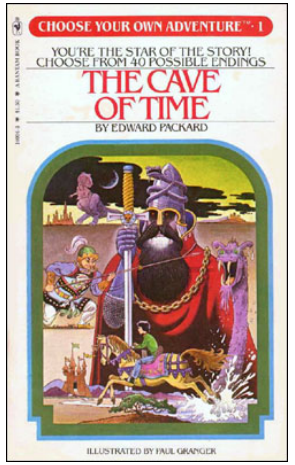
And why do we struggle with BGP?

Usually, it's confusion over why we need all three:

- **EBGP —** to exchange routes with other companies' networks while maintaining our autonomy
- **IBGP —** to keep track of all these routes, figure out which one is best, and let all the other routers know which route is best
- And an **IGP —** so that all our routers know how to reach each other.

How many of us had these as kids?

- “Choose Your Own Adventure” books or Mix & Match books
- Fashion Plates
- Mighty Men & Monster Maker

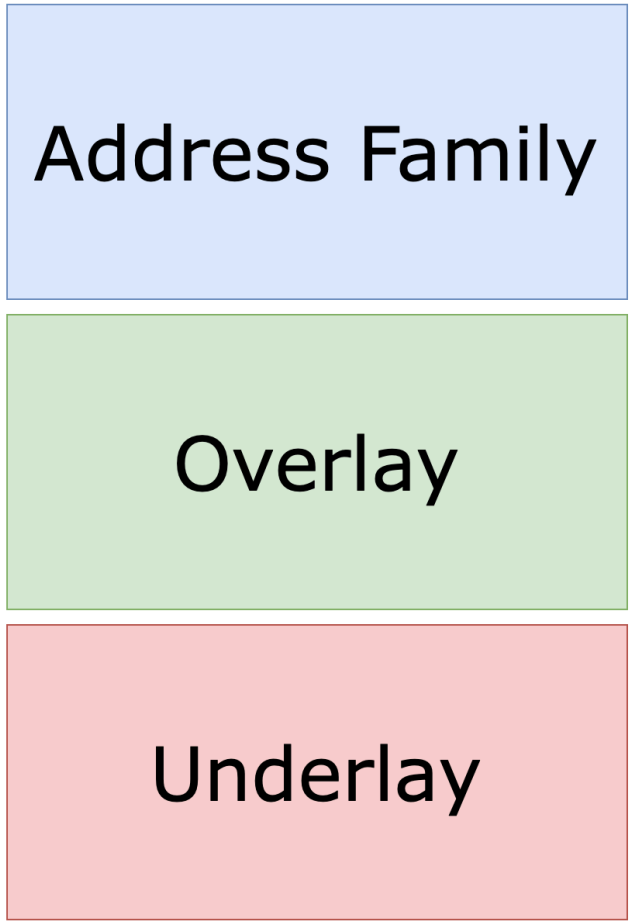


Bantam Books





BGP Plates (Network Maker?)



- The L3 traffic you're trying to send*
- Some kind of overlay
- Some kind of underlay

} Borrowed from VXLAN

That's it! (Most of the time)

* Can be IP only, or multiprotocol BGP — "address families" or AFI/SAFIs. Usually to route to/from VLANs, or PIM or other payload

What do we mean by overlay?



© Nevin Thompson (BOK tower and depot, cropped)

- Let’s say you’re a mail clerk in a big office building.
- Letters and packages are important to your business, so you have several “mail rooms”: USPS, FedEx, and interoffice mail on 3 different floors.
- Any of them could probably deliver something outside the building for you, but you need those backups.

And these mail rooms are noisy



Nevin Thompson (BOK tower and depot, cropped)

- They are all yelling at you (or maybe emailing you regularly) to tell you what destinations they can reach, and what floor they are each on. (next hop)
- But you have to live with it, because you need to know you can get packages and letters out of the building if there's a Post Office holiday and the interoffice mail team goes on strike.

OK, so what's an underlay then?



