Nexus

- Inkoppling 25 gbit/s mellan Cisco Nexus och Palo Alto, FEC
- vPC
- Border Gateway Protocol (BGP)
- MACSEC
- SNMP
- ACL
- EVPN
- Uppgradera NXOS
- PBR, Policy Based Routing
- Breakout
- Buffrar, utgående trafik
- TCAM, TCAM-Carving
- Kontrollera faktisk forwarding med packet capture

Inkoppling 25 gbit/s mellan Cisco Nexus och Palo Alto, FEC

FEC, forward error correction, är en teknik som används för felrättning över fysiska länkar. Det finns olika varianter av FEC. Cisco använder sig per default "auto" för FEC men lyckas inte förhandla det med Palo Alto. Hårdställ istället vad för variant av FEC som ska användas i Nexus-switch.

I Nexus-switch på det fysiska interfacet, ej på länkaggregering, konfigurera följande för att länken ska gå upp:

```
interface Ethernet1/1
  description Palo Alto
  fec rs-ieee
```

Verifiering:

```
nexus# show int eth 1/1 | inc fec
    admin fec state is Rs-IEEE, oper fec state is Rs-IEEE
```

vPC

vPC (Virtual Port Channel) är en teknik som tillåter två Cisco Nexus switchar att se ut som 1 logisk switch, vilket därmed tillåter att de presenterar 1 logisk länkaggregering nedströms och åstadkommer redundans.

vPC-teknik

Två switchar (max 2) bundlas i en vPC domän. Mellan switcharna finns en vPC Peer Keepalive Link för att kontrollera att ens granne är online. Mellan switcharna finns även en vPC Peer-link. Länken är L2. Protokollet CFS (Cisco Fabric Services) används för att synkronisera data mellan switcharna. Exempelvis så synkas MAC-adress table, BPDUer, ARP, ND och vPC konfiguration. Man kan lägga till i vPC domain konfigurationen så att IPv4 ARP och IPv6 ND synkroniseras när vPC kommer upp. Per default sker synkningen inte som en bulk-synkning när länken kommer upp.

En switch i vPC domänen är Primary och en är Secondary. Rollerna påverkar inte skyffling av trafik, båda switcharna är aktiva. Kontrollplanmässigt så utför primary vissa uppgifter som secondary inte gör.

När switcharna är konfigurerade för vPC så skapas port-kanaler med interface i båda vPC-switchar nedströms, till exempelvis andra switchar eller hostar. vPC portar till enheter utanför vPC domänen kallas för vPC Member Ports. Om något är inkopplat i enbart en medlem i vPC domänen så kallas den porten för en Orphan port och slutenheten för Orphan Device.

Consistency checks

Det finns två typer av inconsistencies, typ 1 och typ 2. Följande parametrar måste matcha, annars blir det en typ 1 inconsistency:

- STP Mode
- STP VLAN state
- STP Global settings
- LACP Mode
- MTU

Matchar inte ovan nämnda parametrar kommer vPC-portar på sekundär switchen att stängas ned, med det menas alltså porten mot en enhet nedströms.

Följande är typ 2 checkar och bör matcha:

- VLAN Interface (SVI)
- ACL
- QoS
- IGMP
- HSRP

Matchar inte ovan på switcharna kommer ingenting att stängas ned, men trafikflöden blir inkonsekventa, märkliga, konstiga och kanske till och med fel.

vPC Konfiguration

vPC peers använder ett unikt tilldelat vPC domain ID för att automatiskt tilldela ett unikt vPC system MAC-adress. System MAC-adressen används i STP BPDU, LACP BPDU och IGMP advertisements. Man måste använda ett unikt domain ID för samtliga vPC par i en L2-domän.

```
vpc domain 50
  role priority 22000
  system-priority 10000
  peer-keepalive destination 192.13.3.7 source 192.13.3.6 vrf VPC_KEEPALIVE
  auto-recovery
  ip arp synchronize
  ipv6 nd synchronize

! Verifiering
show vpc role ! Inkluderar roller, system mac osv

interface port-channel5
 description vPC keepalive link
 no switchport
 vrf member VPC_KEEPALIVE
 ip address 192.168.70.52/31

interface port-channel6
 description vPC peer link
 switchport mode trunk
 switchport trunk native vlan 1337
 switchport trunk allowed vlan 10,20,1337
 spanning-tree port type network
 vpc peer-link
```

Använd `ip arp synchronize` och `ipv6 nd synchronize` för att synkronisera tabellerna snabbt i vPC paret.

Här följer några best practices för vPC:

- Använd unika vPC Domain ID:s för att undvika att MAC-adresser överlappar varandra
- Använd dedikerade point-to-point länkar för vPC Peer Link (om man inte använder sig av VXLAN Fabric Peering)
- Använd en management switch för vPC Peer Keepalive länkens kommunikation
- Använd vPC `peer-gateway` för snabbare skyffling av trafik. Trafik med destinations MAC-adress tillhörande den andra vPC switchen kan hanteras av sin granne
- Använd vPC `peer-switch` för att optimera BPDU hantering då det blir ne logisk L2-switch
- Distribuera interface i port-kanaler över olika linjekort på samma chassi
- Skapa en karta för oversubscription i datacentret. En pra princip är 20:1 i access och 2:1 i core.

vPC Delay Restore & vPC Orphan Port Delay Restore

vPC Delay Restore är en teknik som gör att vid uppstart kommer ej vPC-portar nedströms upp förrän efter viss tid. Best practice är att vPC Delay Restore är igång. Är igång per default med 150 sekund. Vill man ändra värdet gör man det i vpc-domänkonfigurationsläge med `delay restore <1-3600>`.

vPC Oprthan Port Delay Restore utför samma funktion som ovan fast för Orphan Ports. Default är samma som vPC delay restore time. Kan justeras i vpc-domänkonfigurationsläge med `delay restore orphan-port 0-300`.

Border Gateway Protocol (BGP)

BGP är ett protokoll som används för att byta ut rötter. Routrar konfigureras med tillhörighet i ett autonomt system (AS). Grannskap mellan AS konfigureras statiskt (minst 1 sida, andra kan vara konfigurerad att lyssna efter grannskapsförsök inom ett IP-spann) och kommunikation sker över TCP 179.

Konfiguration

Konfigurationsexempel

```
router bgp 65337
  router-id 10.10.10.10
  log-neighbor-changes
  address-family ipv4 unicast
    network 10.20.30.0/25
    aggregate-address 10.20.0.0/16 summary-only
  address-family ipv6 unicast
    network 2001:db8::/64
    aggregate-address 2001:db8::/48 summary-only
  neighbor 10.60.60.1
    bfd
    remote-as 65999
    description *** eBGP ***
    password 3 asdfasdfasdasfasd
    update-source loopback1
    ttl-security hops 1
    timers 5 20
    address-family ipv6 unicast
      next-hop-self
      route-map BGP_ROUTE_MAP in
```

```
    route-map BGP_ROUTE_MAP out
    unsuppress-map UNSUPPRESS_MAP
    maximum-prefix 10000 75 restart 1
```

Unsuppress map

En unsuppress map används för att kunna annonsera specifika prefix när man annonserar en summering med **aggregate-address**. Det är en vanlig route-map som förmodligen pekar på en prefix-lista.

```
route-map UNSUPPRESS_MAP permit 10
  match ip address prefix-list UNSUPPRESS_IPV4


ip prefix-list UNSUPPRESS_IPV4 seq 5 permit 10.0.0.0/8 le 32
```

TTL-Security

TTL-Security är en säkerhetsfunktion för att skydda mot spoofade BGP-attacker. TTL security konfigureras med antal hops till grannen. Vid ett hop blir TTL 255, allting lägre än det droppas. 2 blir 254 und so weiter.

AS-Path Filter

Filter kan användas för att verifiera as-path innan installation av rötter sker. AS-path acces-listor med regex används.

