Catalyst

- Catalyst Tips & Trix
- Configuration Rollback
- Option 82, Suboptions
- Patchning Catalyst 6807-XL
- Point-to-Point Protocol
- Spanning-Tree Protocol
- Virtual Switching System (VSS)
- VLAN
- DDNS
- Embedded Event Manager (EEM)
- MACSEC
- EtherChannels / Port-channels / Länkaggregering
- ACL
- Downloadable ACL
- Cisco Discovery Protocol och Link Layer Discovery Protocol
- Policy Based Routing (PBR)
- Routing Information Protocol, RIP
- EIGRP
- Config register
- "Nya" syntax, username, enable secret, password service-encryption
- IPv6
- OSPFv2
- OSPFv3
- IS-IS
- Border Gateway Protocol (BGP)
- Multiprotocol Label Switchning (MPLS)
- EVPN

- Patchning IOS-XE
- Lager 1 (fysiskt)
- NETCONF & YANG
- LISP
- PCAP / Packet Capture i IOS-XE
- Stacking C9000
- Network Address Translation (NAT)
- Send
- Port-security
- kron-jobb
- Logging
- Proxy ARP
- WLC 9800
- TACACS
- SNMP
- SVI
- SSH-nyckel på Cisco-enheter

Catalyst Tips & Trix

Det går med include att specifisera två värden som måste matcha:

```
catalyst9300#sh ip int br | inc Port.*up
Port-channel31         unassigned      YES unset  up                    up
```

Specialtecken: `To type a '?' into IOS, hit Control+V then '?'.`

Configuration Rollback

Cisco IOS-XE har en funktion som gör att konfiguration backas om man ej bekräftar den. I mitt tycke ska den användas för samtliga förändringar, skulle ett misstag ske och management-access tappas till enheten kan den då återställa tidigare konfiguration utan ev. omboot av enheten. Detta ersätter även arbetssätt såsom "reload in x", där man startar om enheten för att återställa konfiguration

Då switchen gör en lokal kopia av konfigurationen måste först archive konfigureras:

```
archive
 log config
  logging enable
  logging size 200
  notify syslog contenttype plaintext
 path flash:
```

När man sedan önskar gå in i konfigurationsläge så slå följande:
`configure terminal revert timer 10`
10 är minuter och kan konfigureras med vilket värde som helst mellan 1 och 120.

När man är klar med konfigurationen måste kommandot `configure confirm` slås i enable-läge, då avbryts rollback-klockan.

Önskar man backa konfiguration i förtyg kan man använda `configure revert now` i enable-läge.

Option 82, Suboptions

När ett DHCP-discover paket forwardas av en DHCP Relay agent så sätts GiAddr-fältet till den adress som är i samma nät som DHCP-klienten. Det är så DHCP-servern identifierar vilket nät den ska dela ut adresser i.

Om man då använder sig av en annan adress som source, ex. en loopback adress, så måste informationen om vilket subnät följa med på något annat sätt då GiAddr blir loopback-adressen. Det gör man genom DHCP Option 82 och dess suboption Link-selection.

Cisco IOS-XE försöker per default att använda sig av en Cisco propertiär variant av suboption Link-selection, suboption 150. Infoblox förstår inte suboption 150, därmed måste industristandarden suboption 5 användas. För att ändra till suboption 5 i Cisco IOS-XE använd det globla konfigurationskommandot `ip dhcp compatibility suboption link-selection standard`.

Patchning Catalyst 6807-XL

För ISSU uppgraderingar följ denna guide. ISSU är det enda sättet att uppgradera utan avbrott.

Har man flera boot statements är det den översta i konfigurationen som gäller.

Skulle det dock vara så att man vill hoppa många steg snabbt så följ nedanstående instruktioner, observera att under uppgraderingen när man kör RPR så kommer det vara 3-5 minuters avbrott när man gör överslag mellan active och standby:

We can upgrade the standby individually directly to 15.5(1)SY12 from 15.5(1)SY1 since 6K supports RPR then we can change the boot variable on the active and do a switchover. This is an easier procedure; however, it may only work with no downtime if you have dual homed connections between both active and standby.

1. Set the boot system to 15.5(1)SY12 on the active which will reflect on both chassis.
2. Hit “redundancy reload peer” on the active, where the standby will be upgraded and reloaded to the new IOS.
3. After the standby comes up, we can then hit “redundancy force-switchover” where the active will now be reloaded and upgraded however when it comes up the old standby will be active now.
4. As discussed earlier, if you have dual homed connections, we should not see an impact as there will always be a chassis that is up.
5. Since 6K supports RPR, then it will run with different images, however it will not be running SSO at the moment of different images.

The concept is that the switches will run RPR at the time they are running different IOS releases, ensuring continuous operation for both switches.

While SSO and RPR differ primarily in failover time and a few other aspects, the presence of dual-homed connections should mitigate any noticeable impact, as previously discussed.

The switches fall back to RPR once there is a mismatch in the IOS. To clarify, even when SSO is configured there is no need for any manual configuration change.

Det går även att bryta klustret om man har helt reduntanta förbindelser nedströms. Töm sekundära sidan på kablar, uppgradera, töm aktiva sidan på kablar och sätt i alla kablar i sekundären. När sekundären har tagit över all trafik, uppgradera den f.d. aktiva, boota och verifiera, stäng av och koppla i samtliga kablar och boota den.

Point-to-Point Protocol

PPP är ett protokoll för direktkommunikation mellan två enheter utan någon annan enhet eller nätverksmanipulation mellan enheterna.

PPPoE

PPP over Ethernet möjliggör PPP över Ethernet. PPPoe virtualiserar Ethernet för att kunna köra flera PPP sessioner över samma segment. Beskrivs i RFC 2516.

I ett DSL-nät, och även över mobilnätverk, så kan en Cisco-router få en dynamisk tilldelad IP-adress genom att använda sig av IP Configuration Protocol, ett subprotokoll till PPP, genom kommandot `ip address negotiated` på interfacet.

PPPoE lägger till 8 bytes overhead så MTU på PPPoE interface sätts ofta ned till 1492 så att allt får plats i ett 1500 byte Ethernet frame. MSS för TCP sätts ofta ned till 1452.

Exempelkonfiguration:

```
interface FastEthernet0/0
 ip nat inside

interface FastEthernet0/1
 pppoe-client dial-pool-number 1

interface dialer1
 mtu 1492
 ip tcp adjust-mss 1452
 encapsulation ppp
 ip address negotiated
 ppp chap hostname Username
 ppp chap password Password
 ip nat outside
 dialer pool 1

ip route 0.0.0.0 0.0.0.0 dialer1
```

Verifiering av PPPoE kan ske med kommandot `show pppoe session`. Det går även att använda debugs med `debug pppoe [ data | errors | events | packets ]`.

Spanning-Tree Protocol

STP är ett nätverksprotokoll som används för att elliminera loopar i L2 nät. Artikeln beskriver först klassisk spanning-tree och sedan nyare varianter. I STP så beskrivs switchar som bryggor.

Beskrivning av STP

STP gör detta genom att se till att vilka interface ej tar emot eller skickar trafik. Dessa portar hamnar i ett läge som kallas **Blocking**. Portar som fortfarande skickar och tar emot trafik är i liget **Forwarding**.

STP protokollmeddelanden kallas för Bridge Protocol Data Units (BPDU). Det finns två typer av BPDUer, Configuration BPDU (typ 0x00) och Topology Change Notification BPDU (typ 0x80). En Configuration BPDU kan även kallas för **Hello BPDU** eller helt enkelt **Hellos**.

Se bild på vad som ingår i en BPDU:

image.png

Switchar och portar identifieras baserat på sitt ID i BPDUn. ID:t har två delar, en del som kan programmeras som är *priority* och en som är statisk som ej kan konfigurereas. Både switchar och portars prioritet kan konfigureras.

STP baseras på möjligheten att jämföra två eller flera Configuration BPDUs och välja den som är bättre - den som är **superior**. För att räkna ut vilken som är superior letar STP efter det först förekommande lägsta värdet av följande parametrar från top till botten:
- Root Bridge ID (RBID)
- Root Path Cost (RPC)
- Sender Bridge ID (SBID)
- Sender Port ID (SPID)
- Receiver Port ID (RPID, ingår ej i en BPDU utan räknas ut lokalt)

När det först förekommande lägsta värdet hittas så avslutas processen och den bästa Configuration BPDUn har hittats.

STP-processen

För att välja vilka portar som trafik ska forwardas på och vilka trafik ska blockas på så följer STP en trestegsprocess:

1. Välj root switch. Den switch med lägst bridge ID blir root.
2. Uträkning av root-port på övriga switchar. Det är portar på switchar som ej är root som leder till root. Den bästa BPDUn har tagits emot på denna port.
3. Välj Designated Port för varje nätsegment. Vid flera inkopplingar till samma nätsegment så är det denna port som BPDUerna skickas ut på.

Bridge ID

Formatet av Bridge ID har förändrats från den första versionen av 802.1D. Idag så definieras vanliga STP i 802.1D-2004.

image.png

Formatet ändrades för att få stöd för flera spanning-tree instanser per switch utan att behöva reservera en MAC-adress per VLAN, nu återanvänds samma MAC-adress för att bygga Bridge IDn. Extension konfigureras med globala kommandot `spanning-tree extend system-id`. Äldre switchar tillåter att man tar bort kommandot och går tillbaka till gamla formatteringen. Nya switchar tillåter inte att kommandot tas bort från konfigurationen.

Prioritet konfigureras per VLAN med det globala konfigurationskommandot `spanning-tree vlan 1337 priority 4096`. Det går att specifisera flera VLAN i samma rad, ex. `spanning-tree vlan 50-51,1337 priority 4096`. Prioritetsvärdet sätts i steg på 4096. Lägst prioritet är bäst. Default är 32768.

```
catalyst9300-lab(config)#spanning-tree vlan 1337 priority 500
% Bridge Priority must be in increments of 4096.
% Allowed values are:
  0     4096  8192  12288 16384 20480 24576 28672
  32768 36864 40960 45056 49152 53248 57344 61440
```

Best practice är att inte använda sig av prioritet 0 utan något högre än så, ex 8192. Väljer man 8192 kan man alltid sätta in en till brygga med lägre prioritet.

Val av root switch

Enbart en switch kan vara root i en STP domän. När en switch startar och initierar STP så antar den att den själv är root, den skapar Configuration BPDUer därefter och skickar ut dessa. Om switchen tar emot en bättre Configuration BPDU så slutar den anse sig själv som root och slutar skicka Configuration BPDUs, istället så forwardar den de bättre BPDUerna från root-switchen.

Uträkning av root-port

När root-switchen är vald måste samtliga andra switchar räkna ut sin väg till root-switchen, alltså bestämma vilken port som är root-port. Processen är enligt följande:

1. Root-switchen skickar ut Configuration BPDUs varannan sekund per default (Hello Time-värdet). BPDUn innehåller root-switchen ID i RBID och SBID. RPC är satt till 0 och SPID är egress-interfacet.
2. Varje icke-root switch som tar emot en configuration BPDU på en viss port lägger till portens **cost** till RPC-värdet i den mottagna BPDUn, vilken resulterar i en **resulting BPDU** . Den port som har tagit emot den bästa Configuration BPDUn blir Root Port.
3. Configuration BPDUs som mottages på Root portar skickas vidare på samtliga designated portar efter uppdatering av RPC, SBID, SPID och MessageAge värdena. Configuration BPDUs som mottages på andra portar analyseras men skickas ej vidare.
4. Configuration BPDUs skickas ej vidare ut på Root portar eller ut på portar som stabiliseras till Blocking state. Configuration BPDUs som eventuellt skulle skickas ut på dessa interface skulle vara ointressanta för mottagaren.

Uträkning av Designated-port

Ut på ett Ethernet-segment ska det bara finnas en Designated-port. I andra änden av en Designated-port ska läget vara Root-port. Enbart Designated-ports skickar vidare Configuration BPDUs. Portar som inte är Root-port eller Designated-port flyttas till Blocking.

Costs

Ett interface tilldelas en spanning-tree kostnad som används för uträkningen av RPC (Root Path Cost). De äldre standardena av spanning-tree ser inte skillnad på cost på ett 1 gbit/s interface mot ett 10 gbit/s interface, det har uppdaterats i senare versioner. För att använda sig av 802.1D-2004 costs globlat i switchen kan man aktivera det med kommandot `spanning-tree pathcost method long`. Per default är `spanning-tree pathcost method short` aktiverat.

Länkhastighet	**Pre-802.1D-1998 Cost**	**802.1D-1998 Cost**	**802.1D-2004 Cost**
10 Mbit/s	100	100	2000000
100 Mbit/s	10	19	200000
1 Gbit/s	1	4	20000
10 Gbit/s	1	2	2000
100 Gbit/s	x	x	200
1 Tbit/s	x	x	20

Det går även att tilldela ett interface en cost manuellt med kommandot `spanning-tree cost x` i interface-konfigläge. Det går även att sätta per VLAN med `spanning-tree vlan 10 cost x`.

Konvergering

När en STP domän har stabiliserat sig efter uppdatering så har den konvergeras. Beräkningar sker dock varje gång en Configuration BPDU mottages, vilket alltså sker per default varannan sekund då en ny Configuration BPDU genereras av root-switchen enligt Hello Time-värdet.

Topologin kan, och kommer, förstår att förändras. I STP-termer är det en **topology change** som inträffar när:
- En Topology Change Notification BPDU tas emot på en Designated port i en switch
- En port förflyttas till Forwarding state och switchen har minst en Designated Port
- En port förflyttas fråN Learning eller Forwarding till Blocking
- En switch blir root

Switchar som märker en förändring i topologin börjar skapa BPDUer med uppdaterad information. Grannar som tar emot dessa behandlar de som vanligt, processar och ev. skickar ut på designated ports.

När STP konvergeras enligt den nya topologin så kan det finnas poster i Content Address Memory (CAM, mac address table) som inte längre stämmer. STP kan då berätta för switchen att oanvända poster ska rensas i förtid. Det måste ske på samtliga switchar och en kort timer måste användas som är likk Forward Delay timer, default 15 sekunder. En Topolocy Change Notification (TCN) BPDU skickas från switchen som märkte topologiförändringen till root. TCN BPDU tas emot på ev. designated ports och vidarebefodras ut på root-portar. Acknowledgements skickas tillbaka för varje hopp genom att märka Topology Change Acknowledgment (TCA) bit i Configuration BPDUer. Root-bryggan skapar vid mottagande av TCN BPDU en Configuration BPDU med Topology Change (CN) bit satt till 1 vilket instruerar alla switchar att minska aging time av CAM poster till sekunderna i ForwardDelay timer.

Interface-lägen

I STP finns flera interface states. Några av dem har redan nämnts här innan. Det finns fler states i klassisk STP än i Rapid STP.

Vid konvergering så kan det finnas Designated Ports eller Root Ports som ska gå från Blocking till Forwarding. Men för att inte skapa några loopar så lyssnar de, Listening, först ett tag och lär sig, Learning, innan de blir forwarding. I klassisk spanning-tree så varar varje övergående tillstånd i 15 sekunder.

State	**Växlar data paket?**	**Lär sig source MAC-adresser från mottagna paket?**	**Övergående eller stabilt tillstånd?**
Blocking	Nej	Nej	Stabilt
Listening	Nej	Nej	Övergående
Learning	Nej	Ja	Övergående
Forwarding	Ja	Ja	Stabilt
Disabled	Nej	Nej	Stabilt

I klassisk 802.1D spanning-tree gäller interface tillståndet en hel länk.

Exempelkonfiguration 802.1D

```
spanning-tree vlan 1 priority 28672


interface FastEthernet0/1
 spanning-tree vlan 1 cost 100
```

Show kommandon 802.1D

Kommando	**Förklaring**
show spanning tree vlan 1337 [detail]	Se status för ett visst VLAN i en STP instans
show spanning tree root	Se vägar till rootbrygga
show spanning-tree [detail]	Information om samtliga STP instanser

STP Standarder

802.1D, Spanning-Tree Protocol

802.1D är den första standarden för L2-bridging vilket inkluderade den första versionen av STP. 802.1D STP är idag föråldrad och är ersatt med 802.1D-2004 STP. 802.1D och 802.1D-2004 har beskrivts ovan.

Per-Vlan Spanning Tree

PVST är en Cisco propertiär implementation av STP. Numera är PVST+ som används. Interface-lägen kan i PVST+ förhandlas per VLAN över en trunk. Det gör att det går att lastbalansera VLAN över olika länkar. VLAN 1 STP instans används för kommunikation med switchar som inte klarar av PVST+. Formatteringen av 802.1D och PVTST+ STP paket skiljer sig och därmed kan switchen avgöra om den ska prata PVST+ full ut eller använda sig av VLAN 1.

802.1w, Rapid STP

Det som en gång var 802.1w har nu införlivas i 802.1D, då den första standarden för STP är föråldrad och används inte längre. Målet med RSTP är att förbättra STPs konvergeringstid till under en sekund. Portar har i RSTP porttillstånd (state), portroller (roles) och porttyper (types). Antalet

porttillstånd har förändrats från 5 i 802.1D STP till 3 i RSTP

Porttillstånd

Administrativt tillstånd	**Porttillstånd STP 802.1D**	**Porttillstånd RSTP**
Disabled	Disabled	Discarding
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding

I RSTP innebär Discarding att ingen trafik switchas, tas emot eller att source MAC-adresser installeras i CAM. En port i Discarding processar dock fortfarande BPDUer, beroende på roll skickar ut BPDUer och tar emot inter-switch signalprotokoll såsom DTP, VTP, CDL, LLDP, PAgP, LACP eller LOOP. Discarding är standardläge för nya portar som tas upp med undantag för Edge-portar som direkt är i Forwarding.

I RSTP är porttillstånd och portroller två olika saker. De fyra portroller som finns är:

Root port	Samma som i 802.1D
Designated port	Samma som i 802.1D
Alternate port	En möjlig ersättare för switchens Root Port. Tar emot BPDUs som är sämre än de som kommer via Root-porten. Vid händelse att Root-porten går ned blir Alternate befodrad till Root direk
Backup port	En möjlig ersättare för switchens Designated Port. Är redo att bli Designated port men blir inte det direkt utan väntar på timers.

Porttyper och länktyper

Porttyper i 802.1w är antingen *Edge port* eller Non-*Edge port*. En edge port konfigureras med `spanning-tree portfast` i interface-konfigurationsläge. En Edge Port är direkt i Forwarding läge. BPDUs skickas ut på en Edge port men tas BPDUs emot så stängs den ned och går igenom RSTP-processen, Discarding, Learning och till slut åter i Forwarding. Portar är per default Non-Edge.

Best practice är att samtliga portar som inte ska ansluta till andra switchar konfigureras som Edge-portar. Det gör att användaren slipper vänta på RSTP-uträkning när de kopplar in något i nätverksuttaget och det undviker eventuell nedtid orsakad av Proposal/Agreement processen som beskrivs längre ned.

Det finns även två länktyper, *Point-to-point link* och *Shared link*. PTP-länkar är en länk till en annan switch och Shared är till ett segment med två eller flera switchar. I moderna nätverk är länkar PTP.

I äldre nätverk där interfacet går upp i halv-duplex så blir länktypen Shared. Det sker om en hub sitter inkopplad mellan switcharna. Länktypen kan konfigureras på interfacet med `spanning-tree link-type { point-to-point | shared }`.

BPDUer

I 802.1w finns bara en BPDU-typ. TCN används ej av RSTP. I STP så skickas Configuration BPDUs av rootswitchen. I RSTP så skickas Configuration BPDUs av samtliga RSTP-switchar, det går alltså att jämföra med Hello-paket i ett routingprotokoll. Om en switch slutar tar emot Configuration BPDUs på ett interface kan switchen med säkerhet anta att felet på är på den länken och snabbt förklara de emottagna sparade BPDUerna som irrelevanta och ta bort dem efter 3 gånger så lång tid som Configuration BPDU intervallen.

Hanteringen av sämre BPDUer har förbättras gentemot STP. I RSTP behöver switchen inte vänta på att de bättre BPDUerna time:ar ut enligt MessageAge-MaxAge värdena. Skulle det mottagna BPDU-värdet på ett interface förändras, ex. bli bättre eller sämre, så accepteras det direkt och uträkning för porttillstånd sker.

Proposal/Agreement Process

Förbättringar har även skett för tillägg av nya länkar som har en bättre kostnad till root. För att undvika loopar vid inkoppling av nya länkar finns Proposal/Agreement processen. För att undvika loopar måste portrollsförändringen ske på den switch som förändrar vilket interface som är mot root. Switchen kan placera alla Non-Edge Designated ports i porttillståndet Discarding innan nya Root-porten blir Forwarding. Det skulle undvika loopar men är trafikstörande. För att undvika att vänta på ForwardDelay-timern två gånger (uträkning för root på switch 1 och designated på switch 2) så skickas signaler mellan switcharna när det är säkert att placera Designated-ports i Forwarding med *Proposal/Agreement*.