© Esther Bubley (Pneumatic tubes in DC, 1943)

It's what you need if you lack basic building navigation skills:

- How do I get this envelope from floor 12 to floor 4?
  > Get in the elevator, push the button with a 4 on it, get out when the doors open, hand it to someone. (Loopbacks)

←Or, if you're lucky, it's the writing on the pneumatic tubes. (Labels)

Outfit 1 - Old-School ISP or CDN

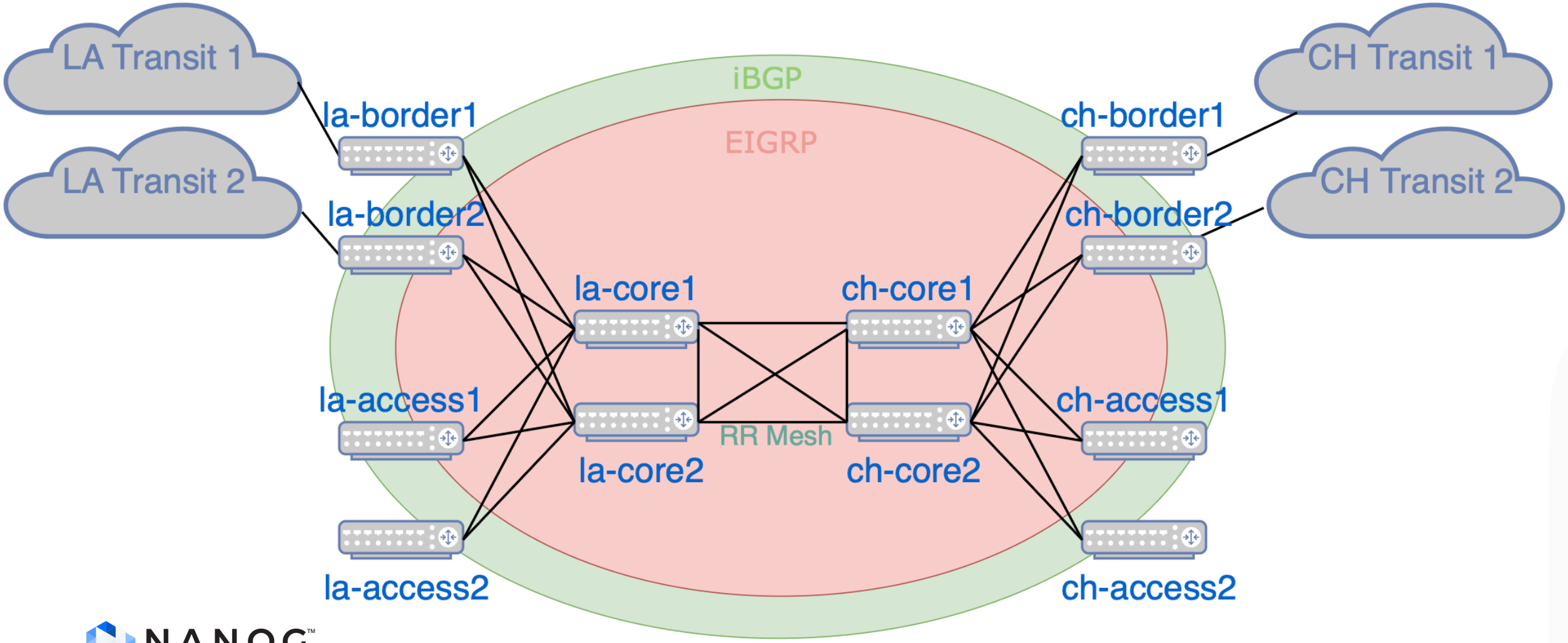


- AF - IPv4 only, with IPv6 added later (maybe)
- Overlay - iBGP (+RRs and communities)
- Underlay - EIGRP (in 2001 at least) or OSPF

Should anyone still use this? Maybe, if:

- You have a really simple use case or old hw
- New RFCs would be overkill
- You have embedded OSPF skillset