Exempel:

Konf	Förklaring
ip as-path access-list 10 seq 5 permit "^$"	Matha enbart prefix som annonseras från lokala routern
ip as-path access-list 20 seq 5 permit "^1337_"	Senaste AS måste vara 1337
ip as-path access-list 30 seq 5 permit "^1257$"	Annonserande AS måste vara 1337

Default-originate

Det går att alltid annonsera default-rutt via BGP. Det konfigureras genom:

```
router bgp 1337
  neighbor 1.2.3.4
   address-family ipv4 unicast
    default-originate route-map OUTGOING_ROUTE_MAP
```

Det är bra att ha med en route-map här, samma som man använder för grannen i övrigt. Detta då default-originate annars hanteras separat från resten av prefixen i BGP-annonseringen, så eventuella åtgärder (såsom prepend) slår ej om man inte har med route-mapen.

MACSEC

https://www.cisco.com/c/en/us/td/docs/dcn/nx-os/nexus9000/102x/configuration/Security/cisco-nexus-9000-nx-os-security-configuration-guide-102x/m-configuring-macsec.html

I MKA policyn kan `should-secure` eller `must-secure` specifiseras. Should-secure innebär att länken kan gå upp och trafik skyfflas utan eller innan MACSEC förhandlas. Must-secure innebär att MACSEC måste etableras innan länken går upp och trafik kan skyfflas över den.

Konfigurationsexempel

```
1 Aktivera feature
feature macsec

2. Skapa PSK
key chain 1 MACSEC_KEY no-show
 key 1000
  key-octet-string 64_TECKEN_HEX cryptographic-algorithm AES_256_CMAC
  send-lifetime 00:00:00 May 13 2024 duration 2147483646

3. Konfigurera MACsec policy (MKA)
macsec policy MKA_POLICY
 cipher-suite GCM-AES-256
 key-server-priority 0
 security-policy should-secure
 window-size 1000
 sak-expiry-time 300

 4. Konfigurera på interface
 interface Ethernet1/12
  macsec keychain MACSEC_KEY policy MKA_POLICY
```

Verifiering

- show key chain name
- show macsec mka session [interface type slot/port] [detail]
- show macsec mka session details
- show macsec mka summary
- show macsec policy [policy-name]
- show running-config macsec

SNMP

SNMPv2

Konfiguration

```
snmp-server contact Jehrlander
snmp-server location Vidavarlden
snmp-server user SNMP_READUSER network-operator auth sha-512 xxxxxx priv aes-128 xxxxx localizedV2key
```

snmpwalk / OID / MIB

Det går att övervaka specifika MIB:ar via SNMP för att punktbevaka exempelvis ett interface.

Först behöver man hitta OID-strängen för det interfacet, gör det med snmp-walk:

```
snmpwalk -Os -c v2communitystring -v 2c sw-nxos.jehrlander.net .1.3.6.1.2.1.2.2.1 | grep Ethernet1/45
ifDescr.436230144 = STRING: Ethernet1/45
```

Nu har vi strängen 436216320 som är unik för Eth1/45. Då kan vi få ut samtliga värden för det interfacet med `snmpwalk -Os -c v2communitystring -v 2c sw-nxos.jehrlander.net .1.3.6.1.2.1.2.2.1 | grep 436230144`.

Lägg till en extra siffra på slutet, efter .2.1, ex. .8 för att enbart se `ifOperStatus`. Man får prova sig fram.

SNMPv3

Bra template saxat från reddit:

```
! SNMP Roles
role name snmp-read-only
  description SNMP read-only role
```

```
  rule 1 permit read feature snmp
!
! SNMP Access List
ip access-list {{ACCESS_LIST}}
 permit ip host {{IP}} any
 permit ip {{SUBNET}}/{{MASK_PREFIX}} any
!
snmp-server globalEnforcePriv
!
snmp-server user {{SNMP_READ_ONLY_USER}} snmp-read-only pwd_type 6 auth sha {{SHA_KEY}} priv
{{PRIV_KEY}}
snmp-server user {{SNMP_READ_ONLY_USER}} use-ipv4acl {{ACCESS_LIST}}
!
```

Förmodligen vill man använda bättre krypteringsalgoritmer, så använd `auth sha-256` och `priv aes-256`.

ACL

Värt att komma ihåg om ACL logging på NX-OS (release 10.3(5) och allting tidigare):

- IPv6 ACL logging is not supported for egress PACL.
- IPv4 ACL logging in the egress direction is not supported.

https://www.cisco.com/c/en/us/td/docs/dcn/nx-os/nexus9000/103x/configuration/security/cisco-nexus-9000-nx-os-security-configuration-guide-103x/m-configuring-ip-acls.html

Accesslistor som innehåller många portar (ex. gt) både som source och destination tar upp väldigt mycket minne i TCAM. Så om möjligt undvik ACEer enligt detta exempel:

```
100 permit tcp 192.168.37.0 0.0.0.255 gt 1024 any gt 1024
```

EVPN

Ethernet VPN är en L2VPN teknik som ska lösa de brister som finns i VPLS. Information om MAC-adresser transporteras via en egen BGP AFI.

EVPN inkluderar Ethernet over MPLS och Ethernet over VXLAN. Den sistnämnda är populär i nutida datacenterimplementationer. Jag skriver om EVPN med VXLAN här.

En variant av EVPN som finns är PBB-EVPN (provider backbone).

EVPN med VXLAN

BGP är kontrollplan och VXLAN är dataplan. RFC 7348, RFC 7432 & RFC 8365. MAC eller MAC och IP enkapsuleras i UDP 4789. Trafik enkapsuleras och de-enkapsuleras av VTEP:ar.

Begrepp

- **BUM** - Broadcast, multicast eller unknown unicast.
- **EVI** - Ethernet VPN Identifier.
- **ES** - Ethernet Segment.
- **ESI** - Ethernet Segment Identifier. Varje ES identifieras via en ESI. 10 bytes stort.
- **I-SID** - Instance Service Identifier. En identifierar som informerar en router hur L2-paketet från kunden ska mappas. 24 bytes stort värde.
- **VNI** - Virtual Network Instance. En logiskt nätverksinstans som hanterar L2 eller L3 tjänster och definierar en L2 broadcastdomän
- **VNID** - Virtual Network Identifier. 24 bitar stort segment ID, tillåter upp till 16 miljoner segment. VLAN mappas till VNID
- **VTEP** - Virtual tunnel endpoint. Routrar/switchar. Är i verkligheten i en loopbackadress.
- **NVE** - Network Virtualization Edge. Enhet som implementerar L2 och/eller L3 virtualisering

VXLAN

VXLAN är en teknik där Virtual Network Identifiers (VNID) ersätter identifieringsfunktionen för ett segment när paket transporteras. Den rollen har VLAN i klassiska Ethernetnät. En VNID är 24 bitar stort vilket tillåter upp till 16 miljoner segment i en domän vilket är en stor skillnad mot maxantalet 4096 i klassiska Ethernetnät.. VLAN används fortsatt för lokal kommunikation på en switch. VNID mappas mot VLAN i konfigurationen på en switch vilket gör att VLAN ID:t enbart är relevant lokalt

på switchen. Se exempekonfiguration där två olika VLAN på två switchar med samma VNID kommer att kunna kommunicera över L2:

```
feature vn-segment
feature nv overlay
feature vn-segment-vlan-based

! Löv 1
vlan 10
 vn-segment 301010

! Löv 2
vlan 20
 vn-segment 301010
```

Underlay

I EVPN (och andra VPN-nät såsom MPLS) används Underlay och Overlay. Underlay används för att skapa routinggrannskap mellan switchar i EVPN-domänen. Overlay är där den faktiska trafiken går. I Overlay går trafiken i olika Tenants, det vill säga i olika VRF:er. Paket som ska transporteras enkapsuleras med en VXLAN header och en extra IP header. Den extra IP headern innehåller source och destinations IP-adresser som finns i underlay:

image.png

IS-IS, OSPF, iBGP eller eBGP används för att skapa routinggrannskap i underlay. Oavsett om man peerar över IPv4 eller IPv6 kan trafik över båda protokoll skyfflas i overlay då de enkapsuleras med IP-adresserna som används i underlay. OSPF är nog det vanligaste protokollet, i exempel nedanför har jag valt att använda IS-IS då det är det mest resurseffektiva protokollet.