När en ny PTP länk tas upp mellan switchar är interfacen per default i Designated Discarding. BPDUs skickas ut med Proposal-bit:en flaggad. Om en av portarna märker att BPDUn är bättre ska portrollen bli Root. En RSTP-switch som tar emot en bättre BPDU på root-interfacet sätter per automatik samtliga Non-Edge Designated ports i läget Discarding, proceduren kallas för *Sync*. En switch i sync är i princip isolerad från nätverket och orsakar inga loopar. Nu är det säkert att flytta den nya Root-porten från Discarding till Forwarding och signalera till överliggande switch att den även kan byta sitt tillstånd på Designated porten till Forwarding. Det genomförs genom att skicka en Configuration BPDU med Agreement-biten flaggad efter sync, vid mottagande av BPDUn så flyttar överliggande switch porttillståndet till Forwarding direkt.

Nu kommer Proposal/Agreement processen att upprepas på eventuella Non-Edge portar nedströms.

Värt att komma ihåg från detta är att nedtid i L2-nät kan bero på direkta länkfel, indirekta länkfel eller tillägg av nya bättre länkar som kräver omräkning för att undvika loopar.

Topologiförändringar

I RSTP räknas endast förändringen från Non-edge port från Non-forwarding till Forwarding som en topologiförändring. En non-edge port som har blivit forwarding skulle kunna ha en bättre väg till MAC-adresser än vad som tidigare fanns och CAM behöver uppdateras. Förlust av en Non-edge port från forwardingläget räknas ej som en topoligiförändring.

Som tidigare nämnt används ej TCN BPDUer i RSTP. Istället skickas en configuration BPDU med TC biten flaggad. När en topologiförändring märks så sker följande:

- tcWhile-timern sätts till samma värde som Hello timern plus en sekund på Root-porten och Designated-portar.
- Tömmer direkt CAM från MAC-adresser som switchen har lärt sig på dessa portar
- Skickar Configuration BPDUer ut på dessa portar enligt Hello-intervall tills att tcWhile timern går ut

Information om topologiförändringen propageras väldigt fort enligt ovan. Edge-portar skapar aldrig en topologiförändring.

Rapid Per-VLAN Spanning-Tree (RPVST+)

RPVST+ är en version av RSTP som utförs per VLAN basis. Konfigureras globalt med `spanning-tree mode rapid-pvst`. Edge-portar kan konfigureras per interface eller default globalt med `spanning-tree portfast default`.

802.1s, Multiple STP

MSTP är en IEEE industristandard implementation av STP. MSTP tillåter att man grupperar olika VLAN i olika instanser och utför STP-uträkningar inom den instansen, till skillnad från PVST där uträkning sker per VLAN. MSTP använder sig av timers från 802.1w RSTP och är därmed ett snabbt protokoll. MSTP är ett bra val om man har flera switchleverantörer i nätet eller får slut på antalet STP instanser som kan köras i ett nät.

MSTP principer

MSTP organiserar nätverket i en eller flera *regione*r. En region är en samling switchar som tillsammans använder MSTP på ett konsekvent sätt. De har samma antal MSTP instanser och kopplar samma VLAN till dessa instanser. En av de största fördelarna med MSTP kontra PVST är att enbart en BPDU behövs skickas för samtliga MSTP instanser. I PVST så skickas BPDUer per VLAN. MSTP tillåter som högst 65 instanser.

Instans 0 är en speciell instans. Den kallas för *Internal Spanning* Tree (IST). Den finns per default även om ingen annan instans är skapad, samtliga VLAN mappas till IST om inget annat konfigurereas. IST är den enda instans som integererar med STP utanför MSTP-regionen.

Interoperabilitet mellan MSTP och andra STP-versioner

För att MSTP ska fungera med andra STP-versioner måste MSTP se ut för sin granne som om den enbart kör en STP-instans. MSTP gör detta genom att använda IST/instans 0 på gränsportar och genom att skicka STP/RSTP BPDUer ut på dessa gränsportar. IST pratar alltså på alla regioner vägnar mot STP/RSTP.

Mot PVST blir det lite mer komplext, men grundtanken är densamma som ovan. En enda representant från MST och PVST väljs, MSTP utför något som kallas PVST SImulation för konsekvent interoperabilitet med PVST. I riktning från MSTP till PVST så replicerar MSTP-switchen IST BPDUs till PVST BPDUs för alla aktiva VLAN. I riktning PVST till MSTP tar MSTP VLAN 1 som representant för hela PVST regionen och processar BPDU informationen i IST.

Konfiguration

```
spanning-tree mst configuration
 name MSTP
 revision 1
 instance 1 vlan 1-500
 instance 2 vlan 501-1000
 ! Går att kontrollera mapping i det här läget med show pending och show current
 exit

spanning-tree mst 0 priority 0
spanning tree mst 1 priority 40962
spanning-tree mst 2 priority 8192

! Går även att synka med VTPv3
vtp version 3
vtp mode server mst
do vtp primary mst
```

Skydda och optimera STP

PortFast

En Edge-port eller PortFast-port är en port som skippar spanning-tree uträkningen och hamnar i Forwarding direkt. Skulle BPDU tas emot på porten så stängs den. Konfigurera globalt med `spanning-tree portfast default` eller direkt på interface med `spanning-tree portfast`. Är default-läge satt stängs det av på port med `spanning-tree portfast disable`. Det går även att göra på trunkportar med `spanning-tree portfast trunk`, gör det aldrig på portar mot andra switchar. Vanligt användningsområde

för portfast trunk är mot ex. Cisco UCS eller lagringsmiljöer.

BPDU Guard

BPDU Guard är en säkerhetsfunktion som slås igång per port eller globalt per PortFast port som stänger porten när en BPDU mottages via errdisable. På interface nivå aktiveras det med `spanning-tree bpduguard enable` och globlat för samtliga portar aktiveras det med `spanning-tree bpduguard default`. Det globala kommandot aktiverar det på portar som verkar som PortFast portar. BPDU Guard kan avaktiveras perport med `spanning-tree bpduguard disable`. Automagisk errdisable recovery går att konfigurera med `errdisable recovery cause bpduguard`.

Root Guard

Root guard är en funtion som ignorerar bättre BPDUer som kommer in på en port så att den inte blir root-port. Aktiveras på portbasis. Vid mottagande av en för bra BPDU så placeras porten i Blocking tills att mottagande av bättre BPDUer på porten avstannar. Konfigureras med `spanning-tree guard root` i interface-konfigläge.

Verifiering:

```
c9500(config-if)#spanning-tree guard root
Aug 16 07:31:30: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port Port-channel31.


c9500#show spanning-tree interface port-channel 31 detail
 Port 2311 (Port-channel31) of VLAN0007 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.2311.
   Designated root has priority 7, address cc70.ed70.2c00
   Designated bridge has priority 7, address cc70.ed70.2c00
   Designated port id is 128.2311, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   Root guard is enabled on the port
   BPDU: sent 2579422, received 4
```

BPDU Filter

BPDU Filter är en funktion för att begränsa sändning och mottagande av BPDUer på en port. Det går att konfigureras globalt för Edge-portar med `spanning-tree porfast bpdufilter default`, dock skickas 11 BPDUs vid upplänk och den processar fortfarande mottagna BPDUer. Kan stängas av per Edge-

port med `spanning-tree bpdufilter disable`. Konfigureras `spanning-tree bpdufilter enable` i Interface-konfigläge så avstannar sändning och mottagande av BPDUer totalt.

UDLD

Unidirectional Link Detection (UDLD) är en Cisco-propertiär teknik som används för att upptäcka och eventuellt stänga länkar som har trafikfel i ena riktningen. UDLD verkar över L2. UDLD verkar i två lägen, normal och agressive. Om en switch slutar ta emot UDLD-paket i normal mode så försöker den att återansluta mot sin granne 8 gånger men gör ingenting med interfacet efter försöken. I aggressive mode så försöker switchen återansluta mot sin granne 8 gånger och tar ned interface i err-disable efter återanslutningsförsöken. UDLD kan konfigureras globalt med `udld { enable | aggressive }`, där enable är normal mode, eller per interface med `udld port [ agressive ]`. Verifiering sker med `show udld` och `show udld neighbors`. En port som har blivit err-disabled kan tas up med shut/no shut eller genom `udld reset`.

Loop Guard

STP Loop Guard är en funktion som skyddar mot loopar vid trafik i ena riktningen. Loop Guard ser till att Root och Alternate portar ej kan bli Designated om de slutar ta emot BPDUer. Porten placeras då i loop-inconsistent blocking state. De tas tillbaka till ordinarie läge när BPDUs börjar tas emot igen. Loop Guard konfigurerat globalt med `spanning-tree loopguard` default och per interface med `spanning-tree guard loop`.

Bridge Assurance

Bridge Assurance är en teknik för RPVST+ och MSTP som används på P2P-länkar. Bridge Assurance är en förlängning av idén med loop guard. Bridge Assurance ändrar reglerna för sändandet av BPDUer, när Bridge Assurance är aktiverat på en port så skickas alltid BPDUer vid varje Hello-intervall oavsett vilken STP roll switchen har. En Bridge Assurance-skyddad port måste alltid ta emot BPDUer, gör den inte det så placeras porten i BA-inconsisten blocking state tills att den tar emot BPDUer igen. Konfigureras globalt med `spanning-tree bridge assurance` och per interface med `spanning-tree protfast network`. Båda sidor av en länk måste vara konfigurerade för Bridge Assurance.

Backbone Fast

Backbone fast är en Cisco propertiär teknik som ska minska konvergeringstid vid återhämtning från indirekta länkfel.

Virtual Switching System (VSS)

VSS är en teknik som tillåter två fysiska switchar att bli en enda logisk switch. Det gör bland annat att det bara finns en administrationspunkt och att länkaggregeringar kan byggas mellan VSS-systemet och andra switchar för att upprätta redundans (MEC, Multichassis EtherChannel). VSS-paret ses då som en spanning-tree brygga.

Active och Standby

I ett VSS-kluster har en switch rollen Active och den andra rollen Standby. Switchen som är Active kontrollerar VSS-klustret, L2 och L3 protokoll för switchmodulerna på båda switchar, management funktioner för VSS såsom module online insertion and removal (OIR) och konsollinterfacet. Standbyswitchen utför packet forwarding för sina lokala interface men skickar all kontrolltrafik till den aktiva switchen för hantering.

Virtual Switch Link (VSL)

image.png

För att switcharna ska kunna agera som en logisk switch måste kontrollinformation och datatrafik delas. VSL är en speciell länk som bär kontroll och datatrafik mellan switcharna i VSS-klustret. Den implementeras oftast som en länkaggregering och kan sålledes bestå av upp till 8 interface. Kontrolltrafik prioriteras över datatrafik över VSL för att säkerställa att kontroll eller management-trafik aldrig kastas. Datatrafik balanseras enligt konfigurerad lastbalanseringsalgoritm.

Protokollet som används över VSL-länkarna heter Link Management Protocol (LMP). Det sker periodvisa checkar var 500 millisekund per default, kallas även VSLP timers.

Role Resolution Protocol (RRP) används även över VSL länkarna för att noderna ska bestämma vilken roll som vardera burk ska axla, dvs active eller standby.

Går samtliga VSL-länkar ned så initieras stateful switchover (SSO). Den passiva switchen blir då active.

VSS Konfiguration

Först så måste en virtuell domän konfigureras i båda switcharna.

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# switch virtual domain 10
Domain ID 10 config will take effect only
after the exec command 'switch convert mode virtual' is issued
SW1(config-vs-domain)# switch 1
SW1(config-vs-domain)# exit
SW1(config)#

SW2# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# switch virtual domain 10
Domain ID 10 config will take effect only
after the exec command 'switch convert mode virtual' is issued
SW2(config-vs-domain)# switch 2
SW2(config-vs-domain)# exit
SW2(config)#
```

Andra värden som kan sättas i virtual domain konfigurationen är `switch 1 priority xxx` som enbart spelar roll om switcharna bootar samtidigt, default är 100 och högst vinner, `mac-address use-virtual`, vilket gör att båda switchar använder sig av samma MAC-adress, samt `dual-active detection pagp` i kombination med dual-active trust `channel-group port channel x` vilket gör att man kan använda sig av en portkanal för DAD. Då det går bra att använda två fristående interface för DAD så finns det ingen anledning att använda sig av PAGP. Det går även att använda sig av BFD för DAD-länkarna.

VSL konfiguration, observera att numreringen för port-kanalerna är unika och att switch virtual link är 1 på första switchen och 2 på andra:

```
Switch 1:
interface Port-channel1
 description VSL
 no switchport
 no ip address
 no platform qos channel-consistency
 switch virtual link 1
```

```
Switch 2:
interface Port-channel2
 description VSL
 no switchport
 no ip address
 no platform qos channel-consistency
 switch virtual link 2
```

`no platform qos channel-consistency` gör att qos channel-consistency ej kontrolleras för att länkaggregering ska gå upp.

Lägg sedan till fysiska interface i port-kanalerna med mode on:

```
SW1(config)# int range gig7/3 - 4
SW1(config-if-range)# switchport mode trunk
SW1(config-if-range)# channel-group 1 mode on
WARNING: Interface GigabitEthernet7/3 placed in restricted config mode. All
extraneous configs removed!
WARNING: Interface GigabitEthernet7/4 placed in restricted config mode. All
extraneous configs removed!
SW1(config-if-range)# exit

SW2(config)# int range gig4/45 - 46
SW2(config-if-range)# switchport mode trunk
SW2(config-if-range)# channel-group 2 mode on
WARNING: Interface GigabitEthernet4/45 placed in restricted config mode. All
extraneous configs removed!
WARNING: Interface GigabitEthernet4/46 placed in restricted config mode. All
extraneous configs removed!
SW2(config-if-range)# exit
```

Interfacen kommer att vara "notconnect", status up men line protocol down, tills att switcharna har bootas. Konvertera nu switcharna till VSS genom kommandot `switch convert mode virtual`, switcharna startar vid konvertering om.

Efter att båda switchar har startat om verifiera VSS med `show switch virtual`. Konsollen på standby-switchen är nu avstängd.

Redundanstekniker, SSO och RPR

I global config och sedan redundancy konfigureras stateful switchover (SSO). Med SSO så är standby switchen alltid redo att ta över om ett fel sker på supervisorn på active switch. Med SSO så synkroniseras konfiguration, forwarding och lägesinformation. Synkronisering sker vid start och vid förändring.

Följande krav finns för att stateful switchover ska användas:
- Båda supervisors använder sig av samma mjukvaruversion
- VSL relaterad konfiguration matchar i båda chassin.
- PFC läget matchar.
- SSO och nonstop forwarding (NSF) är konfigurerat på båda chassin.

Om kraven inte möts för SSO så används istället route processor redundancy (RPR). Standby supervisor är bara delvis synkroniserad och switchingmoduler på standby supervisor är inte aktiva. Vid switchover startas switchingmodulerna vilket tar mellan 2 och 5 minuter.

```
redundancy
 main-cpu
  auto-sync running-config
 mode sso
```

Vid uppstart så synkas Startup-configuration, även om de är i RPR-läge.

Dual-Active Detection (DAD)

DAD är en teknik som används på en extra länk mellan switcharna i VSS-paret för att säkerställa att båda inte blir aktiva switchar i händelse av att VSL-länken går ned, ett så kallat split brain scenario. Det är inte ett krav att använda sig av DAD men det är starkt rekommenderat.

```
switch virtual domain 10
 dual-active detection fast-hello


interface te1/1/1
 description Dual-Active Detection VSS Fast-Hello
 no switchport
 no ip address
 no cdp enable
 dual-active fast-hello
 no shutdown
```

Verifiering sker med kommandot `show switch virtual dual-active fast-hello`.

Osynk, trasigt kluster

Skulle ett kluster vara trasigt, ex. om ena skulle gå sönder, ska den trasiga sidan konfigureras med VSS-konfiguration. Resterande konfiguration kommer att synkas över när klustret återställs.

Patchning

Ett VSS-kluster kan i många fall patchas utan total nedtid för produktion. Se Patchning Catalyst 6807-XL för mer information.

Omstart efter omgenerering av RSA nyckelpar

Vid försök att byta SSH-nyckel 2025-11-13 med kommandot crypto key generate ec keysize 256 så gick först DAD-länkarna ned följt av VSL-länkarna. SSH 2.0 stängdes av. Den aktiva sidan av klustret bootade om. Under ombooten så slog jag kommandot crypto key generate rsa modulus 4096 för att återskapa en RSA-nyckel istället.

Kommandon

Kommando	**Förklaring**
show redundancy	Kontrollera status i klustret
show switch virtual	Se VSS status
show switch virtual role	Se switchrollerna i VSS paret
show switch virtual link	Se status på virtual switch link (VSL), fysiska interface
show switch virtual link port-channel	Se status på virtual switch link (VSL), länkaggregering
show switch virtual dual-active fast-hello	Se status på DAD
redundancy reload peer	Starta om standby switch

redundancy force-switchover	Starta om active switch, standby blir nu active

Resurser

https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/15-4SY/config_guide/sup6T/15_3_sy_swcg_6T/virtual_switching_systems.pdf

VLAN

Ett virtuellt LAN, VLAN, är en teknik som delar upp switchportar i olika broadcast domains. När en switch tar emot en L2-frame med en destinations MAC-adress som den inte känner till så skickas L2 frame:n ut på samtliga fysiska interface inom det VLANet. Om switchen får svar från destinations MAC-adressen så bygger den en post i sitt MAC address-table med MAC-adressen och vilket interface den finns via. Detta sker inom ett VLAN, så om en enhet i VLAN 1 skickar en L2 frame med destinations MAC-adress till en enhet som finns i VLAN 2 kommer kommunikationen inte att fungera på egen hand.

VLAN knyts oftast ihop med ett IP-nät och kommunikation mellan VLAN:en sker genom en router eller L3-switch. Rent tekniskt går det att placera flera IP-nät i samma VLAN. Best practice är ett 1-till-1 förhållande mellan IP-nät och VLAN.

VLAN

Konfiguration

Skapa VLAN

Ett VLAN skapas i konfigurationsläge. Ett VLAN till delas ett nummer mellan 1 och 1001 och mellan 1006 och 4096. VLAN 1 är default VLAN och kan inte tas bort. VLAN 1002-1005 är reserverade. Det är rekommenderat att tilldela ett namn till VLAN:et för att det ska vara igenkännligt, men det är inte ett tekniskt krav. VLAN 1-1005 kallas för normal range och VLAN 1006-4096 för extended range, detta då äldre switchar ej hade stöd för extended range.

```
vlan 1000
 name NATIVE_VLAN
```

Verifiera att VLANet har skapats med `show vlan id 1000`. Det går även att se samtliga VLAN med `show vlan brief`.

Det går att avaktivera ett VLAN utan att ta bort det genom att sätta det i läget *suspend*. Accessportar som är tilldelade ett VLAN som är suspeneded fungerar ej, trafik som kommer in på de portarna droppas.

```
vlan 1999
 state suspend
```

VLAN sparas i running-configuration om man skapar VLAN manuellt och ej använder sig av Vlan Trunking Protocol.

Tilldela VLAN till accessport

En accessport, en port som ansluta sig direkt mot en enhet som ej VLAN taggar själv, kan ha 1 data-VLAN och 1 voice-VLAN. När trafik kommer in på porten så taggar switchen paketen med VLAN numret i L2 frame headern. När trafik skickas ut på porten så tas VLAN numret bort från L2 frame header. Skulle man tilldela ett VLAN som inte finns till en accessport så skapas det VLANet då.

```
interface GigabitEthernet1/0/1
 switchport access vlan 1337
 switchport mode access
```

VLAN Trunking

VLAN Trunking är när man tillåter flera VLAN att kommunicera över samma port. Trunk-portar används när man ansluter L2 mellan switchar eller upp till routrar. Trunking är en Cisco-term, i övriga världen kallar man dessa för taggade portar.

Historiskt sett har det funnits två stycken protokoll för trunking, ISL och 802.1Q. ISL (Inter-Switch Link) är Cisco propertiärt, förstår ej native VLAN och används inte i praktiken längre. Den öppna standarden som är definierad av IEEE, 802.1q, används av samtliga leverantörer. Benämns ofta enbart som en dot1q trunk.

802.1q har något som heter native VLAN. Trafik som inte har blivit taggade tidigare i nätet hamnar på native VLAN. Per default är native VLAN 1, best practice är att byta native VLAN till ett VLAN som inte används till någonting annat, detta för att skydda mot VLAN hopping attacker. Matchar inte native VLAN kan dessa kollidera, trafik som skickas ut på native VLAN 777 tas emot på native VLAN 666.