Example - Old-School ISP or CDN



Example - Old-School ISP or CDN

```
router eigrp 100
 redistribute connected
 redistribute static
 passive-interface Serial0/0/1
 network 192.168.1.0
 no default-information out
 no auto-summary
 eigrp log-neighbor-changes
!
```

- “address-family” — not even in the configs back then.
- AFI/SAFI would’ve been 1/1 for ipv4 unicast

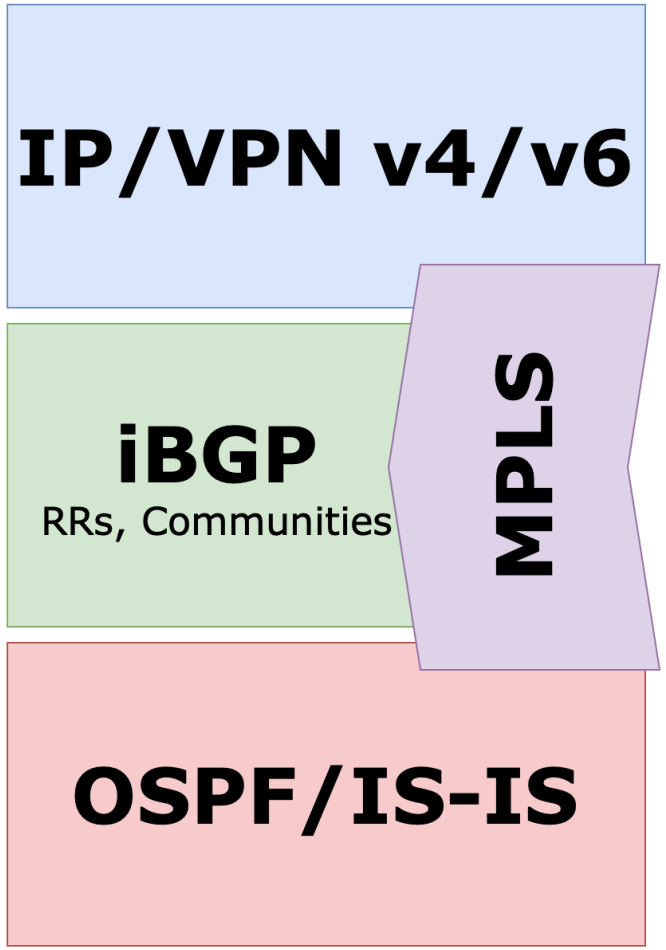
Example - Old-School ISP or CDN

```
router bgp 100
 no synchronization
 bgp cluster-id 2
 bgp log-neighbor-changes
 bgp deterministic-med
 network 172.19.144.0 mask 255.255.240.0 route-map LA-CIDR
 aggregate-address 172.19.144.0 255.255.240.0 summary-only
 neighbor CORE-IBGP peer-group
 neighbor CORE-IBGP remote-as 100
 neighbor CORE-IBGP update-source Loopback0
 neighbor CORE-IBGP send-community
 neighbor CORE-IBGP route-map REMOTE-CORE-IN in
```

Example - Old-School ISP or CDN

```
 neighbor CLIENT-IBGP peer-group
 neighbor CLIENT-IBGP route-reflector-client
 neighbor CLIENT-IBGP remote-as 100
 neighbor CLIENT-IBGP update-source Loopback0
 neighbor CLIENT-IBGP send-community
 neighbor CLIENT-IBGP route-map LA-ACCESS-IN in
 neighbor 10.19.44.40 peer-group CORE-IBGP
 neighbor 10.19.44.40 description ch-core1 ! and other cores
 neighbor 10.19.44.97 peer-group CLIENT-IBGP
 neighbor 10.19.44.97 description la-access1 ! and others
 no auto-summary
!
```