Adresserna som används som source och destination mellan switchar i underlay är ett (eller två) loopback interface som kallas för VTEP, virtual tunnel endpoint. Själva switchen är inte en VTEP, det är loopback interfacet som är det, men folk kallar ofta själva switchen för en VTEP. Best practice är att man har ett loopback interface för router ID, ett loopback interface för VTEP och vid behov ett extra loopback interface för Split Loopbacks om det används (se mer information i sektionen om Split Loopbacks).

Man behöver bara annonsera loopbacks/VTEPs in i underlay-protokollet. Det går lätt att åstakomma med OSPFs Prefix Suppression eller IS-IS advertise passive only. Det går att de faktiska länkarna mellan switchar (mellan löv och spines) är onumrerade (ip unnumbered). Om man använder multicast för hantering av BUM-trafik måste PIM-feature vara aktiverad och konfigurerad.

```
feature bfd
feature isis
feature pim

key chain ISIS-PASSWORD
 key 0
  key-string SYKE

router isis ISIS
 bfd
 net 49.0001.0000.0000.000x.00
 is-type level-2
 log-adjacency-changes
 authentication-type md5 level-2
 authentication key-chain ISIS-PASSWORD level-2
 address-family ipv4 unicast
  advertise passive-only

interface loopback 0
 desc ROUTER-ID
 ip address 198.18.0.x/32
 ip router isis ISIS
 isis passive-interface level-2

interface loopback 1
 description VTEP
 ip address 198.18.2.x/32
 ip router isis ISIS
 isis passive-interface level-2
 ip pim sparse-mode

interface eth1/1
 ip unnumbered loopback 0
 description spine-1 <-> leaf-1
 no shutdown
 ip router isis ISIS
 isis bfd
 isis network point-to-point
 isis circuit-type level-2
 medium p2p
```

```
 isis authentication-type md5
 isis authentication key-chain ISIS-PASSWORD
 mtu 9216
 ip pim sparse-mode
```

När underlay routing är etablerad så ska man etablera BGP peering. Detta då kontrollplansinformation om var rutter och hostrutter finns utbyts via BGP i adressfamiljen EVPN (adressfamilj 70). Spines ska vara BGP route-reflektorer, alltså har spines peering-sessioner med samtliga anslutna löv och löv peerar enbart med spines.

```
! Spinekonfiguration

router bgp 65000
 router-id 198.18.0.0
 log-neighbor-changes
 template peer BGP-TEMPLATE-LEAFS
  remote-as 65000
  update-source loopback 0
  password SYKESYKE
 address-family l2vpn evpn
  send-community extended
  route-reflector-client
 neighbor 198.18.0.1
  inherit peer BGP-TEMPLATE-LEAFS
 neighbor 198.18.0.2
  inherit peer BGP-TEMPLATE-LEAFS

! Lövkonfiguration

router bgp 65000
 router-id 198.18.0.x
 log-neighbor-changes
 template peer BGP-TEMPLATE-SPINES
  remote-as 65000
  update-source loopback 0
  password SYKESYKE
 address-family l2vpn evpn
  send-community both
 neighbor 198.18.0.0
  inherit peer BGP-TEMPLATE-SPINES
```

Det går att konfigurera spines unika Loopbackadresser som RP, men mest effektivt är att använda en flytande Anycast RP. Ryggradsswitcharna har då ett Loopback-interface med samma IPv4-adress.

Se konfiguration:

```
! Löv & ryggradsswitchar som ej är RP
ip pim rp-address 198.18.0.0 group-list 224.0.0.0/4

! Ryggrad, RP
interface loopback 3
description ANYCAST RP
ip address 198.18.10.0/32
ip router isis ISIS
isis passive-interface level-2
ip pim sparse-mode

ip pim rp-address 198.18.0.0 group-list 224.0.0.0/4
ip pim anycast-rp 198.18.0.0 198.18.0.1
ip pim anycast-rp 198.18.0.0 198.18.0.2
```

Nu har man en fungerande underlay.

Overlay

För att få L2-kommunikation att fungera behövs inte all funktion nedanför, men man vill ju routa också så jag tar med samtlig konfiguration.

Har man ännu inte mappat VLAN mot VNI:er så är det hög tid att göra det. Portar mot slututrustning fungerar precis som i klassiskt Ethernetnät, trunkar mot enheter som kan tagga VLAN och otaggat/accessportar mot de som inte kan eller ska tagga själva.

```
vlan 10
 vn-segment 301010

interface eth 1/9
 desc PC VRF A VN 301010
 no shutdown
 switchport
 switchport mode access
```

```
 switchport access vlan 10
```

Om man vill byta det VNI som ett VLANredan är mappat mot bered er på trafikstörning. Accessportar på VLANet kommer att flappa och för trunkar/vPC kommer VLANet att bli suspendat ett tag.

När SVI VLAN 10 skapas senare så vill man att det ska kunna routas. Då måste det tilldelas en VRF. Den här VRF:en behöver tilldelas ett VNI (och i slutändan därmed ett VLAN) som används för L3-kommunikation, alltså L3VNI:et. Det här L3VNI:et mappas alltså till ett VLAN som också ska ha ett SVI. SVI:et ska inte ha en IP-adress men ska ha kommandot `ip forward`. Det kallas ibland för Core Facing VLAN.

```
vrf context VRF-A
 vni 202020
 rd auto
 address-family ipv4 unicast
  route-target both auto
  route-target both auto evpn
 address-family ipv6 unicast
  route-target both auto
  route-target both auto evpn

vlan 2500
 vn-segment 202020

interface vlan 2500
 no shutdown
 mtu 9216
 vrf member VRF-A
 no ip redirects
 ip forward
 ipv6 address use-link-local-only
 no ipv6 redirects
```

Har man inte med `ipv6 address use-link-local-only` så kan inte IPv6 forwardas över ett IPv4-underlay.

VNI:et är unikt per tenant. Därför behövs alltså ett VNI, ett VLAN och ett SVI per tenant för att kunna routa trafik inom tenanten.
RD och RT skapas med fördel automagiskt. Det finns mig veterligen ingen bra anledning att sköta den hanteringen manuellt i EVPN.

Ett NVE (Network Virtualization Edge) interface måste skapas. Hänvisning ska ske till VTEP-interface (loopback) för enkapsulering och de-enkapsulering av trafik. Vilket kontrollplansprotokoll

som skall användas konfigureras här. I NVE interfacet måste samtliga VNI:er som används populeras. De som används som L3VNIer måste specifiseras. Man konfigurerar här vilken multicastgrupp som ska användas för hantering av BUM-trafik, om man använder multicast för hantering av BUM.

Det går att konfigurera ARP suppression globalt eller per-VNI, användning av ARP-suppression är rekommenderat. Observera att det kräver TCAM-carving.