```
interface Port-channel1
 switchport trunk native vlan 666
 switchport trunk allowed vlan 800
 switchport mode trunk
```

När man lägger till VLAN på Cisco-switchar gäller det att hålla tungan rätt i mun. Nyckelordet add är viktigt vid tillägg för att undvika onödiga driftstopp. Använd gärna Configuration Rollback för att undvika längre driftstopp vid misstag. Det går att redigera flera VLAN åt gången genom att använda bindestreck för en serie eller VLAN separerade med kommatcken, ex. `10-20,30`.

switchport trunk allowed vlan 10	Gör att enbart vlan 10 tillåts på trunken och tar bort övriga redan tillåtna VLAN
switchport trunk allowed vlan all	Tillåt alla VLAN över en trunk
switchport trunk allowed except 10-20	Tillåt alla VLAN mellan 1 och 4096, förutom 10 till 20
switchport trunk allowed vlan none	Tillåt inga VLAN över trunken
switchport trunk allowed vlan remove 10,20	Ta bort VLAN 10 och 20 från trunken men tillåt resterande som redan är tillåtna

Stänga av Native VLAN beteendet

Det går att stänga av default native-VLAN beteendet, dvs att det som skickas ut på native VLANet skickas otaggat. Det gör man genom det globala kommandot `vlan dot1q tag native`, då skickas all trafik ut på trunkar VLAN-taggat.

Konfigurera trunkinterface på en router

```
interface Port-channel1.50
 encapsulation dot1Q 50
```

Subinterface numret (po1.50) måste inte matcha VLAN ID:t, men är man inte galen så matchar man.

Det är rekommenderat att skapa ett subinterface för native VLAN. Om det inte görs så skickas otaggad trafik ut via fysiska interfacets konfiguration.

```
interface Port-channel1.1337
 encapsulation dot1Q 1337 native
```

Konfiguration i äldre enheter, VLAN Database Mode

I äldre Catalyst-switchar och routrar så skapas VLANen i VLAN database configuration mode.

```
Switch3# vlan database
Switch3(vlan)# vlan 21
Switch3(vlan)# show current
Switch3(vlan)# show proposed
Switch3(vlan)# show current 22
```

```
Switch3(vlan)# vlan 22 name VLAN-NAME
Switch3(vlan)# show proposed 22
Switch3(vlan)# vlan 21 state suspend
Switch3(vlan)# vlan 21 state active
Switch3(vlan)# apply
```

Show kommandon, VLAN database mode

Kommando	Förklaring
show vlan brief	Se VLAN databas
show current	Se nuvarande VLAN databas efter att man har öpnat VLAN databasen
show proposed	Se föreslagen VLAN databas efter att man har öpnat VLAN databasen och utfört förändring

Private VLAN

Om man önskar att begränsa kommunikation inom ett VLAN kan man använda sig av private VLAN. Ett enda IP-nät kan användas inom ett private VLAN men inga av portarna tilldelade VLANet kan kommunicera med varandra. Private VLAN delar upp VLANs i secondary VLANs (sub-VLANs) och primary VLANs. Sett från utsidan så används primary VLAN precis som ett vanligt VLAN, men internt så används secondary VLANs med andra VLAN IDn. Ett primary VLAN kan alltså beskriveas som ett kluster som innehåller fler secondary VLANs.

image.png

Secondary VLANs kan agera på två sätt:
- som Community VLANs
- som Isolated VLANs

Portar som är tilldelade ett Community VLAN kan kommunicera med varandra direkt men kan inte kommunicera med andra VLAN. Beteendet här är alltså likt det vanliga VLAN beteendet.
Portar som är tilldelade ett Isolated VLAN kan inte kommunicera med andra portar inom Isolated VLANet eller med andra VLAN.

Ett enda primary VLAN kan associeras med flera secondary VLANs. Ett secondary VLAN kan enbart associeras med ett primary VLAN.

Portar som leder till något utanför private VLAN domänen, ex. upplänk till en router, konfigurereas som *promiscuous ports*. Promiscuous porten är inte associeras med något secondary VLAN utan

direkt med primary VLANet. Det som är ansluted mot promiscuous porten kan kommunicera med alla sekondära VLAN och alla sekondära VLAN kan kommunicera med det som är kopplat till promiscuous porten. Promiscuous portar beter sig som accessportar, de använder inte VLAN tagging såsom en dot1q trunk.

Trafik från primära eller sekondära VLAN kan **alltid** skickas ut över en dot1q trunk. Om paketet kom från en community eller isolated port och skickas över dot1q trunk så taggas det med sekundära VLANet. Om det paketet kom från en promiscuous port och skickas över dot1q trunk så taggas det med primära VLANet. När paketet har traverserat trunken så behandlas det på korrekt sätt, såsom om ex. isolated eller promiscuous port var lokalt ansluten.

Det finns två speciella trunktyper tillhörande private VLAN:
- Promiscuous PVLAN Trunk
- Isolated PVLAN Trunk

image.png

När ett paket från ett sekondärt VLAN ska skickas ut över en Promiscuous PVLAN Trunk så skrivs VLAN taggen om till det primära VLAN ID:t.

När ett paket från ett sekondärt VLAN ska skickas ut över en Isolated PVLAN Trunk så skrivs VLAN ID:t om från det primära VLAN ID:t till det som används för sekondärt isolated VLAN. En Isolated PVLAN Trunk används när paket skyfflas från en switch som förstår private VLAN till en switch som inte förstår private VLAN, men som kan hantera *protected ports,* även känt som Private VLAN Edge. Här konfigureras accessportar med `switchport protected` . Protected ports kan inte prata med andra protected ports men kan prata fritt med övriga portar, det går alltså att skapa VLAN-isolering inom samma switch med hjälp av protected ports.

Konfiguration

```
# Skapa private VLAN
vlan 199
 name Isolated
 private-vlan isolated
vlan 101
 name Community
 private-vlan community
vlan 100
 name Primary
 private-vlan primary
 private-vlan association 101,199
```

```
# Konfigurera interface för vlan 101
interface fa0/1
 switchport mode private-vlan host
 switchport private-vlan host-association 100 101

# Konfigurera promiscuous port
interface fa0/2
 switchport mode private-vlan promiscuous
 switchport private-vlan mapping 100 101,199

# Konfiguera SVI, om L3 routing används i samma switch
interface Vlan 100
 private-vlan mapping 101,199
```

Dynamic Trunking Protocol

DTP är ett Cisco propertiärt protokoll för att dynamiskt konfigurera trunklänkar. När det är påslaget finns det i två lägen:
- dynamic auto, läget förhandlas automatiskt men föredrar att vara en accessport
- dynamic desirable, läget förhandlas automatiskt men föredrar att vara en trunkport

För att en trunk ska förhandlas måste ena sidan vara konfigurerad som dynamic desirable eller hårdkonfad som trunk och fortfarande skickar DTP-meddelanden. DTP-meddelanden innehåller även VTP (Vlan Trunking Protocol) domänen.

DTP konfigureras genom switchport mode dynamic auto eller switchport mode dynamic desirable. Det ersätter alltså hårt konfigurerad access eller trunk. En hårt konfigurerad accessport eller trunkport skickar fortfarande DTP meddelanden om det inte stängs av.

DTP kan stängas av på interface-nivå med kommandot `switchport nonegotiate`.

DTP-status kan kontrolleras med `show dtp` och `show dtp interface`.

Q-in-Q

Q-in-Q tunneling är en teknik för att sträcka VLAN över ett annat VLAN. Det kan exempelvis användas för att ett företag ska kunna VLAN-taggga över en service providers nät. När trafik kommer in på en specifik port så läggs en extra VLAN-tag på i Ethernet-headern. Förutom på portarna som är anslutna mot Q-in-Q endpoints så switchas sedan paketen precis som vilket paket

som helst i nätet. Se till att ej använda samma VLAN som native VLAN någonstans, i värsta fall kan det skickas ut till någon annan kund eller nullas om inte native VLAN beteendet är avstängt med `vlan dot1q tag native`.

Konfiguration

```
vlan 100
 name Q-in-Q_Kund1

interface GigabitEthernet1/0/1
 switchport mode dot1q-tunnel
 switchport access vlan 100
 l2protocol-tunnel cdp
 l2protocol-tunnel stp
 l2protocol-tunnel vtp
 l2protocol-tunnel lldp
```

VLAN Trunking Protocol

VTP är ett protokoll som ska tillåta enkelt skapande av VLAN. Konceptet är att när ett VLAN skapas på en master-switch så skapas det på samtliga andra switchar i VTP-domänen istället för att man ska behöva skapa dem manuellt. VTP annonserar VLAN ID, VLAN namn, VLAN typ och status för VLANet. VTP finns i tre versioner, VTPv1, VTPv2 och VTPv3.

VTPv1 har enbart stöd för normal range VLANs.

Förbättringar i VTPv2 innehåller stöd för:
- Token Ring Concentrator Relay Function och Bridge Relay Function (TrCRF och TrBRF) typ VLANs, dessa används ej i Ethernet.
- Stöd för okända Type-Length-Value (TLV) records.
- Optimerad VLAN databas konsistens kontroll, i VTPv1 sker kontrollen vid modifiering av VLAN databasen, i VTPv2 så antar man att den redan har skett i switchen där VLANet lades till.

För att stänga av VTP använder sig många av `vtp mode transparent`. Det stänger dock inte av VTP, switchen bryr sig inte om VTP-meddelanden lokalt men den skickar fortfarande vidare VTP paket så länge domänen matchar den som är konfigurerad på den lokala switchen. En switch utan konfigurerad VTP domän skickar vidare samtliga VTP-meddelanden.

VLANen sparas ej i running-configuration utan i filen `vlan.dat` när man använder sig av VTP..

Skräckprotokollet VTP

I VTPv1 och VTPv2 är det den switch som har högst revisionsnummer den som bestämmer. Så om en switch med ett högt revisionsnummer men med helt annan VLAN-konfiguration än nuvarande kopplas in i ett nät kan hela L2-domänen gå sönder då VTP modifierar nätet enligt den nyinkopplade switchens konfiguration.

Risken att hela vlan-databasen tas bort på det här viset har löst sig i VTPv3.

VTPv3

VTPv3 är den senaste versionen och den mest rekommenderande att använda. VTPv3 har löst problem so

Förändringar i VTPv3:
- Serverroll, det finns två servertyper i VTPv3, primär och sekundär. Enbart den primära servern får modifiera VLAN-databasen. Den sekundära kan ta över rollen som primär vid behov. Det gör att VLAN-databasen ej kan bli rensad som i tidigare VTP-versioner.
- Lösenord, VTP lösenordet kan krypteras så att det inte syns i klartext.
- VTPv3 klarar av hela VLAN-rangen inklusive private VLANs. Pruning funkar dock enbart för normal-range VLANs.
- Off-läge, det går att ställa in så att switchen ej deltar i VTPv3 och kastar samtliga VTP-paket. Det går även att stänga av VTP per trunk.
- VTPv3 är en teknik för att föra över databas-information, ej enbart VLAN-databasen. Exempelvis MST-regioner kan synkar över VTPv3.

Skulle en VTPv3 switch konfigureras som transparent och sedan bytas tillbaka till klient så nollställs ej revisionsnumret såsom det gör i tidigare versioner. Det enda sättet att nollställa revisionsnumret är att konfigurera VTP domännamn eller konfigurera ett VTP lösenord.

En VTPv3 switch som ansluter sig mot en VTPv2 switch går över till VTPv2 på just det interfacet. Mot VTPv1 kan VTPv3 ej göra detta.

Conflict

Om switchar i en VTPv3 domän har olika uppfattning om vilken switch som är primärserver så kallas det för en *conflict*. Switchar i konflikt synkroniserar ej sina VLAN databaser även om alla VTP parametrar matchar. Kontrollera konflikter med kommandot `show vtp devices conflicts`.

VTP-lägen och funktoner

Det finns fyra typer av VTP-lägen en switch kan fungera som:

image.png

Det finns fyra typer av meddlanden som VTPv1 och VTPv2 använder sig av:

Läge	Förklaring
Summary Advertisement	Skickas av VTP server och klientswitchar var femte minut samt vid modifiering av VLAN-databasen. Meddelandet innehåller VTP domännamn, revisionsnummer, identiteten av den som uppdaterade senast, tidsstämpel från senaste uppdatering, MD5 summa över VLAN databasen och VTP lösenord samt antalet Subset Advertisements som följer denna Summary Advertisement.
Subset Advertisement	Skickas av VTP server och klientswitchar efter modifiering av VLAN-databasen. Innehåller hela VLAN databasen. Flera Subset Advertisements kan behövas om VLAN databasen är stor.
Advertisement Request	Skickas av VTP server och klentswitchar för att be sina grannar skicka hela VLAN databasen eller delar av den. Skickas när en VTP klientswitch startas om, när en switch blir klient eller när en server eller klientswitch får en Summary Advertisement med ett revisionsnummer höger än sitt egna.
Join	Skickas av VTP server och klientswitchar periodvis var femte minut om VTP Pruning är aktivt. Join-meddelanden innehåller ett bit-fält för varje VLAN i normal range som indikerar om det är aktivt eller inaktivt - och därmed ska "prune:as".

VTP-meddelanden skickas enbart ut över trunklänkar och accepteras enbart att tas emot över trunklänkar.

VTP processer och revisionsnummer

Varje en gång VLAN-databasen uppdaterade i VTPv1 och VTPv2 så räknar revisionsnumret upp. Som tidigare nämnt fungerade VTPv1 och VTPv2 som så att när de tar emot ett meddelande med högre revisionsnummer än sitt egna så litar de på den uppdatering och uppdaterar VLAN-databasen därefter.

Per default så är Cisco-switchar i VTP server mode men de skickar inga VTP uppdateringar förrän ett domännamn är satt. VTP klienter behöver dock faktiskt inte sätta ett domännamn utan antar att domännamnet som är med i den första VTP-uppdateringen de mottager är det som gäller. De måste dock konfigureras som klienter först.

För att skydda ett nät från otillåtna switchar kan man använda sig av VTP passwords. VTP Summary Advertisements skickar med en MD5 hash som är gjord från VLAN databasen och VTP lösenordet, om det är konfigurerat. Mottagande switch räknar ut sin egen MD5 hash efter mottagen uppdatering och jämför den med hashen från skickande switch.

Konfiguration

VTPv2

```
vtp version 2
vtp domain VTP-NAMN
vtp password PASSWORD
vtp mode server/client/transparent
```

VTPv3

```
vtp version 3
vtp mode client client/off/transparent/off
vtp password PASSWORD hidden


do vtp primary
```

Spanning-Tree Protocol

STP är ett protokoll som används för att ej skapa L2-loopar. STP ska alltid användas i ett bryggat nät. Se mer information här.

Show kommandon

show vtp status	Se information om VTP version, domännamn, VLAN, senaste förändring osv
show vtp password	Se VTP lösenord
show vtp counters	Statistik
show vtp devices	Enbart VTPv3
show vtp interface	Se om VTP är igång på interface

Cisco Resilient Ethernet Protocol (REP)

REP är ett Ciscopropertiärt protokoll för att snabbt konvergera ett L2-nät i ringtopologi. Det är ett alternativ till STP. Konvergensen är väldigt snabb, mellan 50 och 300 millisekunder. En REP domän kallas för segment. Segmentet är en omkretsen början-till-slut mellan två edge-portar. Destinations MAC-adress `0180.c200.0000` används, vilket vanligtvis används för STP. Enbart en port i hela ringen kommer att blockera VLAN för att undvika loopar.

image.png

Bilden till vänster visar ett ring segment där båda edge-portar är på samma switch och bilden till höger visar edge-portar i ett "open segment" där edge-portarna är på olika switchar.

Ett VLAN dedikerat till REP-admin behövs. Sedan specifiseras `rep admin vlan x` i globbalt konfigläge. Verifiering kan ske med `show interface GigabitEthernet0/1 rep { detail }`.

För att REP ska aktiveras på ett interface måste det vara ett NNI (network-to-network interface) och det måste vara en trunkport. REP stöds på länkaggregeringar. Edgeportar ska konfigureras där en är primary och en är secondary.

```
! Edge-portar i ringsegment
interface Port-channel1
 port-type nni
 switchport mode trunk
 rep segment 1 edge primary

interface Port-channel2
 port-type nni
 switchport mode trunk
 rep segment 1 edge

! Portar på övriga switchar
interface Port-channel 1
 port-type nni
 switchport mode trunk
 rep segment 1

Warning: Enabling REP automatically disables STP on this port. It is
recommended to shutdown all interfaces which are not currently in use to
prevent potential bridging loops.
```

Verifieringar sker med `show rep topology segment 1`.

Age-timers, alltså hur länge en switch väntar innan REP-grannskapet anses dött, kan konfigureras med interface kommandot `rep lsl-age-timer x`, där x är millisekunder. Hello-värdet är 3 gånger så

litet som age-timern. Timers ska matcha mellan switchar, annars kan en ostabil miljö skapas.

Debugging kan utföras med `debug rep failure-recovery` och `debug rep prsm`.

Ett val utförs när ringen formas. Varje port har en 64-bitar stor prioritet. De första 4 bitarna innehåller en failed bit och konfigurerbara fält. Följande 12 bitar innehåller portidentifierare och de kvarstående 48 bitarna innehåller switchens MAC-adress. Alternate portval sker npå en blockad port när en "öppen" port måste väljas. Switchar utbyter blocked port advertisements (BPA) med varandra vilket innehåller prioriteter. Prioritet per interface går att se med `show rep topology segment 1 detail`. En port med bättre prioritet blir "open" och en port med sämre blir "alternate". BPAs skickas bara av en alternate port under valet då BPAerna innehåller VLANinformation för blockerade VLAN.

Det går ej att sätta en specifik portprioritet. Det går dock bra att konfigurera ett interface som preferred med `rep segment 1 edge preferred`. Ingen preemption sker, vill man byta roll måste länken flappa. När en länk går ned så togglas "failed" bit som nämndes ovan.

Port IDn nämndes ovan i BPA paketet. De går att se med `show interfaces rep detail`. Genom att använda oss av port IDn kan vi åstakomma VLAN load sharing. Det konfigureras på primary och edge-portarna. Man specifiserar en port i segmentet där vissa VLAN ska blockeras. Ett interface kan specifiseras genom att manuellt ange port IDt, använda neighbor offset number manuellt eller genom perferred-nyckelordet.

För att förändring ska ske när man har justerat för VLAN load balacing måste man manuellt skaka om ringen lite. Det gör man med exec-kommandot `rep preempt segment 1`.

```
! Ändra per port ID
interface Port-channel1
 rep block port id 0011001D4692F700 vlan 13-37

! Numreringsmetod
interface Port-channel1
 rep preempt delay 15
 ! Siffran nedanför är distans till primary edge eller distance tills econdary edge
 rep block port -2 vlan 13-37

! Preferred-metod
interface Port-channel1
 rep block port preferred vlan 13-37

! Verifiering med show rep topology segment 1
```

I den här andra designen, REP open segment, har den fjärde Cisco-switchen (3560) inte möjligheten att köra REP. Den kommer att köra RPVST+.

image.png

Länkarna mot 3560 kommer att konfigureras lite annorlunda, se exempel:

```
 ! Interface mot 3560 som är SEC Edge port
interface GigabitEthernet0/1
 port-type nni
 switchport mode trunk
 rep segment 1 edge no-neighbor preferred
 rep stcn stp

 ! Interface mot 3560 som ej är PRI/SEC Edge port
 interface GigabitEthernet0/1
  port-type nni
  switchport mode trunk
  spanning-tree guard loop
  rep stcn stp
```

För att REP ska informera STP om förändringar i nätet ska `rep stcn stp` läggas till på interfacekonfen.

Ethernet Ring Protection Switching (ERPS)

ERPS är en teknik som fungerar likt REP men som är industristandard. Även känd som ITU G.8032. Ett segment kallas för "ring" och en länk mellan två ring-noder kallas för "ring link". "Ring ports" är portar i ring links. Alternate-porten (blockerad) kallas för ring protection link (RPL). Noden som blockerar en port kallas för RPL owner node. Precis som i RPL kommer bara en port i hela ringen att blockera VLAN föratt undvika loopar.

VLAN Mapping

VLAN Mapping innebär att en switch på en trunkport tar emot paket på ett VLAN och skriver om L2 headern till ett annat VLAN.
Konfigurationen är enkel, ex: `switchport vlan mapping 13 37`. Det första VLANet är mottaget VLAN och det andra är det som det skall skrivas om till.

DDNS

Dynamisk DNS är en teknik som tillåter en host att uppdatera ett DNS-record. Det är ex. användbart om man har routrar som får dynamiskt tilldelade adresser.

Exempelkonfiguration

Domain name måste vara konfigurerat för att DDNS ska fungera. Webbsträngen nedanför HTTP uppdateras beroende på hur mottagande server önskar HTTP POST meddelandet.