Outfit 2 - Evolved Service Provider

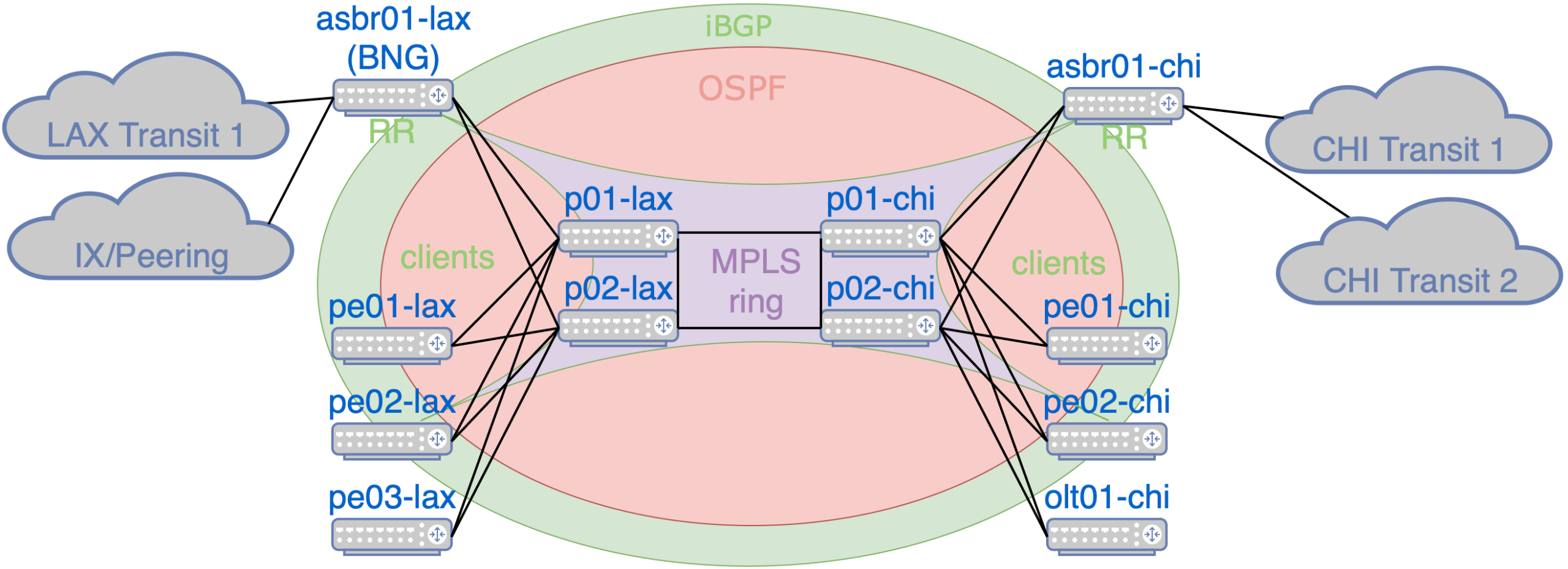


- AF - IPv4 and IPv6 + maybe VPNv4 and v6*
- Overlay - iBGP (still RRs and communities)
- Underlay - OSPF or IS-IS

* With an “interlay” like MPLS (pneumatic tubes in the building)

Some fiber networks may still use this, perhaps with BNG/OLTs. Same reasons still apply; but the choices come with tradeoffs

Example 2 - Evolved Service Provider



NANOG™

Example 2 - Evolved Service Provider

```
lo0 {
    unit 0 {
        family inet {
            address 192.168.16.1/32;
        }
        family mpls;
    }
}
```

- Here, mpls is enabled on lo0 with a “family” command but it’s not AFI/SAFI
- Next slide: AFI/SAFI “family” on BGP neighbors for
  - ➡ l2vpn signaling (25/65), but
  - ➡ inet-vpn not shown since these are P clients (1/128)