Samtliga VNI:er måste läggas till under EVPN-konfigurationsläge.

```
interface nve1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 global suppress-arp
 member vni 202020 associate-vrf
 member vni 301010
  mcast-group 239.0.0.1
  suppress-arp disable

ERROR: Please configure TCAM region for Ingress ARP-Ether ACL before configuring ARP supression.
hardware access-list tcam region arp-ether size double-wide

! tcam-size —TCAM size. The size has to be a multiple of 256. If the size is more than 256, it has to be a multiple of 512.
! Configuring the hardware access-list tcam region arp-ether size double-wide command is not required for Cisco Nexus 9200, 9300-EX, and  9300-FX/FX2/FX3 and 9300-GX platform switches.



evpn
 vni 301010 l2
   rd auto
   route-target both auto
```

Nu är det dags att till sist skapa SVI:et som enheter kommer ha som next-hop/default gateway. De ska vara mappade till det lokala VLAN-ID:t, ej VNID (vilket inte är möjligt ändå). Men innan det behöver man specifisera en MAC-adress för Distributed Anycast Gateway (DAG). Detta då SVI:erna kommer att finnas på flera, i många fall samtliga, löv behöver de dela MAC-adress.

```
fabric forwarding anycast-gateway-mac abba.cafe.1337


interface vlan 10
```

```
 no shutdown
 mtu 1500
 vrf member VRF-A
 ip address 10.10.10.1/24
 fabric forwarding mode anycast-gateway
 no ip redirects
```

Typ 2 rutterna propageras redan i nätet. Men vi vill även ha typ 5 då vi har ett SVI. Detta kan man enkelt lösa genom att redistribuera connected i adressfamiljen för respektive VRF i BGP:

```
 route-map REDISTRIBUTE_CONNECTED permit 10


router bgp 65000
 vrf VRF-A
  address-family ipv4 unicast
   redistribute direct route-map REDISTRIBUTE_CONNECTED
   maximum-paths ibgp 2
  address-family ipv6 unicast
   redistribute direct route-map REDISTRIBUTE_CONNECTED
   maximum-paths ibgp 2
```

Nu bör L3-forwarding och L2-forwarding inom en EVPN-domän att fungera.

För att få kommunikation med omvärlden, eller en annan routingdomän, behöver en av löven axla rollen som border-löv. I större när har man dedikerade löv för detta men det går bra att kombinera med ett löv som har hostar anslutna till sig. Ett länknät behövs per VRF. Önskvärt är att BGP används, men det går att använda ett annat protokoll och sedan redistribuera rutter åt båda hållen. Ej rekommenderat. Peeringen sker över subinterface som är tilldelade respektive VRF. Rekommenderat är att inte annonsera ut hostrutter den här vägen utan att använda sig av filtrering så att inga /31or (och /128or) annonseras.

```
ip prefix-list OUTSIDE_IN permit 0.0.0.0/0
ip prefix-list INSIDE_OUT permit 0.0.0.0/0 ge 8 le 31


route-map OUTSIDE_IN permit 10
 match ip address prefix-list OUTSIDE_IN


route-map INSIDE_OUT permit 10
 match ip address prefix-list INSIDE_OUT


interface eth 1/1
 no shutdown
```

```
  description OUTSIDE
  mtu 9216

 interface eth1/1.1500
  no shut
  encapsulation dot1q 1500
  description VLAN 1500, eBGP VRF-A till utsida
  mtu 9216
  vrf member VRF-A
  ip address 10.200.201.11/29

 router bgp 65000
  vrf VRF-A
   neighbor 10.200.201.9
    remote-as 665
    password outside
    update-source ethernet 1/1.1500
    address-family ipv4 unicast
     route-map OUTSIDE_IN in
     route-map INSIDE_OUT out
```

BGP rutter / EVPN Route-Types

Ett antal typer av BGP rutter används i EVPN AFI.

Route typ	**Beskrivning**
0. Reserverad	
1. Ethernet active discovery (AD) route	Används för att signalera tillbakadragande av MAC-adresser, aliasing och split-horizon etiketter.
2. MAC/IP advertisement route	Innehåller MAC-adresser med EVPN ESI:er och MPLS etiketter. Måste innehålla en MAC-adress men kan innehålla IP-adress.
3. Inclusive multicast route	Innehåller attributer för att representera ingress replication.
4. Ethernet Segment (ES) route	

5. IP Prefix Route	En route för ett helt nät.
6. Selective Multicast Ethernet Tag Route (SMET)	Rutter för multicast,
7. IGMP Join Synch Route	
8. IGMP Leave Synch Route	
9. Per-Region I-PMSI A-D route	
10. S-PMSI A-D route	
11. Leaf A-D route	
12-255. Unassigned	

Typ 2

En typ 2 innehåller information tillhörande en host. En route typ 2 måste innehålla:

- MAC Address (/48)
- MPLS Label1 (L2VNI)
- Route Target for MAC-VRF

Den kan innehålla:

- IP Address (/32 eller /128)
- MPLS Label2 (L3VNI)
- Route Target för IP-VRF
- Router MAC

NVE lär sig IP-attribut genom ARP eller ND. Typ 2 importeras i en viss VRF/MAC-VRF.

Ex. om VNI är 551337 används för L2-segmentet kan route-target för L2/MACVRF se ut enligt 65500:551337. Om L3VNI 31337 används kan route-target för IP-VRF/L3 att se ut enligt 65500:31337.

Typ 5

En route typ 5 måste innehålla:

- IP Prefix
- MPLS Label (L3VNI)
- Route Target för IP VRF
- Router MAC

Den kan även innehålla Gateway-IP. NVE lär sig IP-attribut genom redistribuering eller routing-protokol.

Symmetric/asymmetric IRB

När ett paket transporteras mellan två VXLAN/EVPN routrar så märks paketet med ett VNID (Virtual Network Identifier). Beroende på vilken leverantör (och kanske konfiguration?) så används antingen symmetrisk eller asymmetrisk IRB.

Med asymmetrisk IRB så används taggas det routade paketet med VNID:t som används för det lokala segmentet på routern som paketet routas till. För att det ska fungera måste VNID:t finnas definierat på både den lokalswitchen och fjärrswitchen. I stora nät där det kan finnas enormt många VNI:er så skalar det här dåligt.

Med symmetrisk IRB så används samma, utpekade, VNI för att tagga paketen. I nve-interface konfigurationen pekas VNI ut. Kallas för transit L3 VNI. En transit L3 VNI används per tenant (VRF).

BGP NLRI innehåller route-target, ex. 65001:50000, där den andra delen (50000) är vilket VNI som ska användas för enkapsulering av paketet.

Cisco NXOS använder sig av symmetrisk IRB. Om ett paket routas mellan olika nät, ex. från 192.168.1.20/24 till 172.16.10.20/24, kommer L3VNI att användas för att tagga VXLAN-paketen, alltså transit L3 VNIn som är konfigurerad för tenaten. Om ett paket brygas inom samma nät, från 192.168.1.10/24 till 192.168.1.20/24, kommer L2VNI:n att användas för att tagga paketen. L2VNI:n har i konfigurationen kopplats ihop med ett lokalt VLAN.

Ingress replication

Används ingress replication för att transportera BUM-trafik (istället för multicast) så används typ 3 rutter för att dessa. Grannar går att se med `show nve vni ingress-replication`.

Ingress replication-grannar kan konfigureras statiskt. Se exempel (NXOS):

```
interface nve1
  no shutdown
  source-interface loopback1
  member vni 1000
    ingress-replication protocol static
      peer-ip 203.0.113.2
      peer-ip 203.0.113.3
      peer-ip 203.0.113.4
```

Det går även att konfigurera dynamiskt. Då signaleras intresse av att ta emot trafik via IR över BGP.

```
  member vni 300100
    ingress-replication protocol bgp
```

Det går att kombinera ingress-replication med multicast-replication. Verifiering med `show nve vni ingress-replication`.

ARP Suppression

ARP Suppression används för att minska ARP-trafiken inom en fabric. Närmaste VTEP till sluthosten blir proxy för samtliga ARP-frågor. Aktiveras per L2VNI. Verifiering med `show ip arp suppression-cache { detail }`.

Exempelkonfiguration (NXOS):

```
interface nve1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 member vni 202020 associate-vrf
 member vni 301010
  mcast-group 239.0.0.1
  suppress-arp
```

Switchen kommer då att snoopa efter ARP-frames och bygga Typ 2-rutter som populeras till grannar. Om en ARP-fråga kommer till en switch skickas den inte vidare till hosten utan switchen svarar direkt på frågan med sluthostens MAC-address. Alltså fungerar det inte som proxy-arp där switchen skulle svara med sin egen mac-address.

TCAM carving krävs för att ARP Suppression ska fungera.