```
ip domain name jehrlander.net

ip ddns update method DDNS
 interval maximum 0 2 0 0
 HTTP
  add http://HOSTNAMN:LÖSENORD@ddns-server.domain.com

ip http client source-interface Cellular0/1/0

interface Cellular0/1/0
 ip ddns update DDNS
```

Embedded Event Manager (EEM)

EEM är ett verktyg för att kunna skripta saker direkt i IOS. Skripten kan vara återkommande eller reagera på ex. ett syslogmeddelande.

Exempelkonfiguration

```
! Specifiering av epost
event manager environment email_server 1.1.1.1
event manager environment email_to mottagare@jehrlander.net
event manager environment email_from router@jehrlander.net

! Nedan reagerar på ett syslog-meddelande
event manager applet RELOAD_MEMFAIL
 event syslog pattern "SYS-2-MALLOCFAIL"
 action 2.0 reload
 action 3.0 cli command "y"
 action 4.0 syslog msg "Startar om pga minnesfel"

! Nedan sker varje dag klockan 02:30 enligt switchens konfigurerade tid
event manager applet RESTART_EVERY_NIGHT
 event timer cron cron-entry "0 02:30 * * *"
 action 2.0 reload
 action 3.0 cli command "y"
 action 4.0 syslog msg "Omstart varje natt"

! Skicka mail vid syslog
event manager applet EMAIL_SYSLOG_CERT_EXPIRY
 event syslog pattern "-CERT_EXPIRY_" maxrun 60
 action 0005 syslog msg "Certificate expiry warning, sending e-mail to mottagare@jehrlander.net"
 action 0010 info type routername
 action 0020 mail server "$email_server" to "$email_to" from "$email_from" subject "CERT_EXPIRY_WARNING -
```

$_info_routername" body "Cert expiring soon. Check with cmd: show crypto pki timers detail" source-interface Loopback1

! Event manager startar om radiomodul i IR1101 om IP SLA ICMP check har misslyckats i 30 minuter
ip sla 10
 icmp-echo 192.168.1.1 source-interface Loopback0
  frequency 10

event manager applet VPN-DOWN-RESET-CELLULAR-MODEM
 event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.10 get-type exact entry-op lt entry-val "2" poll-interval 10
 trigger occurs 179 period 1790
 action 10 syslog msg "Ping failed last 30min, restart Cellular modem"
 action 20 cli command "enable"
 action 30 cli command "configure terminal"
 action 40 cli command "service internal"
 action 50 cli command "end"
 action 60 cli command "test cellular 0/1/0 modem-reset"

MACSEC

MACSEC (802.11ae) är ett protokoll som innebär kryptering över L2 point-to-point, mellan switch och switch eller från switch till klient.

Komponenterna som ingår är:

- MACSEC Key chain (måste vara i HEX)
- MKA Policy (Macsec Key Algorithm)
- Konfigurering på interface

MACSEC stödjs ej direkt på länkaggregeringar. Interface som ska ingå i port-kanaler, men ändå kryptera trafik via MACSEC, måste först tas ur port-kanalen, konfigureras med MACSEC och sedan konfigureras tillbaka in i portkanalen.

MACSEC mellan switch och klient vid 802.1x innebär att injicering av trafik från tredje part på samma länk omöjliggörs.

För att undvika att länkar går ned oväntat kan man tvinga periodvis omförhandling av nycklar med `sak-rekey interval x`.

Exempelkonfiguration switch till switch

```
key chain MACSEC_KEY macsec
 key 20
   cryptographic-algorithm aes-256-cmac
  key-string 0 xxx....

mka policy MKA_POLICY
 key-server priority 100
 macsec-cipher-suite gcm-aes-256
 confidentiality-offset 30
 sak-rekey interval 300
 ssci-based-on-sci
```

```
interface GigabitEthernet1/0/48
 macsec replay-protection window-size 1000
 macsec network-link
 mka policy SMKA_POLICY
 mka pre-shared-key key-chain MACSEC_KEY
```

Kommandon

- show mka *
- show mka sessions interface te 1/0/34 detail
- show macsec *
- debug mka *
- debug macsec *

Se även Macsec-konfiguration för NXOS.

EtherChannels / Port-channels / Länkaggregering

Kärt barn har många namn. Etherchannel, eller port-channel, är Cisco IOS-XEs implementation av länkaggregering, alltså möjligheten att gruppera flera fysiska interface till ett logiskt interface. Genom att gruppera flera interface till ett logiskt interface tror spanning-tree att det enbart är ett interface och därmed kan man använda samtliga portar utan någon blockering.

Lastbalansering

Ett enda Ethernet-frame kommer enbart att skickas över en av länkarna i aggregeringen. En hashings-funktion baserat på Ethernet-framen väljer vilken av de fysiska länkarna Ethernet-framen ska skickas över. Funktionen skapar samma hash för samtliga Ethernet-frame:ar i ett flöde och därmed skickas samtliga Ethernet-frames ut på samma länk. Det gör att även om 2 x 1 Gbit/s bandbredd finns tillgängligt i en länkaggregering kan enbart 1 x 1 Gbit/s användas per flöde.

Lastbalanseringsmetoderna beror på switch, mjukvara och inställning. Lastbalanseringen baseras på innehållet i L2, L3 och/eller L4 headers. Metoden går att ställa in med kommandot `port-channel load-balance x`. Verifiering sker med `show etherchannel load-balance`.

En länkaggregering kan i Cisco IOS-XE bestå högst av 8 medlemmar. Vid 2, 4 och 8 medlemmar som lastbalanseras trafik jämt över länkarna. Vid övriga så lastbalanseras trafik ojämnt, se bild nedan. Det går alltså att konfigurera länkaggregeringar med ex. 3 interface men är generellt rekommenderat att hålla sig till 2, 4 eller 8 interface.

image.png

Spanning-Tree Protocol

Då hela portkanalen behandlas som ett interface i STP så skickas enbart en BPDU ut för samtliga medlemmar i aggregeringen. BDPU-framen skickas ut enligt samma hash som för övrig datatrafik över aggregeringen.

Cisco har implementerat en skydssfunktion som heter *STP EtherChannel Misconfig Guard*. Den är aktiverad per default. Tekniken innebär att när två switchar är ihopkopplade via en

länkaggregering så används enbart en MAC-adress per switch för kommunikationen mellan switcharna. Skulle en ny source MAC-adress skicka en BPDU så blir portkanalen err-disabled. Funktionen går att stänga av med `no spanning-tree etherchannel guard misconfig`.

Konfiguration

Interface i länkaggregeringar måste använda sig av samma konfiguration för följande värden:

- Samma hastighet och duplex
- Samma operativt läge, trunk, access eller dynamisk
- Om ej trunk ska det vara samma access-VLAN
- Om trunk ska det vara samma tillåtna VLAN samt native-VLAN
- Samma cost i spanning-tree
- SPAN ska ej vara konfigurerat

När man skapar en port-kanal så kommer det interfacet (interface Port-channel x) att ärva konfiguration från det första deltagande fysiska interfacet. Nästa interface som läggs till kommer att jämföras mot port-kanalens konfiguration och om det ej matchar så kommer det fysiska interfacet hamnar i läge *suspended* och därmed inte fungera tills att konfigurationen stämmer exakt. Konfigurationsförändring direkt på Port-channel interfacet skjuts ned till de fysiska medlemsportarna som inte är i suspended.

Följ riktlinjerna nedan vid konfiguration av port-kanaler:

- Skapa ej port-kanaler manuellt innan de fysiska portarna har knutits till interfacet
- Vid borttagande av port-kanaler ta bort interfacet så att det ej skapar problem vid skapande av en port-kanal med samma nummer senare
- Se till att konfigurationen på de fysiska interfacen är identiska innan de läggs till i samma port-kanal
- Om en fysisk ports konfiguration ej matchar port-kanalen så modifiera den fysiska porten först och utför sedan förändringar på port-channel interfacet

För att statiskt lägga till ett interface i en portkanal konfigurereas `channel-group x mode on`, där x är en siffra, i interfacekonfigläge. Helst ska en portkanal konfigureras med dynamisk förhandling och inte statiskt för att undvika att trafik droppas i andra änden eller för att orsaka eventuell problem med STP.

Två protokoll finns för att dynamiskt förhandla länkaggregingar:

- Port Aggregationa Protocol (PAgP), Cisco propertiärt
- Link Aggregation Control Protocol (LACP), industristandard

PAgP tillåter 8 länkar i en aggregeringen. Det går att konfigurera hur ofta kontrollmeddelanden skickas ut via `pagp timer { normal | fast }` i interface-konfigläge. Normal är var 30 sekund efter att

portkanalen är etablerad och fast är varje sekund.

LACP tillåter max 16 länkar i en portkanal varav max 8 är aktiva länkar i portkanalen. Resterande länkar placeras i Standby-läge, när en aktiv länk går ned tar en standby länk över. En av de två switchar som pratar LACP kommer att välja vilka standby länkar som blir aktiva. Det är switchen med lägst LACP system ID som gör detta. ID:t består av en prioritet och MAC-adress, precis som i STP. Prioriteten konfigureras globalt med `lacp system-priority 0-65535` och per interface med `lacp port-priority 0-65535`.

PAGP konfigureras med `channel-group X mode { auto | desirable }` varav ena sidan måste vara desirable för att förhandlingen ska börja.

LACP konfigureras med `channel-group x mode { passive | active }` varav ena sidan måste vara active för att förhandlingne ska börja.

Det går att ställa in i interface-konfigurationsläge att enbart LACP eller PAGP ska tillåtas med `channel-protocol { pagp | lacp }`.

```
c9300(config)#interface gigabitEthernet 1/0/16
c9300(config-if)#channel-protocol lacp
c9300(config-if)#channel-group 47 mode desirable
Command rejected: the interface can not be added to the channel group
c9300(config-if)#channel-group 47 mode active
Creating a port-channel interface Port-channel 47
```

ACL

Network Object Group

NOG är ett bra sätt att gruppera nät eller hostar.

IPv4-syntax:

```
object-group ip address IPv4-NOG
 host-info 1.3.3.7
 1.3.3.0 255.255.255.0

ip access-list extended IPv4-NOG_ALLOW
 permit ip addrgroup IPv4-NOG any
```

IPv6-syntax:

```
object-group network v6-network IPv6-NOG
 host 2001:abba::1337
 2001:abba::/64

ipv6 access-list IPv6-NOG_ALLOW
 permit ipv6 object-group IPv6-NOG any
```

Downloadable ACL

DACL är en teknik som tillåter ACL-applicering vid Radius/dot1x-autentisering. Tekniken lämpar sig väl för att implementera klientisolering inom samma subnät närmast användaren.

Själva accesslistan byggs i ISE under Policy > Policy Elements > Authorization > Downloadable ACLs. En ACL kan vara så lätt att den tillåter eller nekar all trafik. Observera att man använder en DACL per protokoll, i händelse av dual-stack behövs två DACL. ACLn byggs enligt IOS-standard, ex:

```
permit icmp any any
deny ip any 192.168.1.0 0.255.255.255
permit ip any any
```

För att DACLn ska användas måste den kopplas mot den Authorization Policy som används i Policy Set:et. Lägg till DACL i Common Tasks enligt skärmdumpen.

image.png

När den här sedan är applicerad i Policy Set:et kan man verifiera att den har blivit tillämpad i show access-session, se exempel:

```
c9300#show access-session interface gigabitEthernet 1/0/3 details
           Interface:  GigabitEthernet1/0/3
             IIF-ID:  0x19AE4F44
         MAC Address:  aaaa.bbbb.ccc
        IPv6 Address:  fe80::aaa:bbbb:cccc:ddd
        IPv4 Address:  192.168.1.1
           User-Name:  client
              Status:  Authorized
              Domain:  DATA
      Oper host mode:  single-host
    Oper control dir:  both
     Session timeout:  N/A
   Common Session ID:  140BA8C000001D44B98DD61C
     Acct Session ID:  Unknown
              Handle:  0x3c000d3d
      Current Policy:  POLICY_Gi1/0/3
```

```
Local Policies:
 Service Template: DEFAULT_LINKSEC_POLICY_SHOULD_SECURE (priority 150)
      Security Policy:  Should Secure
      Security Status:  Link Unsecured

Server Policies:
          Vlan Group:  Vlan: 1337
            ACS ACL: xACSACLx-IP-PERMIT_USER_IPv4-64f82bfd
        ACS ACL IPV6: xACSACLx-IPV6-PERMIT_USER_IPv6-664d9894



Method status list:
      Method          State
       dot1x          Authc Success
```

Det går även att titta på ACL:en "precis som vanligt", så länge någon port använder sig av den och den därmed är nedladdad.

```
c9300#show ip access-lists xACSACLx-IP-PERMIT_USER_IPv4-64f82bfd
Extended IP access list xACSACLx-IP-PERMIT_USER_IPv4-64f82bfd
    1 permit icmp any any
    2 deny ip any 192.168.1.0 0.255.255.255
    3 permit ip any any
```

Cisco Discovery Protocol och Link Layer Discovery Protocol

Två protokoll som utför (nästan) samma funktion. Ena är propertiärt och andra inte.

Det händer att sårbarheter hittas i CDP och LLDP, ex. har switchar kraschat vid mottagande av LLDP-frames konstruerade på ett visst sätt. Därför anser jag att det är bra att stänga av CDL och LLDP på portar där det inte behövs.

CDP och LLDP bör alltid vara avstängt på interface som går till nätverksutrustning utanför en egen organisations kontroll, såsom partners och internetleverantörer.

Cisco Discovery Protocol (CDP)

CDP är ett protokoll som tillåter Cisco-enheter att uppätcka andra Cisco-enheter. Som namnet antyder så används det av Cisco-enheter men andra leverantörer har implementerat stöd för att ta emot och förstå CDP-meddelanden, ex. äldre HPE Aruba switchar har den möjligheten.

Ett CDP paket skickas var 60 sekund som ett L2 multicast meddelande ut på varje interface där CDP är påslaget. Per default är CDP påslaget på alla interface. `show cdp` visar hur ofta CDP-paket skickas. Timers kan konfigureras globalt med `cdp timer x` och `cdp holdtime x.`

Följande information finns i ett CDP-paket:

- Hostnamn enligt konfigurerad hostname
- IOS mjukvaruversion
- Hårdvaruförmågor, såsom routing, switching och bridging
- Hårdvaruplattform, ex. 2960, 9300
- Enhetens L3-adresser
- Namn på interfacet som CDP skickades via
- Duplex-information från interfacet som CDP skickades via
- VTP domän, om relevant
- Native VLAN, om relevant

CDP kan stängas av globalt med `no cdp run`. Det kan stängas av per interface med `no cdp enable`. Det går att att stänga av globalt och sen aktivera på interface. `show cdp neighbors [ detail ]` visar information om grannar.

Link Layer Discovery Protocol (LLDP)

LLDP är inte propertiärt och har implementerats hos många olika tillverkade, exempelvis hos andra nätverksfabrikat men återfinnes i nästan allt som har ett nätverkskort, såsom skrivare, mötesplattor och datorer. LLDP är definierad i IEEE standarden IEEE 802.1AB.

LLDP skickas över L2. LLDP-frames innehåller Type Length Value fält (TLVs). Följande är tvingande fält i TLVerna:

- Portbeskrivning
- Systemnamn
- Systembeskrivning
- Systemförmågår
- Management IP-adress

LLDP aktiveras genom globala kommandot `lldp run` och sedan `lldp receive` och/eller `lldp transmit` i interface-konfigurationsläge. Timers kan konfigureras globalt med `lldp holdtime x`, `lldp reinit x` och `lldp timer x`.

Verifieringar kan ske med ex. `show lldp`, `show lldp interface` och `show lldp neighbors`.

Policy Based Routing (PBR)

PBR är ett verktyg för att göra routing-beslut baserad på en policy. Routern kommer då vid matchning ta beslut enligt en konfigurerad policy istället för via routing-tabell/FIB. Policyn konfigureras via en route-map som sedan appliceras på inkommande interface.

image.png

PBR är en bra teknik att använda i en VXLAN/EVPN-fabric (troligtivis i en Nexus-miljö, om Cisco) för att låta en brandvägg sköta policy-beslut inom en VRF mellan VLAN. PBR-konfiguration måste då tilldelas samtliga SVI:er inom en VRF. Det kan vara en bra idé att undanta tung trafik, ex. lagring, från PBRn för att inte överlasta brandväggen.

Konfiguration

Följande val finns i en route-map för beslut:

Kommando	Kommentar
set ip next-hop *ip-address* set ipv6 next-hop *ipv6-address*	Bestäm next hop. Måste vara direktanslutet.
set ip default next-hop *ip-address* set ipv6 default next-hop *ipv6-address*	Samma som ovan, men försöker först att routa enligt RIB, och om ingen träff finns så används PBR. Default-routen i RIB ignoreras.
set interface *interface*	Bestämmer next hop interface. Bör enbart användas för P2P interface.
set default interface *interface*	Samma som ovan men enligt samma logik som set ip default next-hop.
set ip df *0/1*	Sätter don't fragment biten till 1 eller 2. Stöds enbart för IPv4.
set ip precedence *number* | *name*	Sätter IP precedence bit. 0-7 eller text-namn.
set ipv6 precedence *number*	Sätt IPv6 precedence bit 0-7.
set ip tos *number* | *name*	Sätt ToS bit enligt decimalvärden eller ASCII namn. Stöds enbart för IPv4.

Exempelkonfiguration IPv4

```
route-map PBR-IPv4 permit 10
 match ip address prefix-list PBR-FILTER
 set ip next-hop

interface Vlan 10
 ip policy route-map
```

Debug

Debug kan man göra med `debug ip policy`. Exempel på debug-output:

```
.Aug 25 14:02:30: SW2: IP: s=172.13.37.35 (Vlan1337), d=172.16.1.17, len 52, FIB policy match
.Aug 25 14:02:30: SW2: IP: s=172.13.37.35 (Vlan1337), d=172.16.1.17, len 52, PBR Counted
.Aug 25 14:02:30: SW2: IP: s=172.13.37.35 (Vlan1337), d=172.16.1.17, g=192.168.18.18, len 52, FIB policy routed
```

Här syns då source IP, source VLAN, destinations-IP samt om PBR sker med ny destination `g=`.

Local PBR

Local PBR används för trafik som initieras från routern själv. Sätts med ett global kommando, `ip local policy route-map ROUTE_MAP_NAMN`. Verifiering med `show ip local policy`.

Problem, lärdomar

På en Catalyst C6807-XL plattform på mjukvara 15.5(1)SY12 så användes objekt-grupper för att identifiera de nät som ska PBRas. En ACL användes för att PBRa trafik som kommer från adresser i objektgruppen till adresser i objektgruppen.

Jobbet gick dåligt och konfigurationen behövde backas. Att uppdatera objektgruppen löste ingenting. Att ta bort ip policy från SVI eller att ta bort och lägga till route-map löste ingenting. Hela

ACLen behövdes tas bort och läggas till på nytt för att felet skulle lösas.

I samma plattform så såg det ut enligt följande:

Första raden i ACL var "any till specifikt nät" och andr araden var "specifikt nät till any". Det borde ha undantagit nätet från PBR.
I tredje raden används objektgrupp som source och destination. En större nätmask täckte nätverket som var med i rad 1 och 2 i objektgruppen.
Rimligtvis borde det ha gjort att nätverket specifiserat i rad 1 och 2 inte PBRades, men det gjorde det.

Jag fick bygga om till att använda ACL som match och route-maps som inte gör någon åtgärd för att lösa felet. Exempelkonf:

```
ip access-list extended NOT_PBR_IPv4
 permit ip 172.13.37.0 0.0.15.255 any
 permit ip any 172.13.37.0 0.0.15.255

route-map PBR_IPv4 permit 5
 match ip address NOT_PBR_IPv4
! Ingen åtgärd sker här och matchning slutar

route-map PBR_IPv4 permit 10
 match ip address PBR_IPv4
 set ip next-hop 192.168.18.18
```

Det gick att bekräfta via debug ip policy:

```
.Aug 26 13:09:23: SW2: IP: s=192.13.37.193 (Vlan11), d=172.13.37.99, len 1500, FIB policy match
.Aug 26 13:09:23: SW2: IP: s=192.13.37.193 (Vlan11), d=172.13.37.99, len 1500, PBR Counted
.Aug 26 13:09:23: SW2: IP: s=192.13.37.193 (Vlan11), d=172.13.37.99, len 1500, FIB policy rejected - normal forwarding
```

Routing Information Protocol, RIP

RIP är ett routingprotokoll som finns i flera former. RIP, RIPv2 och RIPng. Originalet är RIP och stöder IPv4, RIPv2 är en utveckling av RIP och RIPng är för IPv6.

RIPv2

RIPv2 är ett klasslöst distans-vektor protokoll som utbyter information enligt timers. Paketen som skickas är via multicast mot 224.0.0.9 och är UDP paket på port 520. Intervallen är per default 30 sekunder. Vid varje intervall skickas hela routing-databasen, men för on-demand länkar kan RIPv2 skicka hela databasen en gång och är sedan tyst tills förändring i databasen. Metric som används är hop-count där 15 är det absolut längsta som går. Autentisering sker via plain text password eller MD5. RIPv2 kan tagga rötter som redistribueras in i protokollet. RIPv2 har stöd för att annonsera ett annat next hop än sig själv.

image.png

Det går att konfigurera RIPv3 så att paket skickas till broadcast 255.255.255.255 istället för 224.0.0.9 med kommandot `ip rip v2-broadcast` i interface-konfigurationsläge.

Per default kan RIPv2 installera 4 stycken rötter i RIB för ECMP. Mellan 1 till 32 kan konfigureras under processen `router rip` med kommandot `maximum-paths x`.

Konfiguration

```
router rip
 version 2
 network 10.0.0.0
 network 172.31.0.0
 distribute-list ACL_DIST out
 distribute-list prefix PREFIX_DIST out


interface FastEthernet0/1
```

```
 ip rip authentication key-chain x
 ip rip authentication mode { text | md5 }
```

RIPng

RIP för IPv6!

image.png

Konfiguration

```
ipv6 unicast-routing
ipv6 cef

interface FastEthernet0/0
 ipv6 address 2001:DB8:1::1/64
 ipv6 rip 1 enable
 ! För att skicka enbart default route till granne
 ipv6 rip 1 default-information only

ipv6 router rip 1
 poison-reverse
```

EIGRP

Enhanced Interior Gateway Routing Protocol (EIGRP) är ett tidigare Cisco-propertiärt routingprotokoll som främst används i miljöer som enbart består av Cisco-produkter. Då protokollet tidigare var Cisco propertiärt har andra leverantörer ej implementerat protokollet. Protokollet ersatte Interior Gateway Routing Protocol år 1993.

EIGRP

Grunder

EIGRP transporteras ej med hjälp av UDP och TCP utan är ett eget protokoll med protokollnummer 88.
Reliable Transport Protocol (RTP) används för att säkerställa att paket har kommit fram till grannar korrekt.
Administrative Distance (AD) är 90.
Protokollet använder sig av en metric som enligt standard är baserad på bandwidth och ackumulerad delay för att välja bästa väg till destination.
Hello timers används för hur ofta en EIGRP-router ska annonsera sin närvaro. Hold timers används för att berätta för en granne hur länge de ska vänta på Hello paket innan ett grannskap anses vara dött.
Multicast används för att skapa grannskap. Multicastadresserna är 224.0.0.10 och FF02::A.
Hela EIGRP databasen utbyts när ett grannskap etableras. Därefter utbyts enbart förändringar.
Protokollet stöder autentisering via MD5 och SHA.
EIGRP kan tagga rötter och filtrera baserat på distribute-lists och route-maps.
Protokollet kan annonsera rötter med annat next-hop än sig själv.
Det går att manuellt summera rötter var som helst inom routingdomänen.

Loopförhindrande

För att ej skapa loopar i ett EIGRP nät som har EIGRP något som kallas för Feasability Conidition. Om en rot har en viss Feasable Distance , ex. 2000, så måste en annan route nå en lägre Received Distance (advertised distance + interfacet egna distance) för att anse vara en bättre väg utan loopar. Är Received Distance lägre än nuvarande Successor-routen (den bästa routen) så kommer routen ej att installeras i routing-tabellen.

Även Split Horizon och Poisoned Reverse är aktiverat per default i EIGRP. Split Horizon innebär att en route annonseras ej via samma interface som det togs emot på. Split Horizon kan behöva stängas av på interface som ansluter mot flera routrar, ex. på hub:en i en hub-and-spoke topologi. Stängs av per interface med `no ip/ipv6 split horizon eigrp` i classic mode och `no split-horizon` i named mode.

Poisoned Reverse är ett "tillägg" till Split Horizon, de arbetar ihop. Poisoined Reverse annonserar tillbaka mottagna rötter på samma interface som det tas emot på men med en såpass hög metric att nätverket ej är nåbart.

Metrics

För att räkna ut den bästa routen så används metrics. Metrics kommer i två varianter, classic och wide. De olika värdena hänvisas till som K-värden.

Classic metrics

Metric	**Förklaring**
Bandwidth (K1)	Statiskt värde som konfigureras med *bandwidth* kommandot i interface-konfigurationsläge. Finns inget konfigurerat så autoskapas ett. Vid fysiska interface så används den sanna hastigheten enligt länken. EIGRP använder sig av den lägsta bandwidth:en enligt hela roten. Enbart ett värde används alltså längst hela vägen oavsett hur många routrar som finns.
Delay (K2)	Statiskt värde som konfigureras med *delay* kommandot i interface-konfigurationsläge. Värdet 123 blir 1230 mikrosekunder. EIGRP använder sig av den totala ackumulerade delay-värdet enligt hela routingvägen.
Reliability (K3)	Ett värde mellan 1 och 255. 255 innebär 100 procentlig reliability. 230 innebär 90 procentlig reliability. Det minsta värdet av reliability används längs routingvägen.
Load (K4)	Ett värde mellan 1 och 255. 1 är ett tomt interface och 255 är ett fullt. Den maximala Txload:en (ej Rxload) används för att räkna ut Load-metric.
MTU (K5)	Den minsta MTU:n enligt hela routingvägen.
Hop count (K6)	Värde mellan 1 och 255. Enligt EIGRPs defaultkonfiguration är dock det maximala hopp-räknaren 100.

Komposit-metricen av K-värdena räknas ut genom den här formeln:

image.png

MTU (K5) och Hop count (K6) är inte med i formeln.

Wide metrics

Classic metric i EIGRP får problem när länkhastigheter överskrider 1 Gbit/s. Classic metrics ser ej skillnad på 10 gbit/s interface och snabbare interface. Wide metrics adresserar de här problemen.

Metric	**Förklaring**
Throughput	Likartad till Bandwidth. Räknas ut enligt formeln 65536×10 7 /Interface Bandwidth och hastigheten skrivs ut i Kbit/s. Uträkningen visar hur långsammare interfacet är jämfört med en 655.36 Tbit/s länk.
Latency	Likartad till Delay. Räknas ut enligt 65536×Interface Delay/10 6 och visas i pikosekunder.
Reliability	Identisk till Reliability i classic metrics.
Load	Identisk till Load i classic metrics.
MTU	Identisk till MTU i classic metrics. Annonseras men används ej.
Hop count	Identisk till Hop count i classic metrics. Annonseras men används ej i path selection. Används till att undvika eventuella routingloopar.
Extended Metrics	Placeholder för framtida användning. Inkluderar Jitter, Energy, och Quiescent Energy.

Uträkningen sker enligt denna formel:

image.png

Wide metrics används när EIGRP är konfigurerat med named mode. Routrar som kör wide metrics är bakåtkompatibla med classic metrics men föredrar att köra wide metrics.

Influera EIGRPs vägval

För att manueltl influera EIGRPs vägval kan man konfigurera bandwidth eller delay på interfacen. Dock så avråds det från att ändra bandwidth-värdet av flera anledningar:

- QoS kan använda sig av bandwidth värdet

- EIGRP begränsar sin trafik till att använda max 50% av bandwidth värdet, vid belastning kan alltså EIGRP-paket kastas istället för att skickas och därmed orsaka problem med grannskapet
- Ändrande av delay orsakar ej problemen beskrivna ovan

Alltså är rekommendationen att man enbart ska använda delay för att påverka EIGRPs vägval.

Det går att ändra hur många procent av ett interface bandbredd som EIGRP får använda med kommandot `bandwidth-percent x` i af-interface läge.

EIGRP Paket

EIGRP paket bärs direkt över IP, ingen UDP/TCP inblandad, med protokollnummer 88. Den maximala storleken på EIGRP paket tas ifrån interfacets maximala MTU.

Varje EIGRP paket innehåller en 20 byte lång header följd av innehåll i form av TLV:er, Type Length Value triplets. TLV:erna innehåller information om EIGRP och TLV versioner, K-värden, Hold timers, kontrollinformation för att främja säker multicasting och route reachability informationen. Det finns ingen dedikerat RTP (Reliable Transport Protocol) header. Istället används Flags, Sequence Number och Acknowledgement för den mesta av RTP funktionaliteten i EIGRP. Viss funktionalitet är via TLV:er.

image.png

Fält	**Förklaring**
Version	Ett 4 bitar stort värde för EIGRPs protokollversion. EIGRP protokollet har inte ändrats sedan dess skapande och är alltid satt till 2.
Opcode	4 bitar stort värde som specifiserar EIGRP pakettyp. 1 = Update, 3 = Query, 4 = Reply, 5 = Hello/Ack, 10 = SIA Query, 11 = SIA Reply.
Checksum	24 bitar sort värde som används för att utföra en sanity check på EIGRP paketet. Fältet baseras på EIGRP paketet utan dess IP header.
Flags	32 bitar stort värde som indikerar specifika flaggor. 0x1 = Init, 0x2 = Conditional Receive, 0x4 = Restart, 0x8 = End-of-Table
Sequence	32 bitar stort värde som innehåller ett sekvensnummer som används av RTP. Används för att säkerställa säkert leverans av paket.

Acknowledgement	32 bitar stort värde som används för RTP. Det senaste sekvensnummret i sequence emottaget från granne placeras här.
Virtual router ID	16 bitar stort värde som identifierar den virtuella routern paketet är associerat med. 0x1 = Unicast Address Family, 0x2 = Multicast Address Family, 0x8000 = Unicast Service Address Family
Autonomous System Nunber	16 bitar stort värde som identifierar EIGRP domänen.
Type-Length-Value (TLV)	Används för att bära route-information och DUAL information. Följande TLV:er finns: 0x0001 EIGRP Parameters (General TLV Types) 0x0002 Authentication Type (General TLV Types) 0x0003 Sequence (General TLV Types) 0x0004 Software Version (General TLV Types) 0x0005 Next Multicast Sequence (General TLV Types) 0x0102 IPv4 Internal Routes (IP-Specific TLV Types) 0x0103 IPv4 External Routes (IP-Specific TLV Types) 0x0402 IPv6 Internal Routes (IP-Specific TLV Types) 0x0403 IPv6 External Routes (IP-Specific TLV Types) 0x0602 Multi Protocol Internal Routes (AFI-Specific TLV Types) 0x0603 Multi Protocol External Routes (AFI-Specific TLV Types)

TLV:er är ett specifikt sätt att lagra och skicka olika typer av information i ett enda datagram. En TLV innehåller:

- Type, en numerisk kod som indikerar vad för information som finns i Value-fältet
- Length, den total storleken av TLV fälten. Notera att vissa protokoll, ex. EIGRP, enbart sparar längden av Value-fältet i Length
- Value, bytes av olika storlek som innehåller den intressanta informationen

I EIGRP så innehåller varje Internal och External Route TLV ett enda route-värde. Update, Query, Reply, SIA-Query och SIA-Reply paketen innehåller minst en sådan TLV för att annonsera ett specifikt nätverk eller för att fråga om det.

EIGRP Pakettyper

Sju olika pakettyper finns i EIGRP. Dessa syns i Opcode värdet i EIGRP headern. Följande pakettyper finns:

1. Hello
2. Acknowledgement
3. Update
4. Query
5. Reply

6. SIA-Query
7. SIA-Reply

Update, Query, Reply, SIA-Query och SIA-Reply kallas för *reliable packets* för att EIGRP ser till att de levereras i rätt ordning med RTP.

Vad för typ av paket som har skickats kan kontrolleras med kommandot `show ip eigrp traffic`. Används EIGRP i en VRF så är kommandot `show ip eigrp vrf VRF-NAMN traffic`.

```
c6807#show ip eigrp vrf VRF-NAMN traffic
EIGRP-IPv4 VR(EIGRP-1) Address-Family Traffic Statistics for AS(1)
        VRF(VRF-NAMN)
  Hellos sent/received: 1474623/2949935
  Updates sent/received: 4878/5343
  Queries sent/received: 83/0
  Replies sent/received: 0/83
  Acks sent/received: 5051/4617
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 1258
  PDM Process ID: 1155
  Socket Queue: 0/10000/4/0 (current/max/highest/drops)
  Input Queue: 0/10000/4/0 (current/max/highest/drops)
```

Hello

EIGRP skickar Hello paket ut på interface som har lagts till i EIGRP processen och som inte är passiva. Hello meddelanden används för att identifiera grannar, verifiera att grannarna är kompatibla och som keepalive när ett grannskap är etablerat. Hello paket skickas via multicast till 224.0.0.10 eller FF02::A beroende på IP-version. Om statiska grannar är konfigurerade så skickas Hello paketen som unicast till den konfigurerade grannen. Intervallen som Hello-paketen skickas kan modifieras och är default 5 sekunder. På NBMA interface eller med bandbredd 1544 kbit/s eller mindre är default-intervallen 60 sekunder. Paketen har Opcode 5 och kommer inte att få en Acknowledgement tillbaka via RTP.

Acknowledgement

Acknowledgement (ACK) paket skickas för att bekräfta vissa EIGRP-paket för att säkerställa leverans. ACK Skicaks som svar till Update, Query, Reply, SIA-Query och SIA-Reply paket och skickas som unicast till sändaren. De använder samma Opcode som Hello, Opcode 5. Det ser i princip ut som ett Hello paket men innehåller inga TLV:er. Fältet Acknowledgement Number i EIGRP paket som skickas innehåller värdet från Sequence number i mottaget paket.

Update

EIGRP Update paket innehåller uppdateringar om routing information. De kan skickas både som multicast och unicast. På interface där det enbart finns en granne så skickas paketen som unicast. På interface där och när flera nya grannar märks samtidigt (ex. DMVPN) så används multicast. Efter full synkning av databasen så skickas nya uppdateringar som multicast. Routern förväntar sig ett Ack-svar på Update paket Om en granne ej skickar ett Ack-svar tillbaka så kommer routern att skicka om paketet via Unicast. På interface som är konfigurerade som P2P och mot statiskt konfigurerade grannar kommer unicast alltid att användas. Update paket använder sig av Opcode 1.

Query Paket

Query paket skickas när en granne letar efter den bästa routen till en destination. Ack-svar förväntas på Query paket. De kan skickas via multicast och unicast. Per default på multiaccess-interface (ex. Ethernet) så skickas paketen som Multicast. Kommer ingen Ack så kommer routern att skicka om paketet via unicast. På P2P interface och mot statisk konfigurerade grannar kommer paketen att gå som unicast. Query paket använder sig av Opcode 3.

Reply

Reply paket innehåller svaret på frågan i Query paket. Ack-svar förväntas. De skickas alltid som unicast till frågeställaren. Opcode 4 används.

SIA-Query och SIA-Reply

SIA står för *Stuck In Active*. Dessa opcodes används när en router ej har mottagit ett EIGRP Reply paket som svar på en EIGRP Query. SIA-Query frågar om en router fortfarande letar efter svaret på en Query. Om grannen fortfarande kör DUAL och letar efter ett bra svar svarar den direkt med en SIA-Reply. Då nollställs tiden som frågeställaren väntar på svar. Paketen skicas som unicast och Ack förväntas. SIA-Query använder Opcode 10 och SIA-Reply använder Opcode 11.

Reliable Transport Protocol

Hur RTP fungerar har beskrivits ovan, men bygger på att vissa Opcodes förväntar sig Ack-svar. I svarspaketet har siffran från Sequence Number kopierats till Acknowledgement Number.

RTP har en funktion som kallas Conditional Receive. Det låter EIGRP dela upp grannar på ett multiaccess-interface (ex. Ethernet med flera EIGRP grannar) i två grupper. I den ena gruppen finns grannar som har skickat Ack på samtliga multicast EIGRP paket och i den andra de som har misslyckats att svarar på minst 1. För att inte behöva vänta på samtliga grannar för att gå vidare med nästa paket så skickas paket med en flagga till den grupp som har Ack:at samtliga tidigare. Utan Conditional Receive hade EIGRP varit tvungen att vänta på samtliga grannar innan EIGRP kan skicka nästa paket. Detta åstakommes med specifika TLV:er, Sequence TLV och Next Multicast Sequence TLV. I Sequence TLVn finns en lista med IP-adresser. Finner en router sin egen IP adress i den listan kommer den att ignorera paketet. En router som ej finner sig själv i listan kommer placera sig själv i Conditional Receive mode (CR-mode).

Routergrannskap

Per default så hittar EIGRP grannar automagiskt. Det går även att manuellt konfigurerat grannar.

Dynamiska grannskap etablera genom att EIGRP är aktiverat på ett icke passivt interface vilket triggar att Hello-paket skickas till 224.0.0.10/FF02::A. När potentiella grannar upptäcker varandra så kontrolleras först kompabilitet. Följande värden måsta matcha:

1. Autentisering
2. Likadant konfigurerade K-värden
3. AS-nummer
4. Primära adresser för grannskap
5. Samma IP-nät på ett enda subnät

Hello-intervallen, alltså hur ofta ett Hello paket skickas, och Hold-intervallen, hur länge en router väntar på ett Hello-paket innan grannskapet anses dött, behöver inte matcha i EIGRP. Detta då Hold-intervallen som konfigureras på Router A indikirerar hur länge Router A:s grannar, ex. Router B, ska vänta på Hello paket från Router A.

```
! Konfiguration klassisk EIGRP

interface TenGigabitEthernet1/1/1
 ip hello-intervall eigrp 1 1
 ip hold-time eigrp 1 3

! Konfiguration named EIGRP

router eigrp EIGRP-NAMN
 !
 address-family ipv4 unicast autonomous-system 1
  !
  af-interface TenGigabitEthernet1/1/1
   hello-interval 1
   hold-time 3
```

image.png

Proceseen börjar när R2 mottager ett Hello-paket från R1. Vid mottagning av Hello-paket kommer R2 att skicka ett eget Hello-paket och placera grannen i läget Pending. Även R1 placerar R2 i Pending. Sedan skickas ett tomt Update paket från R2, det innehåller alltså inga TLV:er, med Init flaggan. Init flaggan indikerar att det är ett nytt grannskap och grannen ska skicka över hela EIGRP-databasen efter etablering. R1 skickar också ett tomt Update paket med Init-flaggen. Notera att Ack-fältet innehåller Seq-värdet som kom med R2s paket. R2 flyttar R1 från Pending till Up och

skickar ett Ack. R1 flyttar nu R2 från Pending till Up. Nu kommer hela EIGRP databasen att utbytas via Update paket. Därefter skickas enbart uppdateringar vid förändringar eller queries samt Hello-paket.

Grannskap kan kontrolleras via `show eigrp address-family ipvx detail`. Byt ut x till 4 eller 6, beorende på vilket routing-protokoll som används. Släng in `vrf VRF-NAMN` efter adressfamiljen om en VRF används. Detail är inte så mycket större än utan detail.

```
! Exempeloutput


catalyst#show eigrp address-family ipv4 vrf VRF-NAMN neighbors  detail
EIGRP (EIGRP) Address-Family Neighbors for AS(1)
          VRF(VRF-NAMN)
H   Address             Interface           Hold Uptime   SRTT   RTO  Q  Seq
                                          (sec)        (ms)       Cnt Num
1   1.2.3.4         Vl1337               12 6w1d       1   100  0  1641
   Static neighbor
   Version 23.0/2.0, Retrans: 0, Retries: 0, Prefixes: 479
   Topology-ids from peer - 0
0   5.6.7.8         Vl1337               12 6w2d       1   100  0  1748
   Static neighbor
   Version 23.0/2.0, Retrans: 1, Retries: 0, Prefixes: 12
   Topology-ids from peer - 0
Max Nbrs: 0, Current Nbrs: 0
```

H-värdet (Handle) är ett värde som EIGRP processen tilldelar grannar. Address är grannens adress. Interface är vilket interface grannskapet är etablerat över. Hold visar hur många sekunder det är kvar på grannskapet innan det tas ned, förutsatt att det inte kommer ett nytt Hello paket såklart. Uptime visar hur länge grannskapet har varit aktivt. SRTT (Smooth Round Trip Time) är en uppskattning för hur lång tid det tar att skicka ett paket till grannen och få tillbaka en Ack. RTO (Retransmit Time Out) visar hur länge routern väntar på en Ack för ett omskickat unicast paket när tidigare paket ej fick en Ack tillbaka. Q Cnt visar hur många paket som har förberets för att skicka och paket som har skickats men som ingen ACK har kommit tillbaka från grannen. Bör vara 0 i ett bra nätverk (vilket det även är i exemplet ovan). Sekvensnumret är sekvensnumret i det senast skickade EIGRP paketet.

Diffusing Update Algorithm (DUAL)

DUAL är en konvergensalgoritm som varje EIGRP router använder sig. av. Algoritmen räknar ut den bästa vägen till rotdestinationen och förhindrar loopar.

EIGRPs databas kallas för *topology table*. Varje post i databasen innehåller:

1. Nätverk (adress och nätmask)
2. Feasible Distance för prefix
3. Adress för EIGRP granne som annonserar nätverk samt utgående interface mot grannen
4. Metrics för nätverk annonserat av grannen (advertised distance) samt metrics för att nå nätverket genom lokala interfacet (received distance)
5. Status för nätverket
6. Extra information om nätverket (flaggor, nätverkstyp, origin m.m.)

Databasen populeras av lokalt injicerade nätverk, det vill säga direkt anslutna EIGRP nätverk och rötter som redistribueras lokalt, samt innehållet från samtliga EIGRP Update, Query, Reply, Sia-Query och SIA-Reply paket. För varje nätverk kommer EIGRP att köra DUAL-uträkning för att hitta den väg med minst kostnad och som är utan loopar till destinationen. Vinner nätverket i EIGRP sedan mot rötter genom andra källor så kommer den att installeras i routing-tabellen.

Varje nätverk har en status. De kan vara Passive (bra) eller Active (dåligt). Nätverk som är Passive är DUAL klara med och bästa vägen hittats. Nätverk som är Active letar EIGRP efter bästa vägen till genom Query-paket. När ett nätverk är Active får inte routern modifiera nuvarande post i routingtabellen. Status Active försvinner först när alla grannar har svarat på Query-paketet med en Reply.