Example 2 - Evolved Service Provider

```
set protocols bgp group P-CLIENTS type internal
set protocols bgp group P-CLIENTS local-address 192.168.16.1
set protocols bgp group P-CLIENTS passive
set protocols bgp group P-CLIENTS family l2vpn signaling
set protocols bgp group P-CLIENTS cluster 192.168.16.1
set protocols bgp group P-CLIENTS bfd-liveness-detection minimum-interval 300
set protocols bgp group P-CLIENTS bfd-liveness-detection multiplier 4
set protocols bgp group P-CLIENTS neighbor 192.168.16.3 description p01-lax
set protocols bgp group P-CLIENTS neighbor 192.168.16.4 description p02-lax
set protocols bgp group P-CLIENTS neighbor 192.168.16.5 description p01-chi
set protocols bgp group P-CLIENTS neighbor 192.168.16.6 description p02-chi
set protocols ospf export directly-connected
set protocols ospf reference-bandwidth 100g
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols mpls interface all
```

Example 2 - Evolved Service Provider

```
asbr01-lax> show route table mpls.0

mpls.0: 38 destinations, 38 routes (38 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 34w2d 02:11:08, metric 1
                       to table inet.0
13                 *[MPLS/0] 34w2d 02:11:08, metric 1
                       Receive
16                 *[VPN/0] 34w2d 02:11:07
                    >  via lsi.0 (test-vrf1), Pop
4523               *[VPN/170] 00:38:12, metric2 110, from 192.168.16.6
                    >  to 192.168.20.2 via xe-0/1/2.0, Swap 262145, Push 1647(top)
```

Example 2 - Evolved Service Provider

```
asbr01-lax> show route table inet.3

inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

192.168.16.1/32    *[LDP/9] 6d 02:34:33, metric 10
                    >  to 192.168.20.1 via xe-0/0/0.0
192.168.16.2/32    *[LDP/9] 03:58:18, metric 250
                    >  to 192.168.20.38 via ae1.0, Push 300096
192.168.16.4/32    *[LDP/9] 03:58:18, metric 150
                    >  to 192.168.20.38 via ae1.0, Push 300112
192.168.16.5/32    *[LDP/9] 03:58:18, metric 50
                    >  to 192.168.20.38 via ae1.0
192.168.16.6/32    *[LDP/9] 03:58:18, metric 100
                    >  to 192.168.20.38 via ae1.0, Push 300080
```

An OSPF underlay, for all time?