IGMP Snooping

Användning av IGMP Snooping i EVPN-domänen är önskvärt. Det stöds enbart om BGP används som kontrollprotokoll, ej flood and learn. Aktiveras med `ip igmp snooping vxlan`. Det går även att konfigurera `ip igmp snooping disable-nve-static-router-port`
vilket gör att trafik i overlay enbart skickas till intressenter, alltså de som redan har skickat en IGMP Membership Report. Default är att trafiken repliceras.

Exempelkonfiguration:

```
ip igmp snooping vxlan
ip igmp snooping disable-nve-static-router-port

advertise evpn multicast

vlan configuration 13
  ip igmp snooping querier 1.1.1.1
vlan configuration 37
  ip igmp snooping querier 2.2.2.2
```

`advertise evpn multicast` gör att Type 6 EVPN Selective Multicast (SMET) Routes skapas, vilket optimerar skyfflandes av multicasttrafik inom fabriken. SMET-rutter är typ 6 i EVPN/VXLAN. För varje VLAN behöver en querier pekas ut.

DHCP Relay

Finns DHCP i nätet måste DHCP Relay konfigurereas. Då DHCP-frågan kan komma från olika löv och svarstrafiken måste hitta rätt måste en bra source för frågorna konfigureras, då DAG används kan inte den adressen användas för att proxa-trafiken. Man kan välja vilken VRF som ska användas för att agera DHCP-relay oavsett vilken VRF som frågorna kommer ifrån. Det kanske inte är önskvärt att exempelvis använda VTEP-adressen som source för denna då man isåfall måste routa ut från underlay.

För att det här ska fungera med ex. Infoblox DHCP-server vill man inte använda Ciscos egna implementation av DHCP option 82, utan industristandard suboption link-selection (sub-option 5) som sätter GiAddr-fältet i DHCP-frågan till loopbackens IP-nummer.

```
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface loopback 3
 vrf member VRF-A
 ip address 10.10.10.10/32
 ipv6 address 2001:1::1/128

interface VlanlOOl
```

```
 ip dhcp relay address 1.3.3.7 use—vrf VRF-A
 ip dhcp relay source—interface loopback2
 ipv6 dhcp relay source-interface loopback12
 ipv6 dhcp relay address 2001:1::1
```

vPC i EVPN

Ett vPC par kommer att ha en egen VTEP-IP men även en sekondär VTEP-IP. Den sekondära är den delade VTEP-IPn. Se exempelkonfiguration:

```
interface loopback0
 description RID
 ip address 198.18.0.1/32

interface loopback1
 description VTEP
 ip address 198.18.1.254/32
 ip address 198.18.1.1/32 secondary
```

Annonsering av samma typ 2 rutter

Ofta kommer båda vPC-switchar att annonsera samma typ 2 rot i EVPN-adressfamiljen. De två olika rutterna kan särskiljas från varandra då Route Distinguisher blir olika per switch. Detta då RD byggs på router ID, vilket skiljer sig från varandra.

Annonsering av statiska rutter (typ 5)

Om en statisk route enbart annonseras av 1 vPC medlem och trafiken dyker upp på den andra vPC-switchen så kan trafiken per default inte skickas över till sin granne. Detta då peer-länken enbart hanterar L2. För att trafik ska fungera finns två val.

Det första är att skapa ett länknät per vrf mellan vPC switcharna som byter ut rutterna via BGP. Det skalar dåligt. Det andra alternativet är advertise-pip som beskrivs nedan.

Ett tredje alternativ är att dubbelansluta så ingen orphan port är inblandad, men det är såklart inte alltid möjligt.

Advertise-pip & advertise virtual-rmac

I ett VPC-par så annonseras nätverken med en virtuell VTEP. Det kan orsaka blackholing om länken till destinationen på en av VPC-medlemmarna går ned. Då rutterna annonseras med den virtuella VTEPen kan båda VPC-medlemmarna ta emot trafiken.

För att undvika blackholingen kan man använda sig av advertise-pip (primary IP). Då annonseras ruttern med switcharna egna VTEP-adresser som next-hop och routen tas tillbaka om länken till destinationen går ned. Det konfigureras i BGP adressfamiljen. Nackdelen är nu att NVE peers ökar, det blir totalt 3 peerings till VPC-par (1 för VPC1, 1 för VPC2 och en för virtuella IPn).

Vill man fortsätta annonsera typ 2 rutter (host) med virtuella VTEP-adressen måste `advertise virtual-rmac` läggas till i konfigurationen för nve interfacet. Best practice är att man konfigurerar båda samtidigt, alltså enligt nedan.

```
router bgp 1337
 address-family l2vpn evpn
  advertise-pip

interface nve 1
 advertise virtual-rmac
```

Fabric Peering

Med Fabric Peering så skapar man en virtuell peer-länk över länkarna som går till spines. Skapas över L3. VTEP IP-adressen ska inte användas för peeringen, ta ut en ny loopback i underlay för fabric peering. CFS (Cisco Fabric Services) trafik bör prioriteras i forwardingen. En port-kanal måste skapas för att fabric peering ska fungera men inga interface ska tilldelas port-kanalen. `port-type fabric` måste läggas till på upplänkarna mot spines.

Peer Keepalive-länken behövs fortfarande. Kan vara out-of-band eller in-band.

Konfiguration:

```
vpc domain 1
 peer-switch
 peer-keepalive 10.10.10.82 source 10.10.10.81
 virtual peer-link destination 10.44.0.4 source 10.44.0.3 dscp 56
 delay restore 150
 peer-gateway
 auto-recovery reload-delay 360
 ipv6 nd synchronize
 ip arp synchronize

interface port-channel1337
```

```
 description vpc-peer-link
 switchport
 switchport mode trunk
 spanning-tree port type network
 vpc peer-link


interface Ethernet1/49
 mtu 9216
 port-type fabric
```

BUM-traffik i vPC

Med ingress replication kommer BUM-trafiken att skickas till den delade anycast VTEP-adressen. `ip arp synchronize` och `ipv6 nd synchronize` bör vara konfigurerat i vPC-domänen för att snabbt sprida informationen.

Används multicast så skickas även trafiken till anycastadressen. Enbart en av vPC medlemmarna är vald som en decapsulation nod. Den eper med lägst cost till RP blir vald till decapsulation nod. Är det samma cost till RP så blir vPC Primary vald.

Vid initiering av BUM-trafik (arp request, neighbor discovery) från vPC-klustret så kommer mottagande medlem att skicka ut multicastfrågan mot RP men även till vPC-granne, även om trafiken först kommer in till secondary vPC medlem. Svarstrafiken kommer att komma till den vPC-medlem som är decapsulation node

vPC Infrastructure VLANs

vPC Infrastructure VLANs används när en vPC switch upplänkar mot spines går ned. Det används som Backup Routing Path och ska finnas i den globala VRFen. Infrastructure VLAN:et ska finnas tillåtet på vPC Peer-link. Ett routinggrannskap måste finnas över vPC Infrastructure VLAN:et.

Konfiguration:

```
vlan 1337
 name VPC_INFRASTRUCTURE_VLAN


interface Vlan1337
 ip router isis ISIS



system nve infra-vlans 1337
```

vPC Delay Restore & vPC Orphan Port Delay Restore

vPC Delay Restore är en teknik som gör att vid uppstart kommer ej vPC-portar nedströms upp förrän underlay och overlay har konvergerats. Best practice är att vPC Delay Restore är igång. Är igång per default med 150 sekunder. Vill man ändra värdet gör man det i vpc-domänkonfigurationsläge med `delay restore <1-3600>`.

vPC Oprthan Port Delay Restore utför samma funktion som ovan fast för Orphan Ports. Default är samma som vPC delay restore time. Kan justeras i vpc-domänkonfigurationsläge med `delay restore orphan-port 0-300`.

SVI Delay Restore

Är per default 10 sekunder. Modifieras i vpc-domänkonfigurationsläge med `delay restore interface-vlan 1-3600`.