```
! Det går att slänga på all-links på slutet av show-kommandot för att se rötter som inte har klarat Feasability Condition.

catalyst#show eigrp address-family ipv4 vrf VRF-NAMN topology
EIGRP-IPv4 VR(EIGRP-1) Topology Table for AS(1)/ID(5.5.5.5)
          Topology(base) TID(0) VRF(-NAMN)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 1.1.1.0/30, 1 successors, FD is 656670720, tag is 8928
        via 11.11.11.11 (656670720/656015360), Vlan1337

...

catalyst#show eigrp address-family ipv4 vrf -NAMN topology 1.1.1.0/30
EIGRP-IPv4 VR(EIGRP-1) Topology Entry for AS(1)/ID(5.5.5.5)
          Topology(base) TID(0) VRF(-NAMN)
EIGRP-IPv4(1): Topology base(0) entry for 1.1.1.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 656670720, RIB is 5130240
  Descriptor Blocks:
  193.44.81.54 (Vlan1337), from 11.11.11.11, Send flag is 0x0
```

```
    Composite metric is (656670720/656015360), route is External
    Vector metric:
      Minimum bandwidth is 1000 Kbit
      Total delay is 20000000 picoseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 1
      Originating router is 172.31.255.75
    External data:
      AS number of route is 65534
      External protocol is BGP, external metric is 10
      Administrator tag is 8928 (0x000022E0)
```

I exemplet ovan så finns bara 1 successor till 1.1.1.0/30 och Feasible Distance är 656670720. Det som syns nedanför är till vänster Computed Distance och till höger Reported Distance/Advertised Distance, alltså uträknad metric inklusive vägen till grannen och den annonserade metricen från grannen. Finns flera successors, eller feasible successors, så syns de här.

Värt att veta om Feasible Distance är att det är en historisk siffra som ändras när en ny successor har en bättre computed distance. Skulle nuvarande successor ha en Computed Distance på 2000, Feasible Distance är 2000, men ex. delay ändras så att Computed Distance blir 1000 blir nu Feasible Distance 1000. Men om dess Computed Distance går upp till 2000 igen och ingen annan successor är bättre så kommer Feasible Distance att ligga kvar på 1000.

DUAL körs när en ny granne kommer upp på samtliga rötter som annonseras via Update/Query/Reply/Sia-Query/SIA-Reply. När en granne går ned så sätts alla rötter mottagna från den till metric infinity - alltså onårbara. Finns en Feasible Successor så kommer den direkt att befodras till en Successor. Det sker lokalt i routern, ingen granne är inblandad, och kallas därmed *local computation*.

Finns ingen feasible successor och bästa hittade väg i EIGRP topologin är via en granne som ej är en feasible successor kan den routen inte användas direkt då en routingloop kan skapas. Nuvarande route i RIB låses, FD sätts till nuvarande CD genom den oförändrade Successorn och nätverket markeas som Active. Query paket skickas till grannar för att hitta bästa vägen. Finns en Feasible Successor eller Successor som inte går via frågeställaren kommer grannarna att svara tillbaka med Reply till frågeställaren och en ny routinväg är etablerad. DUAL har inte körts på routrar som redan har en successor/FS. Finns däremot ingen FS/Successor som inte går via frågeställaren kommer nätverket att markeras som Active, DUAL-uträkning körs och Queries kommer att skickas till grannar. När alla routrar har svarat med Replies kan till slut nätverket markeras som Passive, antingen via originalposten som fanns i EIGRP topologin eller via en ny väg. Nätverket i RIB låses upp och ny route installeras.

Stuck-In-Active (SIA)

SIA är ett läge där en router har skickat Query men inte mottagit en Reply, nätverket är alltså markerat som Active i EIGRP topologin. Varje EIGRP router måste vänta på Replies från sina grannar innan de kan skicka tillbaka sitt egna Reply till den ursprunliga frågeställaren. Skulle en router längs vägen bete sig konstigt och inte svara så får frågeställaren inte något Reply alls. När en Query först skickas så börjar Active timer att ticka för nätverket routern frågar om. När tiden för timern är slut (default 3 minuter, kan konfigureras med `timers active-time x` i EIGRP-kontext) då ingen Reply har mottagits så anses nätverket vara SIA och grannskap mot grannar som ej svarar tas ned. Det kan alltså leda till totalt trafikavbrott.

För att undvika detta finns SIA-Query (opcode 10) och SIA-Reply (opcode 11). Efter halva Active timer-tiden så skickas en SIA-Query som frågar "Jobbar du fortfarande på min Query?". En korrekt fungerande router svarar med en SIA-Reply med innehållet "Ja, jag inväntar Replies" eller "Nej, uträkningen är färdig, här är min metric till destinationen". Active timer nollställs vid mottagande av SIA-Reply. Max tre stycken SIA-Queries kan skickas. Är uträkningen inte klar efter tre stycken SIA-Queries kommer grannskapet att tas ned. Skulle ingen SIA-Reply kommer på SIA-Query kommer grannskapet att tas ned när Active timer är slut.

EIGRP Named Mode

Named mode är hur EIGRP ska konfigureras idag. Äldre varianten kallas för classic mode. Det går att uppgradera en miljö i classic mode till named mode med kommandot `eigrp upgrade-cli`. Det går att köra EIGRP processer samtidigt i både named och classic mode.

I classic mode är konfigurationsplatserna spretiga. Viss konfig är under EIGRP processen och viss konfig i interfaceläge. I named mode är all konfig samlad på samma ställe.

Exempelkonfiguration EIGRP named mode:

```
router eigrp NAMED-MODE
 !
 address-family ipv4 unicast autonomous-system 1337
  !
  af-interface default
   passive-interface
  exit-af-interface
  !
  af-interface Tunnel1
   authentication mode md5
   authentication key-chain KEY-CHAIN
  hello-interval 1
   hold-time 3
```

```
 no passive-interface
exit-af-interface
!
topology base
 distribute-list prefix PREFIX-LIST out Tunnel1
 maximum-paths 6
 variance 4
 redistribute static
exit-af-topology
network 10.13.37.1 0.0.0.0
eigrp router-id 1.3.3.7
exit-address-family
!
address-family ipv6 unicast autonomous-system 1337
 !
 af-interface default
  shutdown
  passive-interface
 exit-af-interface
 !
 af-interface Loopback0
  no shutdown
 exit-af-interface
 !
 af-interface TenGigabitEthernet1/0/1
  hello-interval 1
  hold-time 3
  no passive-interface
 exit-af-interface
 !
 af-interface Vlan1337
  no shutdown
 exit-af-interface
 !
 topology base
  distribute-list prefix-list PREFIX_LIST out TenGigabitEthernet1/0/1
  maximum-paths 6
  variance 4
  redistribute static metric 1 1 1 1 1
 exit-af-topology
```

```
 eigrp router-id 1.3.3.7
exit-address-family
```

För IPv4 används fortfarande network kommandot för att lägga till interface i EIGRP processen. För IPv6 är det "no shutdown" som gäller för att lägga till interface i EIGRP processen. Per default är *alla* IPv6 interface med i EIGRP processen. En `shutdown` under `af-interface default` behövs för att ändra det.

Router ID

EIGRP använder sig av router ID för att identifiera en router. Värdet är 4 bitar stort och ser därmed ut som en IPv4 adress. Samtliga nätverk som annonseras i EIGRP har router ID med i paketet. En router som mottager ett nätverk med sitt eget router ID kommer att kasta det paketet. Det är en loop-förhindrande mekanism.

Ett router ID kan (och bör) konfigureras statiskt med `eigrp router-id x.x.x.x` kommandot i EIGRP processen. Konfigureras det ej manuellt tas automagiskt den högsta IPv4-adressen på ett Loopback interface som är uppe till router ID. Finns inget loopback interface tas den högsta IPv4-adressen från övriga IP adresser. Ett autokonfat router ID ändras ej om man ex. tar bort IPv4 adressen på interfacet. Det ändras när EIGRP processen startar om eller om ett manuellt router ID konfigureras. Nuvarande router ID kan hittas med `show eigrp protocols` och med `show ip protocols`.

Unequal-Cost Load Balancing (Variance)

En unik funktion som EIGRP tillför är unequal-cost load balancing, alltså möjligheten att lastbalansera trafik över länkar som inte är lika snabba. Det som möjliggör detta är Feasible Successors. Detta då trafik som går via feasible successors ska vara en loopfri väg. Det konfigureras med `variance x` i `topology base`. Om x = 1 utförs ingen unequal-cost load balancing. Trafikmängden på de olika interfacen är ej jämt fördelad, de snabbare interfacen får hantera mer trafik. Antalet vägar som trafik kan gå konfigureras med `maximum-paths x` kommandot. Det gäller även för trafikvägar med samma cost.

Add-Path

Vid användning av ex. DMVPN kan en hub känna till två lika bra vägar till en spoke. Dock kan hub:en per default inte annonsera dessa två vägar till andra spokes. De andra spokes:n har per default enbart en väg dit. För att tillåta en hub annonsera två vägar finns add-path. Det aktiveras på af-interface nivå (funktion finns enbart i named EIGRP) med kommandot `add-paths x` där x är en siffra mellan 1 och 4. maximum-paths måste vara satt till minst samma nivå och split-horizon måste vara avstängt på interface mot hubbarna. Add-Path går att kombinera med `Unequal-Cost Load Balancing`.

Stub

Stubar är ett sätt att snyggt öka skalbarheten och stabiliteten på ett nät. En stub router annonserar per default enbart sina egna nätverk och inte de den har lärt sig från grannar. Det går att annonsera andra nätverk genom att använda sig av en *leak-map*. Det går att definiera exakt vilka av sina egna nätverk en EIGRP stub router ska annonsera i `eigrp stub` kommandot med tilläggen *summary, connected, static, redistributed* och *received-only*. Att en router är stub annonseras via en specifik TLV i Hello-paketen vilket gör att en granne aldrig skickar en query till stub routrar.

Om en stub router skulle mottaga en query så kommer den ändå att processa den, det beror dock på vad för nätverk queryn frågar efter. Queries som routern själv har skickat kommer hanteras som vanligt. En query för lokala nätverk (summary/connected/static/redistributed) samt de som annonseras via en leak-map kommer stubroutern att hantera som vanligt. För ett nätverk som inte faller under nyss nämnda kategorier men som routern känner till kommer routern att svara med prefixet men med en oändlig distans, dvs ej nåbar. För nätverk som routern inte känner till kommer den att svara med oändlig distans. En stubrouter kan motta queries om en granne kör gammal kod som inte känner igen Stub TLVn eller om den sitter på ett delat segment med flera stub-routers där en stub router skickar Querien. För multiaccesslänkar där vissar är stub och andra inte kommer EIGRP att skicka queries via unicast till de som inte är stub eller så används Conditional Receive i RTP för att skicka multicast som enbart processas av icke stub routrar.

Stub är jättebra att använda mot siter som kanske enbart har en upplänk och inget bakom sig. Det ser till att suboptimal routing ej sker, stub routrar med sämre länkar kommer aldrig bli transit-routrar och antalet Query paket minskas i domänen vilket kan leda till att man undviker SIA-problem.

Per default så annonseras connected och summary när man konfigurerar `eigrp stub`. Verifiering kan ske med `show ip protocols` och grannar som är stubbar kan hittas med show `eigrp address-family ipvx neighbors detail`.

Stubnät	**Förklaring**
receive-only	Stubroutern annonserar inga prefix. Statisk routing eller NAT behövs.
leak-map x	Matchar en route-map (som matchar ACL eller prefix list) för att annonsera specifika nät som EIGRP känner till.
connected	Annonsera direktanslutna nät
static	Annonsera statiska rötter. Redistribueringskommando behövs också.
redistributed	Tillåt att rötter som redistribueras in i EIGRP annonseras.

Att en router är stub påverkar inte vad som annonseras uppströms till routern. Vill man ex. begränsa routingtabellen krävs route-filtrering och/eller route-summering.

Route Summarization

En stark fördel i EIGRP kontra ex. OSPF är att ruttsummering kan ske varsomhelst på interface-nivå. Korrekt implementerad ruttsummering, exempelvis att annonsera större block mot stub:ar, gör att antalet queries i ett nätverk minskar.

Konfigureringen sker i interface-läge (af-interface i named EIGRP) med kommandot `summary-address 10.0.0.0 255.0.0.0`. En *leak-map* går att slänga på på slutet för att skicka med vissa specifika rötter trots summeringen.

När en summering installeras så skapas också en discard route (nullroute) till Null0 för nätverket specifiserat i summeringen. Den discard routen får AD 5 per default. Det kan hända att den får en bättre AD än en manuellt konfigurerad summeringsrutt som då göms av denna. Antingen får man ändra AD:t på den manuella konfigureringsrutten eller så kan man ändra AD:t på EIGRPs summeringsrutt med kommandot `summary-metric 10.0.0.0 255.0.0.0 distance X` i `topology base`. Ändra ej AD:t till 255. IOS installerar då ej rutten och annonserar den ej.

EIGRP använder sig den lägsta metricen från de rutter som täcks av summeringen vid annonsering av summeringen. Varje gång den sämsta metricen ändras måste summeringsrutten annonseras på nytt med en ny metric. Det går att sätta en statisk metric med kommandot `summary-metric x` i `topology base` för att slippa det beteendet då i större miljöer innebär det många uträkningar och omskick.

Den äldre automatiska summeringen är avstängd per default. I äldre versioner (innan IOS 15.0(1)M) behöver den stängas av med `no auto-summary` i EIGRP konfläge.

Passive interface

Ett passivt interface är ett interface som finns med i EIGRP processen men som inte skickar Hello-paket. Per default skickar samtliga EIGRP interface hello paket. Passive interface ska användas på samtliga interface som inte är tänkt att gå mot en EIGRP granne.

Best practice är att i `af-interface default` konfigurera `passive-interface` och sedan gå in under de af-interface som går mot andra EIGRP routrar och konfigurera `no passive-interface`. Det gör att man inte råkar skicka Hellos ut på interface mot ex. klienter.

Graceful shutdown

Graceful shutdown är en funktion som tillåter EIGRP routrar att annonsera till sina grannar att EIGRP avaktiveras. Det kan vara per interface, adressfamilj eller hela processen. Paketet är ett Hello-paket med alla K-värden satta till 255. Funktionen är påslagen per default och går ej att ändra på.

Graceful restart

Stöd för NSR/GR finns. Aktiveras genom att slå `nsf` i adressfamiljen.

Verifiering går med show ip protocols | sec EIGRP NSF.

```
c9300#show ip protocols | sec EIGRP NSF
  EIGRP NSF enabled
    NSF signal timer is 20s
    NSF converge timer is 120s
```

Timers kan sättas enligt:

```
router eigrp EIGRP
 address-family ipv4 unicast autonomous-system 1337
  nsf
  timers nsf signal 25
  timers nsf converge 100
  timers graceful-restart purge-time 150
```

Autentisering

Klassisk EIGRP har stöd för MD5 autentiseringen. Named mode har stöd för MD5 och SHA-autentisering. Key chains används. Med SHA-autentisering kan man även konfigurera lösenord direkt på interfacen. Exempelkonfiguration finns längre upp för named mode. För classic konfigureras `ip authentication mode eigrp` och `ip authentication key-chain eigrp` på interfacen.

Default route

EIGRP har inte något kommando för att skapa en default route. Istället så kan man redistribuera in en default route eller använda sig av manuell summering. Då det är en redistribuerad route blir AD:t 170.

Det går även att annonsera ut en default-route via ett interface genom summering, ex. `summary-address 0.0.0.0 0.0.0.0`. Dock så kommer då en route för 0.0.0.0/0 installeras till Null0 med AD 5. Ens

redan installerade defaultroute kan alltså bli dold av denna.

EIGRP Over the Top

EIGRP OTP är ett sätt att köra EIGRP mellan routrar som ej är direktanslutna, ex. över ett L3 VPN nät via en SP. EIGRP OTP använder sig av LISP för dataforwarding, UDP enkapsulerad trafik. Det går även att köra route-reflectorer i EIGRP OTP, de fyller samma funktion som i BGP i en hub-and-spoke topologi. Grannar konfigureras statiskt med neighbor-kommandot.

```
! Exempelkonf


router eigrp EIGRP
 !
  address-family ipv4 unicast autonomous-system 1337
   !
   af-interface GigabitEthernet0/0
    no next-hop-self
    no split-horizon
   exit-af-interface
   !
   topology base
   exit-af-topology
   ! Nedan konfas på spokes
   neighbor 1.1.1.1 GigabitEthernet0/1 remote 100 lisp-encap
   ! Nedan kan konfas på route-reflektor för att dynamiskt finna grannar
   remote-neighbors source GigabitEthernet0/0 unicast-listen lisp-encap
```

En route till grannens IP-adress behövs inte. Då interfacet som används specifiseras i EIGRP konfigurationen så används det för att skickas iväg paketet med rätt IP header. Dock vid Ethernet interface kommer routern skicka en ARP fråga ut på det interfacet, så det är nog bäst att specifisera i routingtabellen. Om man inte vill leva med bös som Proxy ARP.

Det går även att lägga tillen ACL för remote neighbors på route reflektorn.

Logging

Loggingnivå konfigureras under routingprocessen. Det som kan konfigureras är eigrp event-log-size (maxstorlek på EIGRP loggen), event-logging (logga routing-events), log-neighbor-changes (logga när grannskap går upp/ned) och log-neighbor-warnings. Loggen kan kommas åt geom att slå `show eigrp address-family ipvx events`.

Route Filtering

Det går att manipulera vilka rötter som skickas ut eller installeras via vilket interface som helst. Det går även att konfigurera för en hel adressfamilj under *topology base*. `distribute-list` kommandot används. Det går att filtrera på ACLer, prefix-listor och route-maps.

Syntax:

- ACLs: distribute-list acl-number | acl-name { in | out } [interface]
- Prefix lists: distribute-list prefix prefix-list-name { in | out } [interface]
- Route maps: distribute-list route-map route-map-name { in | out } [interface]

Offset list

En offset list ändras för att modifiera metric på specifika nätverk inkomna via visst interface eller för hela processen istället för ex. att ändra hela interfacets delay. En ACL används för att identifiera trafiken. Syntax är ex. `offset-list 10 out 100 serial 0/0`. 100 läggs då till på rötter identifierade i standard ACL 10 för rötter som annonseras ut på interface serial0/0.

Rensa EIGRP databasen

`clear ip route *` tömmer routingtabellen men inte EIGRP topologin. För att tömma hela EIGRP topologin och ta ned samtliga grannskap kan man använda sig av `clear eigrp address-family ipvx neighbors`.

Config register

Vid händelse att man behöver bryta sig in i en Cisco-burk kan man ändra config-register värdet till ett som gör att enheten ignorerar startup-config.

Boota om routern och skicka break-sequence så att man hamnar i Rommon. Ändra konfig-register till 0x2142. Boota om routern och kopiera över startup-config till running. Sedan går det bra att ändra tillbaka config-reg direkt i IOS-XE till standard 0x2102.

"Nya" syntax, username, enable secret, password service-encryption

Username & enable secret

För att undvika att ens konf för lokala användare och enable lösenordet blir deprecated ska det konfigureras enligt nedan:

```
enable algorithm-type sha256 secret LÖSENORD

username LOCAL_USER algorithm-type sha256 secret LÖSENORD
```

Password service-encryption

Det här upptäckte jag vid uppgradering av Catalyst 9300 till 17.12.3 (från 17.9.4a) där OSPFv3 kryptering slutade att fungera. Konfigurationen försvann från interface, följande felmeddelande kom vid försök att aktivera det på nytt:

```
service password-encryption is enabled, OSPFv3 AES-CBC 256bit key encryption configuration is not supported

% OSPFv3: Encryption was not enabled

! Vid försök att återaktivera efter att kryptering är pålagt på interfacet
catalyst(config)#service password-encryption
%service password-encryption failed due to un-supported configuration
Aug  5 09:35:18: %OSPFv3-3-IPSEC_AES_CBC_256BIT_POLICY_CONFIGURED_INTERFACE: service password-encryption is not supported as IPSEC encryption policy SPI 256 with AES-CBC 256bit is configured under interface Vl1337
```

Tar man bort service password-encryption så lirar det som tänkt. Men man vill ju ha någon typ av kryptering av klartextlösenord.

Det går bra genom att användning av `password encryption aes`. Lösenorden kommer då att bli krypterade enligt typ 6, vilket är bättre än typ 5 och typ 7. super-secret-password är då en sträng som används för att kryptera klartextlösenord. Informationen kommer från den här tråden.