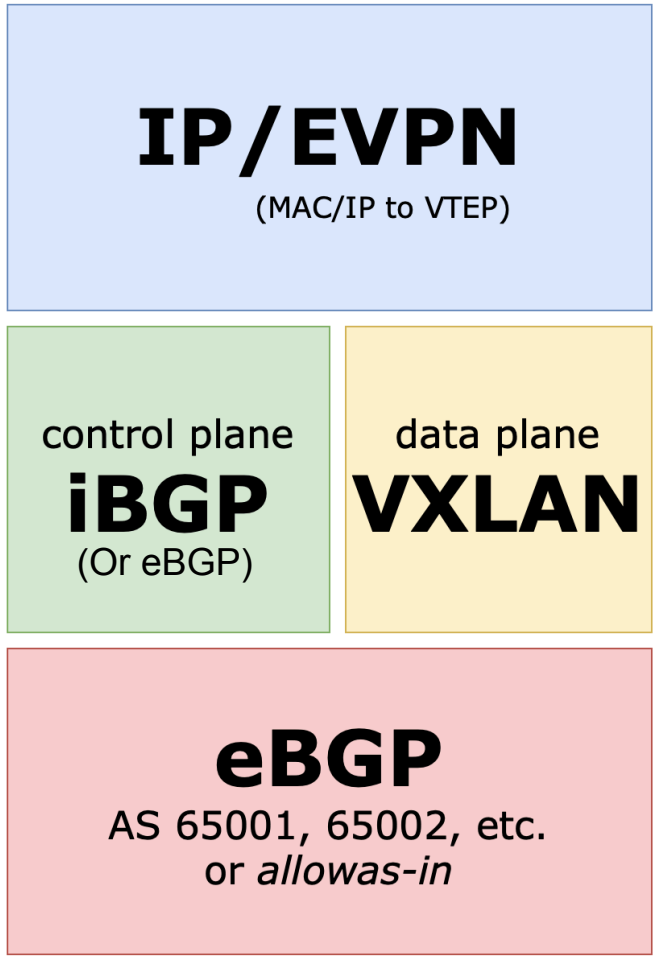


Parks & Recreation

Leslie, could one say that a Clos design could scale into the hundreds, with nothing more than OSPF in the underlay, and become the tapestry of the greatest network ever sculpted?

One ***could*** say that, but should one?

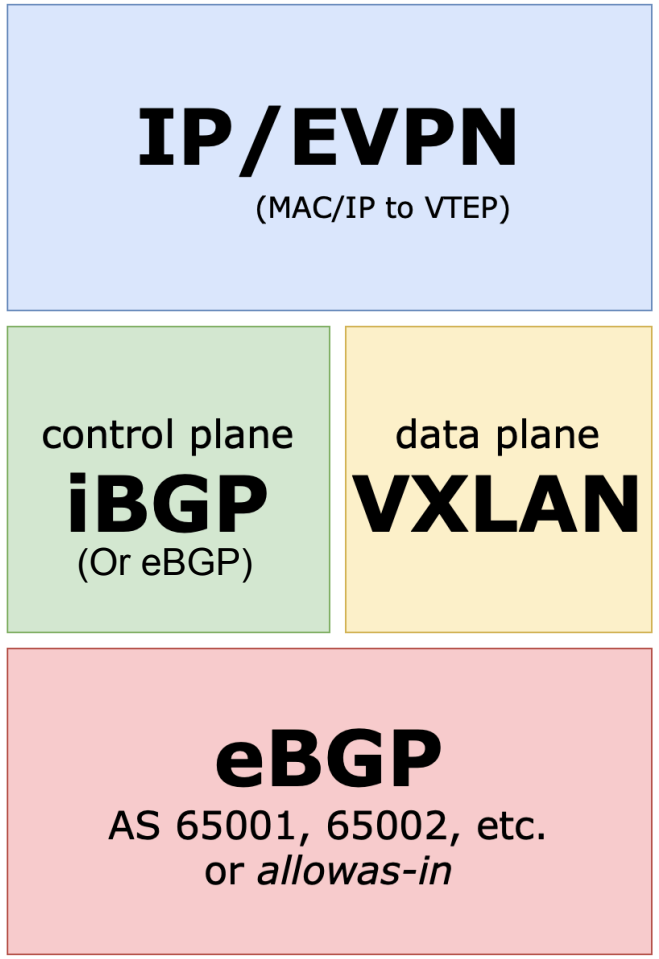
Outfit 3 - Future-proofed Fabric



- AFI/SAFI - Most of them, including EVPN (25/70)
- Overlay - iBGP/VXLAN, or even eBGP/VXLAN
- Underlay - eBGP unnumbered, perhaps