NVE Hold-Down timer

Under uppstart vill man inte annonsera sina nät och hostar till resten av EVPN-fabric innan länkar och SVI är redo. Därför ska NVE Hold-Down timern komma sist. Per default är tiden 180 sekunder, vilket är högst om man inte modifierar övriga räknare. Kan ställas in i vpc-domänkonfigurationsläge med `source-interface hold-down <1-1500>`.

Split Loopbacks (Orphan ports, fabric-ready time)

Används för att minska nedtiden för orphan ports. Finns i NXOS 10.4.1 och senare. Olika loopbacks används för anycast VTEP och PIP. För typ 2 orphan rutter och typ 5 rutter minskas konvergeringstiden. Konfigureras i vpc-domänkonfigurationsläge med `fabric-ready time <1-1200>`. Default är tre fjärdedelar av source-interface hold-down-timer, alltså 135 sekunder.

Fullständig konfiguration ser ut såhär:

```
interface loopback1
 description VTEP
 ip address 10.200.200.254/32

interface loopback2
 description Anycast VTEP Split Loopback
 ip address 10.200.200.123/32

interface nve1
 source-interface loopback1 anycast loopback2
```

VXLAN vPC Consistency Checking

Följande ska matcha mellan vPC-medlemmarna:

- VLAN-till-VNI mapping
- Samma SVIer ska finnas på switcharna
- Samma VNI måste användas för samma transportmetod av BUM-trafik
- När VNI använder multicast replication av BUM måste båda VTEPar använda smama multicastgrupp för specifiserad VNI

Om vPC VTEP consistency checken misslyckas kommer NVE loopback interface att bli admin shut på sekondära vPC-switchen.

Tenant Routed Multicast (TRM)

Adressfamilj MCAST-MVPN (SAFI 5) används som kontrollplan för multicast i EVPN-VXLAN.

Det finns två TRM lägen:
Layer 2/3 Mode TRM
Används i en blandad plattform där alla VTEPar inte är kapabla att använda TRM.
Layer 3 Mode TRM
L3 mode implementeras när alla VTEPar är kapabla att använda TRM. Är den vanliga implementationen.

RP behövs i overlay för L3 mode TRM.

Anycast RP
Samtliga VTEPar som deltar i en VRF konfigurerad som RP. Ett loopbackinterface med PIM aktiveras med en delad IP-adress används.

External RP
En PIM-router ansluts till ett border-löv som har ett PIM-interface som är anslutet till VXLAN/EVPN-nätverket. Externa routern kör troligtvis VRF lite och i fabriken har den pekats ut som RP.

RP Everywhere
Används när man migreras från en extern RP till VXLAN Anycast RP. Tillåter att båda modeller fungerar samtidigt.

Konfiguration:

```
feature ngmvpn


! Löv BGP
router bgp 65501
  neighbor 10.100.100.201
    remote-as 65501
```

```
    update-source loopback0
    address-family l2vpn evpn
      send-community both
    address-family ipv4 mvpn
      send-community both
    address-family ipv6 mvpn
      send-community both

! Distribuerad anycast lo
interface loopback250
 vrf member Tenant1
 ip address 10.51.51.254/32 tag 12345
 ip pim sparse-mode



ip multicast overlay-spt-only



!Note We recommend that underlay multicast groups for Layer 2 VNI, default MDT, and data MDT not be shared. Use separate, non-overlapping groups.

interface nve1
 source-interface loopback1
 host-reachability protocol bgp
 member vni 30010
  mcast-group 239.1.1.1

vrf context Tenant1
 ip pim rp-address 10.51.51.254
 vni 50001
 rd auto
 address-family ipv4 unicast
  route-target both auto
  route-target both auto evpn
  route-target both auto mvpn
```

Migrering till EVPN

Förberedelser

Innan migrering från traditionell Ethernet miljö till EVPN/VXLAN vill man byta MAC-adress på SVI:er till den mac-adress som kommer användas av distributed anycast gateway (DAG). Detta så att trafik ska flöda sömlöst när DAG tar över L3-ansvaret.

Är den gamla miljön Cisco Nexus med HSRP så byter man MAC på standby enheten och sen preemptar så den tar över då ett GARP-meddelande skickas vid preempt. Exempelkonf:

```
interface Vlan1337
 hsrp 1
  mac-address dead.cafe.beef
  priority 130
  preempt
```

Byt sedan på f.d. primär efteråt. Har man en IOS/Catalyst-miljö, ex. VSS, så byt mac-adress på SVI:er med kommandot `mac-address cafe.cafe.cafe`.

Default Gateway Coexistence of HSRP and Anycast Gateway

Man kan använda sig av den här tekniken för att ha SVI:erna aktiva i EVPN-miljön och i klassiska miljön samtidigt. Funktionen innebär mer efffektiv routing och mindre störande migreringar.

Tekniken implementeras helt i EVPN-miljön. Ciscos information om detta finns här.

Följande måste konfigureras för att det ska fungera:

- MAC-adress på DAG och SVI måste matcha (se förberedelser ovan)
- TCAL carving behövs utföras för Ingress PACL. Verifiera med `show hardware access-list tcam region`, konfigurera med `hardware access-list tcam region ing-ifacl 512`, starta om
- L2-trunk mellan nya miljön och gamla med `port-type external` konfigurerat på trunken
- Unik BIA (Burnt In Address) IPv4&IPv6 måste konfigureras på samtliga SVIer på border-löv (används för att skicka proxy ARP/ND)
- Om border-lövet är i ett vPC par måste `peer-gateway` och `layer3 peer-router` vara konfigurerat

Exempelkonfiguration:

```
interface port-channel 40
 port-type external

interface vlan 1100
 vrf member vrf50
 ip address 192.168.1.1/24
 ip address 192.168.1.10/24 secondary use-bia
 ipv6 address 2001:DB8:1::1/64
 ipv6 address 2001:DB8:1::10/64 use-bia
```

Erfarenheter

Trafik till default gateway floodas innan migrering

När SVIt ej är migrerat till DAG/SVI i fabriken så tycks trafik till den mac-adressen (SVI/DAG-mac, samma i nya och gamla miljön) floodas till samtliga hostar inom VLANet. Avhjälps när SVIt flyttas till EVPN-fabric.

ACL

Nexus-switchar har MYCKET mindre plats för ACLer än ex. Catalyst-plattformen. Det går inte heller att öka med TCAM-carving hur mycket TCAM som är tilldelat ACL, i alla fall inte på 93600GD-CX. Så tänk en extra gång här och hitta en annan lösning för säkerhet, ex. direktanslutna nät i brandvägg, PBR till brandvägg eller säkerhet på hostnivå.

PBR

En självklarhet för alla förutom jag kanske, men om PBR ska fungera utan loopar så måste man antingen ha ett servicelöv (ej dubblerad, vPC) med brandvägg och PBR på L3-vnier+SVIer närmast hostar eller direktansluta samtliga löv till brandväggar. PBR väljer nästa direktanslutna next-hop.

Konfigurera statiska IPv6-grannar

I NX-OS konfigureras statiska IPv6-grannar på interface, exempelvis:

```
interface Vlan1337
 ipv6 neighbor 2001:13:37:100::1337 0050.5693.efea
```

Det går bra att göra på 1 switch eller 1 vPC-par. Men gör man det på fler vPC-par, eller fler switchar, skapar man en routingloop.

Exempelloggar:

```
2026 Jun  9 11:11:02 sw-leaf-1 %BGP-2-EVPN_STATIC_MAC:  bgp-65005 [18623]  Static/learnt MAC address conflict for 0050.5693.efea in VNI 300300

2026 Jun  9 11:11:02 sw-leaf-2 %ICMPV6-2-REMOTE_STATIC_EXISTS:  icmpv6 [15777]  Remote static exists for,IPv6 = 2001:13:37:100::1337, VLAN = 1337, IOD = 48, Interface = Vlan1337
2026 Jun  9 11:11:02 sw-leaf-2 %BGP-2-EVPN_STATIC_MAC:  bgp-65005 [18620]  Static/learnt MAC address conflict for 0050.5693.efea in VNI 300300
```

Uppgradera NXOS

Kommandot för att installera NXOS-mjukvara är likt det för IOS-XE.

Ex: `install add bootflash:///nxos64-cs.10.3.6.M.bin`

Det är inte rekommenderat, men det går att sätta bootpekaren manuellt, ex: `sw-nxos(config)# boot nxos bootflash:/nxos64-cs.10.3.6.M.bin`. Jag fick göra det när jag fick följande felmeddelande vid försök att installera:

```
sw-nxos# install add bootflash:///nxos64-cs.10.3.6.M.bin
Install operation failed because not enough space is available in /var/cache/dnf/ in the active sup.
```

Maintenance-läge

Inför uppgradering kan man stänga av funktioner, förutom kontrollplan, i switchen.

Enklaste sättet att göra dett är att skriva in `system mode maintenance` i konfläge. Routingprocesser och vPC stängs då ned. Verifiering med `show system mode`.

Snapshot

För att se skillnader kan man skapa snapshots innan och efter uppgradering, eller konfigurationsförändring, för att jämföra konfiguration och information. Skapa snapshot med `snapshot create { name } { description }`. Jämför med `show snapshots compare { snapshot1 } { snapshot2 }`. Man kan även speca efter snapshot2 vad man vill jämföra, exempelvis `ipv4routes` vilket visar skillnader i vilka rötter som finns i switchen.

PBR, Policy Based Routing

I Cisco NX-OS kan man inte använda sig av "deny" statements i en ACL för att undanta trafik från PBR.

Istället får man använda sig av route-map logik, där man först har en accesslista som matchar ett entry som inte utför någonting. Sedan en annan ACL som matchar korrekt där man utför PBR.

Ska man använda PBR i en VXLAN/EVPN-fabric får man tänka till, så att man inte skapar routingloopar. Ett alternativ är att direktansluta destinationen till samtliga löv, exempelvis en brandvägg.

Ett annat är att ha ett servicelöv utan loopar, kanske är svårt med vPC dock.

Erfarenheter

Vid tillägg av nytt SVI i ett VXLAN-EVPN nätverk med PBR på slaget så PBRades inte trafiken. `ip policy route-map NAMN` var applicerat på interfacet och ACL som hänvisades av route-map matchade nätet. Det krävdes en shut/no shut på SVIt för att PBRn skulle fungera.

Det här var i September 2026 på NX-OS version 10.5(4).

Breakout

Ett breakout-interface är ett fysiskt höghastighetsinterface som görs om till flera logiska interface av lägre hastighet.

Inkoppling

I praktiken kopplas enbart 1 speciell SFP (fiber alt. DAC) in i switchen som går till en breakoutkabel. En breakoutkabel delar sig i flera delar. Kabeln kan vara fristående från SFPn eller integrerad i SFPn. Breakout via en fiber-SFP stöder finns för både multimode och singlemode.

Hastigheter

Vilka hastigheter som stöds beror på vilka hastigheter som porten stöder och därmed vilka kanaler (engelska: lanes) som ligger bakom hastigheten. Exempelvis är en 40 Gbit/s-interface egentligen 4x10 Gbit/s, vilket gör att 1 40 Gbit/s-interface kan bli 4x10Gbit/s-interface via breakout.

Rate	Technology	Breakout Capable	Electric Lanes	Optical Lanes*
10G	SFP+	No	10G	10G
25G	SFP28	No	25G	25G
40G	QSFP+	Yes	4x 10G	4x10G, 2x20G
50G	SFP56	No	50G	50G
100G	QSFP28	Yes	4x 25G	100G, 4x25G, 2x50G
200G	QSFP56	Yes	4x 50G	4x50G
2x 100G	QSFP28-DD	Yes	2x (4x25G)	2x (4x25G)
400G	QSFP56-DD	Yes	8x 50G	4x 100G, 8x50G

Konfiguration

Breakout konfigureras ej direkt på interfacet utan i global config-läge.

Modulläge

Det går att konfigurera en hel modul i breakoutläge. Då startar modulen om vid konfiguration.

```
switch# configure terminal
switch(config)# interface breakout module 1
Module will be reloaded. Are you sure you want to continue(yes/no)? yes

! Återställ
no interface breakout module module_number
```

Per-interfaceläge

Ingen omstart av modul sker när det konfigureras i interfaceläge.

```
! ! interface
interface breakout module 1 port 1 map 10g-4xswitch

! flera interface
interface breakout module 1 port 1-4 map 10g-4xswitch

! återställ
no interface breakout module 1 port 1 map 10g-4x

! Beroende på porttyp så syns olika varianter:
C93600CD-GX# show int eth1/32
Ethernet1/32 is down (XCVR not inserted)
admin state is down, Dedicated Interface
  Hardware: 10000/25000/40000/50000/100000/200000/400000 Ethernet, address: 643a.eada.bad0 (bia
643a.eada.bad0)



C93600CD-GX(config)# interface breakout module 1 port 32 map ?
  100g-2x       Breaks out a 200G high BW front panel port into two 100G ports
  100g-2x-pam4  Breaks out a 200G high BW front panel port into two 100G ports with mode pam4
  100g-4x       Breaks out a 400G high BW front panel port into four 100G ports
  10g-4x        Breaks out a 40G  high BW front panel port into four 10G ports
  200g-2x       Breaks out a 400G high BW front panel port into two 200G ports
  25g-4x        Breaks out a 100G  high BW front panel port into four 25G ports
```

50g-2x	Breaks out a 100G high BW front panel port into two 50G ports
50g-4x	Breaks out a 200G high BW front panel port into four 50G ports

Verifiering:

```
switch(config-if)# show int eth1/49 transceiver
Ethernet1/49
transceiver is present
type is QSFP-40G-SR-BD
name is CISCO-AVAGO
part number is AFBR-79EBPZ-CS2
revision is 01
□□□
switch(config-if)# show int eth1/52 transceiver
Ethernet1/52
transceiver is present
type is QSFP-Cazadero
name is CISCO-DNI
part number is CAZADERO-R
revision is 03
nominal bitrate is 10000 MBit/sec per channel
□□□
switch(config-if)# show int eth1/53 transceiver
Ethernet1/53
transceiver is present
type is QSFP-Cazadero
name is CISCO-DNI
part number is CAZADERO-R
revision is 03
nominal bitrate is 10000 MBit/sec per channel
□□□
switch(config)# interface breakout module 1 port 52-53 map 10g-4x
□□□
switch(config-if)# show int br | i up
mgmt0 -- up 10.122.160.192 100 1500
Eth1/49 -- eth routed up none 40G(D) —  Running 40G
Eth1/50 -- eth routed up none 40G(D) --
Eth1/52/1 -- eth routed up none 10G(D) — Broken out to 10G
Eth1/53/1 -- eth routed up none 10G(D) --  Broken out to 10G
```

FEC för breakout

Forward error correction (FEC) behövs på alla kablar förutom 1-2 meter passiva DAC-kablar. Standard i Ciscoswitchar är FC-FEC CL74 men RS-FEC Consortium 1.6, RS-FEC IEEE, och andra FEC-agloritmer kan konfigureras.

För 25gbit/s Ethernet används två primära FEC-algoritmer:

- FC-FEC, felrättning med låg latens under 100 nanosekunder, optimerad för bursts. Används på 3 och 5 meter passiva DAC-kablar och på optiska kablar upp till 10 meter. Används även på 100Gbit/s-interface.
- RS-FEC. Bättre felrättning. Krävs på 25Gbit/s multimode fiber, upp till 100 meter. Kan även behövs på aktiv DAC över 10 meter.