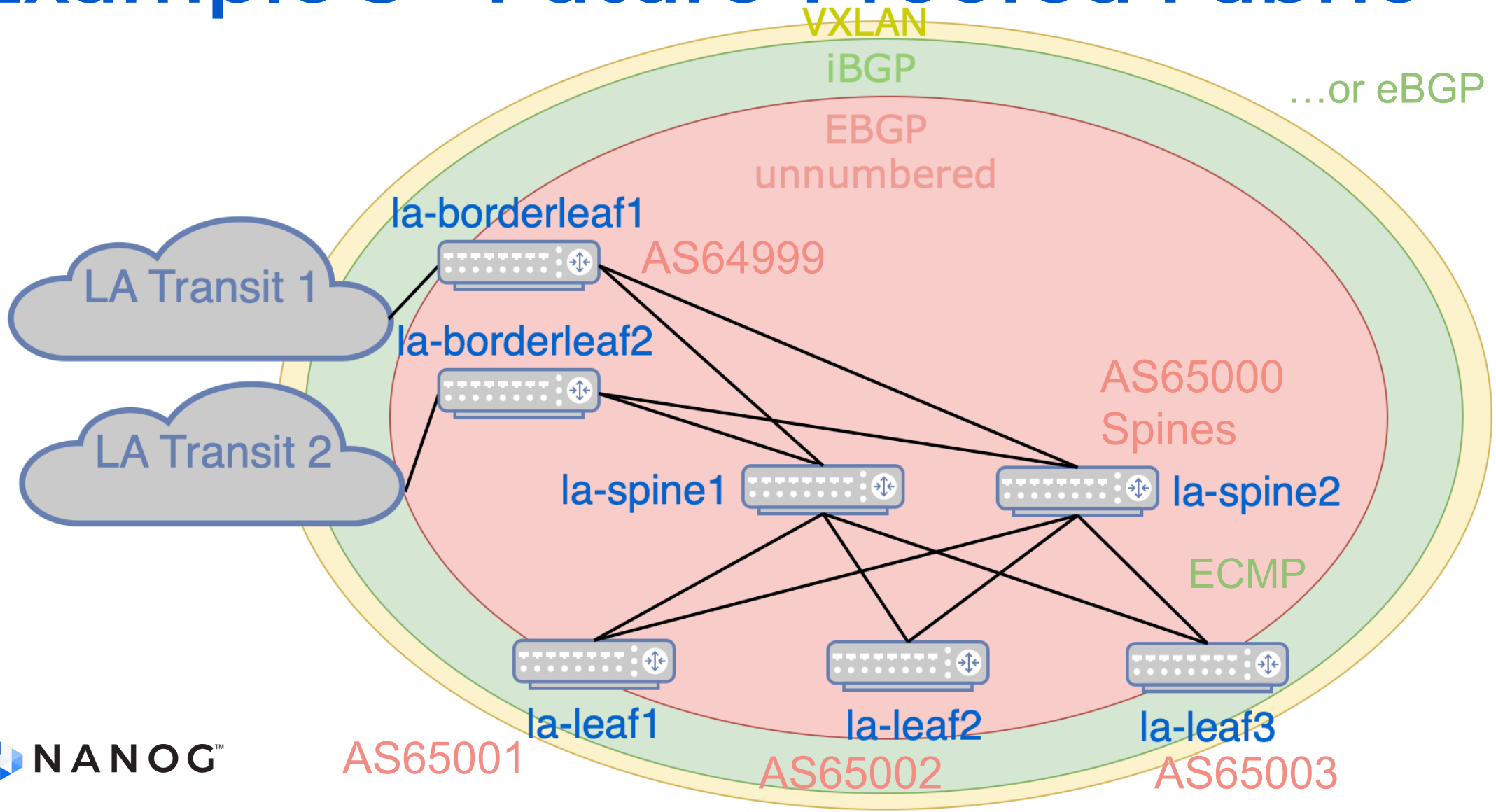
Now that we have VXLAN, we can uninvent the “interlay” again—and have nearly limitless VLAN connectivity (pneumatic tubes everywhere!)

Outfit 3 - Future-proofed Fabric



- VPLS was active-standby and required MAC flooding, but EVPN routes announce MAC+IP and VTEP next-hop
- DF limits BUM traffic; ECMP enables all-active multihoming (EVPN-MH) and load-balancing
- UDP/VXLAN header is 50-54 bytes, so watch out for MTU/Jumbo frames
- Most consider automation a basic necessity

Example 3 - Future-Proofed Fabric



Example 3 - Future-Proofed Fabric

```
#all are prefixed by:
set / network-instance default protocols bgp
```

```
autonomous-system 65000
router-id 10.0.0.1

group underlay-unnumbered admin-state enable
group underlay-unnumbered export-policy [ EXPORT-SYSTEM0 ]
group underlay-unnumbered import-policy [ ACCEPT-ALL ]
group underlay-unnumbered afi-safi ipv4-unicast admin-state enable

dynamic-neighbors interface ethernet-1/1.0 peer-group underlay-unnumbered
dynamic-neighbors interface ethernet-1/1.0 allowed-peer-as [ 65001..65003 64999 ]
```

Example 3 - Future-Proofed Fabric

```
#all are prefixed by:
set / network-instance default protocols bgp
group evpn-overlay admin-state enable
group evpn-overlay peer-as 65000
group evpn-overlay afi-safi evpn admin-state enable
group evpn-overlay afi-safi evpn route-reflector cluster-id 10.0.0.1
group evpn-overlay afi-safi ipv4-unicast admin-state disable

neighbor 1.1.1.1 peer-group evpn-overlay
neighbor 2.2.2.2 peer-group evpn-overlay
neighbor 3.3.3.3 peer-group evpn-overlay
neighbor 9.9.9.9 peer-group evpn-overlay
```

* on ibgp leaves, local-as 65000 is set

Example 3 - Future-Proofed Fabric

```
set / tunnel-interface vxlan1 vxlan-interface 100 type bridged
set / tunnel-interface vxlan1 vxlan-interface 100 ingress vni 10100

set / network-instance Tenant-VLAN100 type mac-vrf
set / network-instance Tenant-VLAN100 interface ethernet-1/10.0
set / network-instance Tenant-VLAN100 vxlan-interface vxlan1.100
```

Example 3 - Future-Proofed Fabric

```
# show tunnel-interface vxlan1
================================================
Tunnel Interface   : vxlan1
Operational State  : up
------------------------------------------------
vxlan-interface    : 100
Type               : bridged
VNI Ingress        : 10100
Network Instance   : Tenant-VLAN100
------------------------------------------------
Dynamic VTEPs:
  Destination VTEP : 2.2.2.2
  Egress VNI       : 10100
```

So what do we think—does this work?

Please ask questions or answer them at the mic—let's keep an open dialog (both here and after we go home). But first…

Don’t Forget

- You’re responsible for what happens, so
- Ask questions (of people, not just LLMs), and
- Check your work / model it before deploying

“In theory, theory and practice are the same. In practice, they are not.”

—Probably not Einstein, Yogi Berra, or Richard Feynman (was it Benjamin Brewster?)

Nothing to it but to test it



- Physical lab, or virtual
- ContainerLab runs on:
  - Windows (WSL)
  - MacOS (even Apple Silicon/M chips)
  - Linux / Docker containers

Other Resources

- MTU Deep Dive - Dan Massameno for Packet Pushers
- IPSpace Blog EVPN Series - Ivan Pepelnjak, et al
- Three Dimensions of BGP Address Family Nerd Knobs - ibid

  “The list goes on and on and on and on… and every time you open another RFC you realize how much you don’t know.” — from the last link above

Thank you

03-JUN-2026

NANOG™

Additional Slides

03-JUN-2026

NANOG™

VPLS - label block and mac-table

```
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.16.3:10119:2:262144/96 *[BGP/170] 00:14:22, localpref 100, from 192.168.16.3
                      Nexthop: 192.168.16.3 via ge-0/0/0.0
                      Communities: target:65001:10119

> show vpls mac-table instance TEST-EPL
Routing instance: TEST-EPL
VLAN BD             MAC address       Type       Age       Interfaces
default             00:50:56:a1:b2:c3 learn      0         ge-0/0/1.0
default             00:50:56:f4:d5:e6 remote     -         ge-0/0/0.0 (Label 262144)
```

EVPN - Mac to VTEP

```
evpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:192.168.16.3:10119::0::00:50:56:f4:d5:e6/304 *[BGP/170] 00:05:10, localpref 100
                      Nexthop: 192.168.16.3 via ge-0/0/0.0
                      Communities: target:65001:10119
```

- RT 65001 and RD 10119 stay the same