Konfiguration och verifiering

```
switch# (config-if)# fec ?
auto FEC auto
fc-fec CL74(25/50G)off Turn FEC off
rs-cons16 RS FEC Consortium 1.6 (25G)
rs-fec CL91(100G) or Consortium 1.5 (25/50G)
rs-ieee RS FEC IEEE (25G)

switch# show interface fec
-------------------------------------------------------------------------------------------------
Name  Ifindex  Admin-fec  Oper-fec    Status  Speed  Type
-------------------------------------------------------------------------------------------------
Eth1/1   0x1a000000  auto  auto  connected     10G  SFP-H10GB-AOC2M
Eth1/2   0x1a000200     Rs-fec    notconneced    auto  QSFP-100G-AOC3M
Eth1/3/1  0x38014000  auto  auto  disabled  auto  QSFP-H40G-AOC3M
Eth1/3/2  0x38015000  auto  auto  disabled  auto  QSFP-H40G-AOC3M
Eth1/3/3  0x38016000  auto  auto  disabled  auto  QSFP-H40G-AOC3M
Eth1/3/4  0x38017000  auto  auto  disabled  auto  QSFP-H40G-AOC3M
```

Lista över Nexusswitchar

Den här saxades 2026-04-01 från Cisco. Det är bäst att läsa deras dokumentation för relevant info.

Switches	4x10G	4x25G	2x50G	2x100G	2x200G	2x400G	4x50G	4x100G	8x100G
N9K-X9636C-RX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9636C-R	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9636Q-R	Yes	No	No	No	No	No	No	No	No
N9K-X96136YC-R	No	No	No	No	No	No	No	No	No
N3K-C3636C-R	Yes	Yes	Yes	No	No	No	No	No	No
N3K-C36180YC-R	Yes	Yes	Yes	No	No	No	No	No	No
N9K-93108TC-EX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-93180YC-EX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-93108TC-FX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-93180YC-FX	Yes	Yes	Yes	No	No	No	No	No	No

Switches	4x10G	4x25G	2x50G	2x100G	2x200G	2x400G	4x50G	4x100G	8x100G
N9K-9348GC-FXP	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9732C-EX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9736C-EX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9732C-EXM	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9736C-FX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9736Q-FX	Yes	No	No	No	No	No	No	No	No
N9K-X9788TC-FX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-X9732C-FX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-C9348GC-FXP	Yes	Yes	Yes	No	No	No	No	No	No
N9K-C9336C-FX2	Yes	Yes	Yes	No	No	No	No	No	No
N9K-C93216TC-FX2	Yes	Yes	Yes	No	No	No	No	No	No
N9K-C93360YC-FX2	Yes	Yes	Yes	No	No	No	No	No	No

Switches	4x10G	4x25G	2x50G	2x100G	2x200G	2x400G	4x50G	4x100G	8x100G
N9K-C9364C-GX	Yes	Yes	Yes	No	No	No	No	No	No
N9K-C9316D-GX	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-C93600CD-GX	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-X9716D-GX	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-C9332D-GX2B	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-C9348D-GX2A	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-X9400-16W	Yes	Yes	Yes	Yes	No	No	Yes	No	No
N9K-X9400-8D	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
N9K-X98900CD-A	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No

Buffrar, utgående trafik

Switcharna har buffrar för att kunna hantera trafik som direkt inte kan skyfflas ut på ett interface. Kommer det exempelvis trafik på ett 10Gbit/s-interface till ett 1Gbit/s-interface kan problem uppstå, då ser man följande i loggen:

`%TAHUSD-SLOT1-4-BUFFER_THRESHOLD_EXCEEDED: Module 1 Instance 0 Pool-group buffer 90 percent threshold is exceeded! Pl`
`ease see http://cisco.com/go/n9k-tahusd-buffer for more information`

Trafik som slängs pga fulla buffrar syns som en output discard:

```
C9348GC-FXP# show int eth 1/21 | include discard
    0 input with dribble 0 input discard
    0 lost carrier 0 no carrier 0 babble 22136689 output discard
```

Vad för typ av trafik (unicast/multicast) som slängs går att verifiera:

```
C9348GC-FXP# show queuing interface ethernet 1/21

slot  1
=======



Egress Queuing for Ethernet1/21 [System]
------------------------------------------------------------------------------
QoS-Group# Bandwidth%% PrioLevel             Shape               QLimit
                       Min         Max       Units
------------------------------------------------------------------------------
    7          -       1         -         -    -          9(D)
    6          0       -         -         -    -          9(D)
    5          0       -         -         -    -          9(D)
    4          0       -         -         -    -          9(D)
    3          0       -         -         -    -          9(D)
    2          0       -         -         -    -          9(D)
    1          0       -         -         -    -          9(D)
    0        100       -         -         -    -          9(D)
+-------------------------------------------------------------+
|                      QOS GROUP 0                            |
```

```
+-------------------------------------------------------------+
|                       |  Unicast       |Multicast       |
+-------------------------------------------------------------+
|                 Tx Pkts |             0|      846136044|
|                 Tx Byts |             0|    190541523833|
| WRED/AFD & Tail Drop Pkts |             0|       22157259|
| WRED/AFD & Tail Drop Byts |             0|       86338207|
|              Q Depth Byts |             0|             0|
|       WD & Tail Drop Pkts |             0|       22157259|
```

Om det finns olika delar av ASIC att dela upp trafiken på, dvs flytta portar mellan olika slice:ar, går att kontrollera med `show interface hardware-mappings`.

TCAM, TCAM-Carving

Varför man ska behöva allokera resurser manuellt i en DC-switch är ett stort mysterium för mig.

Men för att använda vissa funktioner, ha fler resurser tilldelat till ex. ACL behöver man göra det.

Konfiguration

Efter konfiguration behöver man spara och starta om switchen. Det går att verifiera med `show system config reload-pending`.

Verifieringskommandon

Kommando	**Förklaring**
show hardware access-list tcam region	Se konfigurerad tilldelning
show hardware access-list resource utilization	Se hur mycket av tilldelade resurser som används
show system internal access-list globals	Se mycket, ex. aktuell tilldelning (ej konfigurerad, kan skiljas sig mellan omstart)

Kontrollera faktisk forwarding med packet capture

Nedan gjorde jag i felsökning med Cisco TAC där policy based routing inte fungerade på 10.5(4) och 10.5(6) (buggen i fråga). Felet uppstod endast när enbart 1 interface var kopplat till VLANet, när interfacet gick ned och sedan upp så applicerades ej PBR.

```
! Spela in trafik

debug platform internal tah elam
 trigger init
 reset
 set outer ipv4 src_ip 1.1.1.1 dst_ip 2.2.2.2
 start

 report

SUNDOWN1 ELAM REPORT SUMMARY
slot - 1, asic - 0, slice - 0
==============================

Incoming Interface: Eth1/21
Src Idx : 0x51, Src BD : 300
Outgoing Interface Info: dmod 1, dpid 24
Dst Idx : 0x60d, Dst BD : 2618

Packet Type: IPv4

Dst MAC address: 00:11:C0:FF:EE:00
Src MAC address: 00:F8:2C:C9:21:B1

Dst IPv4 address: 2.2.2.2
```

```
Src IPv4 address: 1.1.1.1
Ver     =  4, DSCP    =    0, Don't Fragment = 0
Proto   = 17, TTL     =  255, More Fragments = 0
Hdr len = 20, Pkt len =   75, Checksum       = 0x5f74

L3 DA SGT = 0, L3 SA SGT = 0

L4 Protocol  : 17
UDP Dst Port : 53
UDP Src Port : 26828

Drop Info:
----------

LUA:
LUB:
LUC:
LUD:
Final Drops:

vntag:
vntag_valid    : 0
vntag_vir      : 0
vntag_svif     : 0
```

Här ser vi då inkommande interface (Eth1/21 0x51) samt utgående interface (0x60d). Man tar reda på vilket interface det matchar genom:

```
show hardware internal tah niv_idx 0x60d
niv_idx  : 1549
Interface: Ethernet1/45
```

Det här var beteende vi väntade. När vi senare slog samma kommandon efter att interfacet hade gått ned och upp visades ett annat utgående interface, vilket då var felaktigt beteende i switchen.