```
!
configure terminal
  password encryption aes
  key config-key password-encrypt super-secret-password
end
!
```

Verifiering kan göra genom `show running-config | section x`. Byt ut x mot _5_ och _7_. Finns inga kvar så är det klart.

Följande kommer ej att krypteras med password encryption aes:

- enable secret 9
- BGP MD5 (använd istället BGP TCP Authentication Option)
- OSPF MD5, använd istället HMAC autentisering
- HSRP/VRRP key-string, använd istället authentication key-chain

IPv6

Generellt dokument med olika IPv6-grejer.

Grunder

Header

En IPv6 header är större än IPv4 header. IPv4 headern är 20 byte och IPv6 headern är 40 bytes. Varje IPv6 adress är 16 bytes stor. TTl har bytt namn till hop limit och IP protocol heter nu next-header. Ingen fragmentering är tillåten för IPv6-paket i routrar, däremot kan hostar utföra framgnetering så länge de stöder vissa IPv6 Extension Headers.

Adresstyper

Link-local adress, FE80::/10 (FE80 till FEBF::FFFF), används för lokal kommunikation över en länk. Måste finnas på samtliga IPv6-interface.

Site-local adress, FEC0::/10 (FEC0:: till FEFF:FFF::), privata adresser och bör ej användas.

Unique-local adress, FC00::/7 (FC00:: till FDFF:FFFF::), privat adresser som ersätter site-local. Bör helst inte användas.

Multicast adress, FF00::/8 (allt som börjar med FF), för multicasttransport. Inom den andra byten så representerar den första hex-symbolen speciella flaggor medans den andra hex-symbolen representerar "scopet". Flaggorna är binärt "0RPT". Den viktigaste är 0 och betyder ingenting.

1. R indikerar om IPv6 bär en PIM RP-adress, det används för embedded RP och RP-adressen signaleras i IPv6 multicastgruppen.
2. P indikerar om multicastadressen är tilldelad baserad på nätverksprefixet. Den används för embedded RP och nätverksprefixet är inbäddat i IPv6-adressen. Om R är 1 så måste P också vara 1.
3. T indikerar om multicastgruppen transient eller ej. 0 indikerar att det är en känd multicastgrupp och 1 att det är en dynamiskt tilldelad.

Scopen är enligt följande och används för att dela in multicast i administrativa regioner:

1. Interface local. Används för loopback-transmission av multicast
2. Link-local. Kommunikation sker över ett segment. Används av ex. IGP, PIM, neighbor discovery
4. Admin-local. Det minsta scopet som kan konfigureras. Den här multicasttrafiken går att route:a. Användbar för att ex. route:a trafik till vissa enheter inom en site.
5. Site-local. Ska användas inom en site. Ex. för multicasttrafik som är lokalt inom ett kontor. PIM dense-mode stöds ej i IPv6 på Cisco-plattformar så site-local sparse-mode kan vara ett bra alternativ till deta.
8. Organization-local. För trafik inom en organisation, ex. mellan kontor:
E. Global-scope. Kallas ibland för "VPN scope" av Cisco och har ingen gräns.

Anycast adress. Konceptet Anycast finns för IPv4 men inte på ett LAN-segment. I IPv6 finns konceptet även för LAN. Att konfigurera en Anycast-adress är i princip likadant som en unicastadress fast med Duplicate Address Detection (DAD) avstängt. När en host försöker att få L2-adressen till en anycastadress kan vilken nod som helst på LAN:et svara.

Soliciated-node adress. En adress inom link-local spannet som har räknats ut som en funktion från en nods unicast och anycast adresser. Adresserna skapas genom att ta de lägsta 24 bits från en IPv6 adress och lägga till dem på slutet till prefixet FF02::1:FF00::/104 (FF02::1:FF00:: till FF02::1:FFFF). Varje nod måste gå med i en solicited-node multicast address för varje unicast och anycastadress på alla interface oavsett hur de har konfigurerats. Trafik till en solicited-node adress är som ett semi-riktad broadcast meddelande som bara ska till en liten samling av noder, förhoppningsbara en.

ICMP (Hitta grannar!)

IPv6 använder sig av olika ICMP-typer för att hitta grannar, annonsera sig själv och lite övrigt smått och gott. Validering sker av paketen. IPv6 noder kommer att slänga RA eller RS paket som har en längre hop limit (TTL) än 255 då det skulle innebära att paketet har routats och är ej anslutet på samma LAN.

Neighbor Solicitation (NS). Använder ICMP typ 135. Destinationen är solicited-node multicatadressen för en specifik host på LANet. Source-adressen är link-local adressen från sändande interface. Används för att hitta grannar på LANet och går att jämföra med ARP request i IPv4. NS kan också ha en unicast destination när NS används för att verifiera nåbarheten efter att en granne har hittats, det kallas för Neighbor Unreachability Detection (NUD). Svar på NUD verifiera tvåvägskommunikation.

Neighbor Advertisement (NA). Använder ICMP typ 136. NA är ett svar på en NS. Destinationen är link-local adressen till noden som har skickat NA. Source är link-local adressen på utgående interface och paketet innehåller nodens L2-adress. Om en nods L2 adress ändras så skickas en unsolicited NA till all-nodes multicas adressen (FF02::1) så att grannarnas IPv6 neighbor table uppdateras. Det finns en flagga i NA-paket som heter solicit-flag. Den är satt till 1 (true) när en NA är ett svar på NS. Vid unsolicited NA är den satt till 0.

Router Solicitation (RS). Använder ICMP typ 133. Skickas från hosts för att hitta tillgänliga routrar på ett segment. Source är link-local adressen på utgående interface och destinationen är all-routers multicast adressen (FF02::2).

Router Advertisement (RA). Använder ICMP typ 134. Skickas periodvis av routrar till all-hosts (FF02::1) med interfacets link-local adress som source. RAs inkluderar vanligtvis en eller fler prefix för SLAAC (64 bitar långt), prefix lifetime, hop limit, MTU och autkonfigurationsdetaljer. Interface på Cisco-enheter kommer automatiskt att skicka RAs på Ethernet och FDDI interface när IPv6 aktiveras på interfacet men beteendet går att ändra konfigurera. Innehåller flaggor, ex. M(managed) och O (other config), se mer information i SLAAC-delen längre ned.

Neighbor Redirect (NR). Använder ICMp typ 137. Används för att meddela en host om en bättre väg till destinationen. Samma syfte som IPv4 ICMP redirect men en NR måste veta link-local adressen till den önskade destinationen. Link-local adressen till destinationen finns med i NR paketet. Om den är känd inkluderas även L2-adressen i NR paketet.

Duplicate Address Detection (DAD). När en ny adress tilldelas en länk kommer DAD att köras för att undvika adressdubletter på LANet. Ett NS-paket skickas med ospecifiserad source adress till all-nodes (FF02::1) för att se om någon använder adressen IPv6 noden har tänkt nyttja. Skulle ett svar komma med NA kommer adressen inte användas. Kommer inget svar kommer adressen att användas. Cisco utför ingen DAD på globala eller anycastadresser som är skapade automatiskt enligt EUI-64, då det antas att dessa redan är unika. DAD kan stängas av per interface med `ipv6 nd dad attempts 0`, det gäller samtliga prefix på ett interface. DAD kommer då att meddela routern att samtliga adresser är unika utan att kolla efter dubletter.

Default Router Preference (DRP) är teknik för att signalera en routers prioritet i RA-paket. Finns flera routrar på samma segment kommer en IPv6 nod att välja default gateway till den nod med högst prioritet, om flera finns. Konfigureras per interface med `ipv6 nd router-preference { high | medium | low }`.

Neighbor Table

```
! Exempel
catalyst#show ipv6 neighbors vlan 1337
Interface Vlan1337, count 8, static 0, limit 8000, ignored 0
 ND cache expire time is 14400 seconds
IPv6 Address                 Age Link-layer Addr State Interface
2001:DB8:1337:7331::4821         178 9c7b.ef9e.9b7f  STALE Vl1337
FE80::50F0:CE17:87C2:2119          8 9c7b.ef9e.9b7f  STALE Vl51337
```

När grannar har hittas så placeras de i IPv6 neighbor-tabellen. När de först är nåbara så markeras de som REACH. Därefter flyttas de till STALE efter 30 sekunder när ingen trafik ska till den noden från routern. Per default så är ett adressentry stale i 4 timmar och tas sedan bort, så länge den inte har flyttats till REACH under tiden.

Allt går att konfigurera. Ex. hur länge ett entry anses vara REACH går att konfigurera per interface med `ipv6 nd reachable-time x`, där x är millisekunder. Hur lång tid det tas tills ett entry i STALE flyttas till DELETE går att sätta globalt med `ipv6 nd cache expire x`, där x är sekunder.

Grannar i status GLEAN syns per default inte i show ipv6 neighbors. De som är i GLEAN är noder som har skickat unsolicited NA. Routrar ignorerar dessa för att spara på minne. Det går dock att spara ned dessa genom att aktivera `ipv6 nd na glean` på interfacenivå.

När ett internt routingprotokoll (IGP går upp och prefix installeras med next-hop till grannens link-local adress kommer NUD automatiskt att göra en ND för link-local adressen, även om ingen trafik går den vägen. Det betende går att stänga av med `no ipv6 nd nud igp`.

När NUD körs så är ett entry i cachen i status PROBE till ett NA svar har kommit.

En granne kan vara i DELAY. Det innebär att en re-resolution för grannen pågår, men trafik till den bör fortfarande fungra.

Router Advertisements

Det går att modifiera beteendet för RAs. Med interface-kommandot `ipv6 nd ra suppress` så kommer en router inte att skicka några RAs på eget bevåg. Den kommer dock att svara på RS med RA. Vill man inte att den svarar på RS så lägg till `all` på slutet.

Om man har flera prefix på ett interface kan man välja att inte annonsera vissa av dem. Det gör man med interface-kommandot `ipv6 nd prefix 2001:DB8:1337::/64 no-advertise`. Hur ofta som RA annonseras och hur lång livstid RAt har gåt att konfigurera per interface med `ipv6 nd ra lifetime x` och `ipv6 nd ra interval x`, där x är sekunder.

DHCPv6

Går att konfigurera för relay och med lokal tilldelning.

Exempelkonfiguration relay:

```
interface Vlan1337
 ipv6 address 2001:DB8:1337:1::1/64
 ipv6 enable
 ipv6 mtu 1500
 ipv6 nd prefix 2001:DB8:1337:1::/64 43200 43200 no-autoconfig
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
```

```
 ipv6 nd ra interval 4 3
 no ipv6 redirects
 no ipv6 unreachables
 ipv6 dhcp relay destination 2001:DB8:1337:2::1337
 ipv6 verify unicast source reachable-via rx
```

Exempelkonfiguration lokal DHCPv6 server på routern:

```
ipv6 dhcp pool DHCPv6_POOL_VLAN1337
 address prefix 2001:DB8:1337:1::/64
 dns-server 2001:DB8:1337:2::13
 dns-server 2001:DB8:1337:2::37
 domain-name jehrlander.net

interface Vlan1337
 ipv6 address 2001:DB8:1337:1::1/64
 ipv6 mtu 1500
 ipv6 nd prefix 2001:DB8:1337:1::/64 43200 43200 no-autoconfig
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 nd ra interval 4 3
 no ipv6 redirects
 no ipv6 unreachables
 ipv6 dhcp server DHCPv6_POOL_VLAN1337 rapid-commit
```

Tillägget rapid-commit tillåter en klient att börja använda adressen efter 2 utbytta DHCPv6 meddelanden, Solicit och Reply. Utan rapid-commit krävs hela processen, Solicit, Advertise, Request, Reply.

Stateless Address Auto Configuration (SLAAC)

SLAAC används för IPv6-tilldelning utan DHCPv6. Klienter använder sig av SLAAC när vissa flaggor i Router Advertisement meddelandet är satta.

Flagga	Värde	Förklaring

M (Managed address configuration)	1 eller 0	Flaggan indikerar om det finns en DHCPv6 server tillgänglig när den är satt till 1.
O (other configuration)	1 eller 0	Indikerar att mer information finns tillgänglig via DHCPv6 server, ex. för att berätta för klienten vilken DNS-server som ska användas.
Options	x	Olika typer av värden kan finnas här. För SLAAC ska *Prefix Information Option* finnas tillgänglig.

Konfen är väldigt liten för att SLAAC-användning ska signaleras. `ipv6 nd managed-config-flag` får ej vara konfigurerat på interfacet för det signalerar till klienten att enbart använda DHCPv6.

Exempelkonf:

```
interface Vlan1337
 description SLAAC
 ipv6 address 2001:DB8:0:5::1/64
 ipv6 enable
 ipv6 mtu 1500
 ipv6 nd prefix 2001:DB8:0:5::/64
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 nd ra interval 4 3
 ipv6 dhcp relay destination 2001:DB8:0:1::10
end

c9500#show ipv6 int vlan 1337
Vlan1337 is up, line protocol is up
  IPv6 is enabled, link-local address is x
  ......
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 3 to 4 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is High
  Hosts use stateless autoconfig for addresses.
  Hosts use DHCP to obtain other configuration.
```

Det går även att annonsera en DNS server med SLAAC. Det görs genom Recursive DNS Server (RDNSS).

```
interface Vlan1337
 ipv6 nd ra dns server 2001:DB8:0:1::10

c9500#show ipv6 nd ra dns server
Recursive DNS Server on: Vlan1337
   DNS Server: 2001:DB8:0:1::10 Lifetime: 12 seconds (default)

interface Vlan1337
 ipv6 nd ra dns server 2001:DB8:0:1::10 86400

c9500#show ipv6 nd ra dns server
Recursive DNS Server on: Vlan1337
   DNS Server: 2001:DB8:0:1::10 Lifetime: 86400 seconds (configured)
```

OSPFv2

Open Shortest Path First (v2) är ett routingprotokoll för IPv4. Det är förmodligen världens största interna routingprotokoll (IGP) då det stöds av samtliga stora nätverksleverantörer. OSPFv2 hanterar IPv4, OSPFv3 hanterar IPv6 (och IPv4 hos Cisco), se egen artikel för OSPFv3. OSPFv2 är ett link-state routing protocol, det betyder att en databas för varje länk/nätverk byggs och denna databas delas och ska se likadan ut inom sitt area. Databasen heter link-state database (LSDB). Databasen byts ut mellan OSPFv2 routrar genom link-state advertisements (LSA).

Dijkstras algoritm, uppkallad efter skaparen holländaren Edsger W. Dijkstra, används för att räkna ut kortade vägen till ett nätverk (SPF, shortest path first).

OSPFv2

Database Exchange

OSPFv2 använder sig av 5 olika meddelanden som routrar kan använda för att bygga grannskap och dela med sig av routinginformation.

Router ID

För att en router ska kunna skickas OSPFv2 meddelanden behövs ett router ID (routing identifier, RID). Det är ett 32 bitar långt tecken och ser därmed ut som en IPv4 address. Ett router ID kan vara statiskt konfigurerat med kommandot `router id x.x.x.x` under OSPF processen. Är det inte manuellt konfigurerat tas den högsta IPv4 adressen från Loopback-interface som ej är shutdown. Finns inget loopback interface tas högsta IPv4 adressen från övriga interface.

Finns flera OSPFv2 processer på samma router så kommer de att försöka välja olika router ID:n. Interfacet som router IDt tas från behöver inte vara med i OSPFv2 processen. Interface i Down/down kan väljas som router ID, enbart shutdown som tar bort interface som kandidat. OSPFv2 behöver inte annonsera en rutt till RID. Vilket router ID som används ändras enbart när processen startas om även om ett hårdsätts i konfigurationen. Vid förändring av router ID behöver övriga routrar i samma area räkna om SPF.

OSPF Paket

Paket	Beskrivning

Hello	Används för att hitta grannar, förflytta grannar till 2-Way State och som keepalive
Database Description (DD, DBD)	Används för att dela med sig av LSA Headers under den första topologiutväxlingen så att en router känner till sin grannes LSAer, inklusive versioner
Link-State Request (LSR)	Ett paket som identifierar en eller flera LSAer som en router önskar veta från sin granne
Link-State Update (LSU)	Svar på LSR från granne med full information om begärda LSAer. Skickas även vid topologiförändringar
Link-State Acknowledgement (LSAck)	Skickas som ack efter mottagen LSU

image.png

Kommandot show ip ospf neighbor [details] visar vilket läge nuvarande grannar är i. Se exempeloutput:

```
catalyst#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.2.3.4         1   FULL/BDR        00:00:02    2.3.4.5   Vlan1337

catalyst#show ip ospf neighbor 1.2.3.4 detail
 Neighbor 1.2.3.4, interface address 2.3.4.5, interface-id 145
    In the area 0 via interface Vlan1337
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 1.3.3.7 BDR is  2.3.4.5
    Options is 0x12 in Hello (E-bit, L-bit)
    Options is 0x52 in DBD (E-bit, L-bit, O-bit)
    LLS Options is 0x1 (LR)
    Dead timer due in 00:00:02
    Neighbor is up for 25w6d
    Index 1/2/6, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
 Neighbor 1.2.3.4, interface address 192.168.32.15, interface-id 146
    In the area 0 via interface Vlan1338
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 1.3.3.7 BDR is 3.4.5.6
    Options is 0x12 in Hello (E-bit, L-bit)
    Options is 0x52 in DBD (E-bit, L-bit, O-bit)
```

LLS Options is 0x1 (LR)

Dead timer due in 00:00:02

Neighbor is up for 4d18h

Index 1/5/10, retransmission queue length 2, number of retransmission 0

First 0x0(0)/0x0(0)/0x7F8D50F91D10(2024369) Next 0x0(0)/0x0(0)/0x7F8D50F91D10(2024369)

Last retransmission scan length is 0, maximum is 0

Last retransmission scan time is 0 msec, maximum is 0 msec

Link State retransmission due in 4588 msec

Neighbor States

OSPF grannar är olika lägen beroende på om ett grannskap har lyckats bildas, är påväg att bildas eller har misslyckats att bildas.

Grannläge	**Förklaring**
Down	Grannskap nere. Syns oftast när ett tidigare bildat grannskap har gått ned.
Attempt	Finns enbart på NBMA och P2MP interface. Grannar placeras direkt i attempt på dessa interface och Hello-paket skickas. Flyttas till Down om granne ej svarar.
Init	Hello paket har tagits emot men Hello-paket innehåller inte mottagande routers RID. Mottagande router kan alltså höra skickande router men är inte säker på tvåvägskommunikation.
2-Way	Hello paket har mottagits från granne och mottagande routers RID finns i Hello-paket. Bekräftar tvåvägskommunikation mellan routrarna. Routrar som inte pratar med varandra, såsom DR/OTHER på multiaccesssegment, kommer att stanna i 2-Way.
ExStart	En granne flyttas från Init eller 2-Way till ExStart när tvåvägskommunkationen är bekräftad. I ExStart vilken router som är Master och vilken som är Slave. Tomma DBD paket bytes ut för att jämföra Router ID, bestämma DR och komma överens om vilket sekvensnummer kommunikationen börjar på för kommande DBD paket i Exchange-läge.
Exchange	Granne flyttas hit från ExStart när Master/Slave har bestämts. DBD paket innehållande alla kända LSAer byts ut mellan routrarna.
Loading	Är i loading efter att LSAer har bytts ut i Exchange men routern behöver tid för att ladda ned LSAer.
Full	Grannskapet flyttas hit från Exchange eller Loading när grannskapet är klart. Databasen är synkad och Hello-paket utbytes för keepalive.

Hello process

Hello-paket fyller flera funktioner i OSPFv2. Hello-paket hittar andra OSPFv2 routrar på gemensamma subnät, de utför checkar på vissa konfigurationsparametrar, de bekräftar tvåvägskommunikation mellan routrarna och de övervakar grannskapet och reagerar när inget svar kommer från granne.

För att hitta grannar skickas multicasttrafik på samtliga OSPFv2 interface som ej är passive till 224.0.0.5. 224.0.0.5 är en reserverad multicastadress till OSPF All Routers. Paketen source IP är interfacets IPv4 adress. Det går att använda unnumbered links, alltså att hänvisa till en IP adress på ex. ett loopback-interface, det går bra.

Vissa kriterier måste matcha för att ett grannskap ska bildas. Dessa är:

1. Autentisering måste matcha
2. Måste vara i samma subnät, inklusive exakt match på subnätsmasken
3. Måste vara i samma OSPFv2 area
4. Måste vara samma areatyp (normal, stub, NSSA)
5. Router ID måste vara unikt
6. OSPFv2 Hello och Dead timers måste matcha

Process ID:t behöver ej matcha då det är lokalt för routern. För att DBD paket ska processas mellan grannar så måste MTU matcha men är tekniskt sett inte en del av Hello-processen. Hello-paket innehåller alla router ID:n från grannar en router har. Hello paketen agerar keepalive. Hellos skickas varje Hello-intervall och om ett Hello-paket ej tas emot inom definierad Dead interval så rivs grannskapet. Default-intervall för Hello paket är 10 sekunder på broadcast interface och 30 sekunder på nonbreadcoast och point-to-multipoint interface. Dead interval är per default 4 gånger så lång som helllo intervallen.

Det går att se timers genom kommandot show ip ospf interface x, se exempeloutput:

```
catalyst#show ip ospf interface vlan 1337
Vlan1337 is up, line protocol is up
  Internet Address 1.2.3.5/31, Interface ID 181, Area 0
  Attached via Interface Enable
  Process ID 1, Router ID 1.3.3.7, Network Type BROADCAST, Cost: 1048
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0          1048      no          no          Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.3.3.7, Interface address 1.2.3.5
  Backup Designated router (ID) 1.2.3.4, Interface address 1.2.3.4
  Timer intervals configured, Hello 1, Dead 3, Wait 3, Retransmit 5
    oob-resync timeout 40
```

```
  Hello due in 00:00:00
Supports Link-local Signaling (LLS)
Cisco NSF helper support enabled
IETF NSF helper support enabled
Can be protected by per-prefix Loop-Free FastReroute
Can be used for per-prefix Loop-Free FastReroute repair paths
Not Protected by per-prefix TI-LFA
Index 1/33/37, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 42
Last flood scan time is 0 msec, maximum is 1 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 1.2.3.4  (Backup Designated Router)
Suppress hello for 0 neighbor(s)
Cryptographic authentication enabled
  Youngest key id is 1
```

När två routrar utbyter Hello-paket och parametrarna stämmer kommer inte hela LSA databasen att bytas ut direkt. Först skickas DBD som innehåller rubriker över alla LSAer, en indexerad lista över alla LSAer en router känner till. Därefter så ber routrarna om de LSAer från varandra som de lokalt inte känner till. Varje DBD paket har ett sekvensnummer. Varje DBD paket ackas genom att samma sekvensnummer skickas tillbaka (LSAck). En LSAck behövs för att sändande router ska gå vidare till nästa LSA.

Master/Slave

Dessa roller väljs under ExStart. Rollern dikterar vilket ansvar routern har under utbytet av DBD paket. Enbart master får skicka paket på eget initiatv och öka sekvensnummer. Slave får enbart skicka DBD paket som svar på DBD paket från master. Det blir en polling från master till slave.

DBD paketet innehåller bland annat följande flaggor:

- Master (MS) flag, är flaggad i samtliga paket skickad från master och är ej flaggad i paket från slave
- More (M) flag, flaggas när en router tänker skicka fler DBD paket efter flaggat paket
- Init (I) flag, indikerar att det här är det första DBD paketet från master eller slave, följande paket ska ej ha denna flaggad

Skulle master skicka ett DBD paket och be om fler LSAer fast slave inte har några oskickade så svarar slave med ett tomt DBD paket, detta då slave måste svara på frågor från master. Om en slave har LSAer att dela med sig av som master ej har bett om så kommer slave att sätta M flaggan i sitt DBD-svar till master, då vet master att slave har fler LSAer att dela med sig av. Master slutar att skicka DBD paket till slave när den inte har några fler LSAer att dela med sig av och slaven ej skickade med M-flaggen i sitt senaste DBD paket.

När grannarna placerar varandra i ExStart (efter 2-Way) så anser båda att de är Master och skickar ett tomt DBD paket till varandra med ett slumpat sekvensnumret, MS flaggan, M flaggan och I flaggan. Efter mottagning av varandras DBD paket så ställer den router med lägre RID om sig till slave och svarar med ett DBD paket utan MS och I flaggan och sekvensnumret från routern med högre RID. Master skickar sedan DBD paket med sekvensnumret ökat med 1 och påbörjar utbytet av LSAer.

Requesting, Getting and Acknowledging LSas

Efter utbytet av LSA headers i början av utbytet så känner routrarna till vad sin granne känner till och ber om kompletterande LSAer för det den inte har i sin egen LSDB. För att se vilken router som har mest uppdaterad information jämfört sekvensnumret för en LSA i LSDB mot sekvensnumret för samma LSA i DBD paketet. Varje gång en LSA ändras på så ökar sekvensnumret. Sekvensnumret finns i headern för LSAerna, så om en router ser att en granne har ett högre sekvensnummer för en LSA kommer den att begära att få den LSAn skickad.

Link-State Requests (LSR) används för att begära en eller flera LSAer från en granne. Svar kommer med Link-State Updates. Det här sker i läget Loading. Samtliga LSUer ska ackas med en LSAck eller genom att repetera samma LSU tillbaka.

När LSAerna är synkade och därmed LSDB färdigbyggd kommer routrarna att köra SPF-algoritmen för att hitta snabbaste vägen till nätverket.

Designated Routers

För att LSA synkning på multiaccesssegment, ex. Ethernet, inte ska bli superbökig och kräva fulla utbyten mellan samtliga routrar på ett segment finns Designated Routers (DR). DR och Backup DR (BDR) är de enda routrar på ett segment som har fullt grannskap med andra routrar. Istället för att ex. 10 routrar alla skapar grannskap med varandra (det skulle bli totalt 45 grannskap) så sker full synkning enbart till DR & BDR som i sin tur skickar informationen vidare till övriga routrar. En router som inte är DR eller BDR benäms som DROTHER i Cisco outputs.

Grannskapen mot DR och BDR är alltså de enda som tar sig hela vägen till läget Full, sett från en DROTHER. Övriga routrar stannar i 2-Way. Från DR och BDRs synvinkel är samtliga grannskapslägen Full. För att kommunicera mot enbart DR och BDR så finns en broadcastadress, 224.0.0.6 (All OSPF DR Routers), som enbart DR och BDR lyssnar på. DR och BDR skickar sedan vidare mottagen LSA information som LSU till 224.0.0.5 (All OSPF Routers). LSUer som skickas från DROTHER till 224.0.0.6 får ej en LSAck tillbaka från DR/BDR, LSU flooden till 224.0.0.5 räcker som ack. Övriga routrar, inklusive BDR, måste dock Ack:a DR LSU med en unicast LSAck.

Valet av DR och BDR sker vid första grannskap på en länk. Ingen preemption sker, mottages ett Hello-paket med ett populerat DR-värde så accepterar routern denna som sin DR. För att välja om DR, eller BDR, krävs det att grannskap på segmentet bryts på valfritt sätt. Valet av DR/BDR sker när Hello-paket från en granne innehåller en DR med värdet 0.0.0.0, alltså ingen vald, efter en timer som kallas OSPF wait time. OSPF wait time är samma värde som Dead-timern. Detta för att ge eventuella andra routrar på segmentet möjlighet att svara/skicka Hello-paket.

En router kan bli DR/BDR om OSPF prioriteten på interfacet är mellan 1 och 255. Ett värde på 0 innebär att den inte deltar i valet. Varje router väljer sin DR/BDR på egen hand men algoritmen ser till att det blir samma utfall i samtliga routrar. Om en router mottager en Hello-paket med populerat DR/BDR värde under OSPF wait time så kommer routern direkt påbörja DR/BDR valet. Valet sker bara för roller som ej är tillsatta, finns en DR eller en BDR redan så kommer enbart den ena väljas. Skulle två routrar ha samma OSPF priority så vinner högsta RID. Skulle en DR försvinna från segmentet så blir BDR befodrad till DR och ny BDR väljs.

Skulle samma segment innehålla flera routrar som tror sig vara DR/BDR, exempelvis vid länk som har gått ned mellan ett segment av routrar, så kommer regeln om ingen preemnt att skrotas och ett nytt val av DR/BDR sker.

OSPF Network Type

DR/BDR behövs ej på P2P-länkar. På NBMA nät kan DR vara hjälpsamma. Det går att ställa in om DR ska användas genom att ändra OSPF Network Type på interfacet. Beroende på vilken nätverkstyp man väljer så ändras OSPFv2s timers. Hård konfgurerade timers överskrider nätverkstypens default timers.

image.png

Kommandot för att ändra OSPFv2 nätverkstyp i interfacekonfigläge är `ip ospf network TYPE`, där TYPE är något av valen ovan.

Stabil drift

När OSPFv2 har stabiliserat sig och konvergeras, dvs att samtliga routrar inom ett area har exakt samma LSDB och har räknat ut bästa vägarna via SPF, sker följande:

- Routrar skickar Hello-paket baserat på interface timers
- Routrar förväntar sig att ta emot Hello-paket på interface mot grannar och tar ned grannskap om Dead timern räknar slut
- Routrar floodar LSAer genererade lokalt på nytt efter halva sin livslängd (LSA MaxAge). Hela livslängden är default 60 minuter, flooding sker alltså efter 30 minuter
- Routrar förväntar sig att mottagna LSAer ska uppdateras inom LSA MaxAge (default 60 minuter)

OSPF Design, LSAer och Areas

Interface i OSPF gruppas in i areas. Det viktigaste areat är 0. Formatteras enligt industristandard som 0.0.0.0, i IOS-XE kan man skriva det på båda sätt. Area 0 är känt som backbone area. Samtliga övriga areas måste ansluta sig mot area 0. Area 1 kan inte ha en direktlänk till area 2 utan måste traversera area 0 för att nå area 2. En router som ansluter till flera areas kallas för Area Border

Router (ABR). En router blir ABR när den har interface konfigurerade i flera olika areas inklusive area 0, alltså ex. `ip ospf 1 area 0` på GigabitEthernet0/0 och `ip ospf 1 area 1` på GigabitEthernet0/1, och interfacen är uppe. En router som är ansluten till area 1 och area 2, men inte area 0, anser sig inte vara en ABR.

En router som injicerar rötter in i OSPF via redistribuering kallas för Autonomous System Boundary Router (ASBR).

image.png

En OSPFv2 router har en LSDB per area den är ansluten mot. Innehållet i de olika LSDBerna är isolerade, men en ABR kommer att översätta innehållet kontrollerade mellan LSDBerna. SPF-uträkning körs per area.

Det finns flera fördelar av att använda sig av OSPFv2 areas:

- Mindre LSDBer vid styckning av areas. Routrar i mindre areas behöver inte vara lika fläskiga som routrar i större areas
- Snabbare SPF uträkningar då LSDBn är mindre
- Om en länk går ned behöver SPF inbart köras av routrar inom länkens area
- Det är enbart av ABRer (och ASBRer) som summering och filtrering av rötter kan ske i OSPF

Mellan areas skickas inte Typ 1 eller Typ 2 LSAer. Istället skickas Typ 3 LSAer (Summary LSA) som är mindre detaljerade än en typ 1 eller typ 2. Därmed blir även LSDB storleken mindre med användning av areas, även för rötter som läcks mellan areas.

Vägval

OSPF föredrar alltid en intra-area router, alltså en route som kommer från samma area routern är i, över en inter-area route. Det gäller även om metricen för inter-area routen är bättre än metric för intra-area routen. En ABR kommer ej att processa en typ 3 LSA (summary) om den är mottagen via ett area som ej är backbone. Det är en loopförhindrade mekanism, Split Horizon!

Hackordning är enligt följande (där 1 vinner):

1. Intra-area
2. Inter-area
3. E1/N1
4. E2/N2

Den här hackordningen vinner alltid över cost. Så även om en E2 router är tusen gånger bättre än en Inter-area kommer Inter-area att vinna.

LSA Typer

En LSA kan enbart modifieras av den som skapade LSAn. Övriga routrar har som skyldighet att processa och sprida vidare LSAn enligt sin roll.

LSA Typ	Namn	Beskrivning
1	Router	1 per router i ett area. Innehåller routerns RID och alla dess interface inom det area. Skickas enbart inom ett area.
2	Network	Skapas av DR om sådan finns. Innehåller det subnätet och interface på det subnätet. Skickas enbart inom ett area.
3	Net Summary	Skapas av ABR när typ 1 anländer via ett area och ska annonseras till ett annat area. Definierar subnäten och cost men ingen topologi. Floodas inom ett area.
4	ASBR Summary	Som en typ 3 men annonseras host route för att nå ASBR i ett annat area.
5	AS External	Skapas av ASBR för rötter som injiceras i OSPF. Floodas till alla vanliga areas.
6	Group Membership	Används ej av Cisco IOS
7	NSSA External	Som en typ 5 men från en ASBR i NSSA. Konverteras av typ 5 av ABR.
9-11	Opaque	För framtiden! 9 = link-local flooding, 10 are local flooding, 11 AS flooding som typ 5

Två termer som ska kännas till är Transit network, vilket är ett nätverk som en eller flera OSPFv2 routrar har blivit grannar över, och Stub network, vilket är ett subnät som en OSPFv2 router ej har grannar via.

LSA Typ 1 och 2

Varje router skapar och skickar ut en typ 1 LSA för sig själv i varje area den har interface inom. Typ 1 LSAer beskriver routern, OSPFv2 interfacen inom ett area och en lista över grannar per interface. LSAn identifieras av en link-state ID (LSID) vilket är samma värde som RID.

Typ 2 beskriver ett transit subnet som en DR har blivit vald på. Typ 2 skapas enbart av DR. LSIDn är DRns interface IP address på det subnätet.

Med hjälp av Typ 1 och Typ 2 LSAer kan SPF algortimen räkna ut bästa vägen till ett nätverk inom ett area.

LSAer kan kontrolleras med show ip ospf database. För att se specifikt typ 1 och typ 2 finns show ip ospf database router och show ip ospf database network, se exempel här:

```
catalyst#show ip ospf database router

            OSPF Router with ID (1.3.3.7) (Process ID 1)

□□Router Link States (Area 0)

  LS age: 1583
  Options: (No TOS-capability, No DC)
  LS Type: Router Links
  Link State ID: 1.2.3.4
  Advertising Router: 1.2.3.4
  LS Seq Number: 8000358F
  Checksum: 0x8A41
  Length: 84
  Area Border Router
  AS Boundary Router
  Number of Links: 5

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 2.3.4.5
     (Link Data) Router Interface address: 3.4.5.6
      Number of MTID metrics: 0
       TOS 0 Metrics: 1000

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.0.0.0
     (Link Data) Network Mask: 255.255.255.240
      Number of MTID metrics: 0
       TOS 0 Metrics: 10

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.0.1.0
     (Link Data) Network Mask: 255.255.255.240
      Number of MTID metrics: 0
       TOS 0 Metrics: 10

    Link connected to: a Stub Network
```

```
    (Link ID) Network/subnet number: 10.0.2.0
    (Link Data) Network Mask: 255.255.255.240
     Number of MTID metrics: 0
      TOS 0 Metrics: 10

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 4.5.6.7
    (Link Data) Router Interface address: 1.2.3.4
     Number of MTID metrics: 0
      TOS 0 Metrics: 10

catalyst#show ip ospf database network

          OSPF Router with ID (1.3.3.7) (Process ID 1)

  Net Link States (Area 0)

  LS age: 1475
  Options: (No TOS-capability, DC)
  LS Type: Network Links
  Link State ID: 1.3.3.8 (address of Designated Router)
  Advertising Router: 1.3.3.7
  LS Seq Number: 80000D3F
  Checksum: 0xBC10
  Length: 32
  Network Mask: /28
  Attached Router: 1.3.3.7
  Attached Router: 1.3.3.8
```

För att signalera att ett nätverk har gått ned (ex. interface går ned) så skapar en routar en ny LSA som exkluderar nätverket, alternativt skickas en LSA med där LSA MaxAge har gått ut vilket gör att routrar som mottar den nya LSAn direkt informationen om nätverket.

LSA Typ 3

En ABR kommer aldrig att skicka vidare en Typ 1 eller Typ 2 LSA från ett area till ett annat. Istället så skapar den en Typ 3 som representerar de Typ 1 eller Typ 2or som ABRen känner till. En ABR kommer enbart att hantera en inkommande Typ 3 som är mottagen via Area 0. När en ABR skapar en typ 3 så kommer enbart intra-area rötter från icke backbone att skickas till backbone som Typ 3. I motsatt riktning, från backbone till annat area, kommer både inter-area och intra-area rötter att skickas vidare som Typ 3.

```
catalyst#show ip ospf database summary

            OSPF Router with ID (1.3.3.7) (Process ID 1)

  Summary Net Link States (Area 0)

  LS age: 1033
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.1.3.0 (summary Network Number)
  Advertising Router: 1.3.3.7
  LS Seq Number: 8000039D
  Checksum: 0x6536
  Length: 28
  Network Mask: /24
  MTID: 0  Metric: 66583

! Cost för att nå en border router kan ses med nedan kommando

catalyst#show ip ospf border-routers

            OSPF Router with ID (1.3.3.7) (Process ID 1)


  Base Topology (MTID 0)

Internal Router Routing Table
Codes: i - Intra-area route, I - Inter-area route

i 9.9.9.9 [3096] via 2.3.4.5, Vlan1337, ABR/ASBR, Area 0, SPF 259

sw-distacc#show ip ospf statistics

            OSPF Router with ID (1.3.3.7) (Process ID 1)

  Area 0: SPF algorithm executed 702 times

  Area 1: SPF algorithm executed 114 times

  Area 3: SPF algorithm executed 15974 times
```

```
  Area 7: SPF algorithm executed 763 times

  Summary OSPF SPF statistic

  SPF calculation time
Delta T   Intra D-Intra Summ D-Summ Ext D-Ext Total Reason
17:30:52   0 1 0 0 0 1 2 R, X
12:46:29   0 0 0 0 0 1 2 R, X
12:45:17   0 0 0 0 1 0 1 R
12:45:16   0 0 1 0 0 1 2 R, X
05:30:33   0 0 0 0 1 0 1 R, X
05:29:29   0 0 0 0 1 0 1 R
05:29:29   0 0 0 0 1 0 1 R, X
00:45:05   0 0 0 0 1 0 1 R, X
00:43:45   1 0 0 0 0 1 2 R
00:43:44   0 0 0 0 1 0 1 R, X
```

En bra nyans att känna till är att trafik mellan areas måste flöda genom en router som är ansluten till backbone. Det betyder dock inte att trafiken behöver flöda via ett interface i backbone. Om en ABR är ansluten till area 0, area 1 och area 2 så kommer trafiken att flöda direkt mellan area 1 och 2 i det fallet.

Typ 4 och 5 SLAer samt External Routes Typ 1 och 2

Redistribuerade rötter in i OSPFv2 märks med External Route Type 1 (E1) eller External Route Type 2 (E2). Metrics för E2 förändras inte hop-per-hop utan är samma konstant enligt vad som specifiseras i redistribueringen. Metrics för E1 ändras per hop, precis som för övriga LSAer. En E1 föredras över en E2 vid routingbeslut.

När en extern route injiceras via redistribuering så skapas en typ 5 LSA för subnätet. LSAn innehåller metricen och metrictypen (E1/E2). LSAn skickas till samtliga OSPFv2-grannar. När en LSA typ 5 traverserar areas så skapar ABR en LSA typ 4 som skickas ut i samband med typ 5. Detta då LSA typ 5 innehåller information om vilken ASBR som nätet finns via vilket det andra areat inte känner till via de typ 3 som har skickats in i areat. LSA typ 4 innehåller information om vilken ABR areat ska använda för att nå nätverket annonserat i LSA Typ 5.

```
! Exempel på typ 5:
catalyst#show ip ospf database summary

        OSPF Router with ID (13.3.7) (Process ID 1)
```

Summary Net Link States (Area 1337)

LS age: 1856

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 10.0.0.0 (summary Network Number)

Advertising Router: 1.2.3.4

LS Seq Number: 80000D68

Checksum: 0xFDAC

Length: 28

Network Mask: /21

MTID: 0 Metric: 2106

Stubby Areas

Mellan vanliga areas så sker ingen filtrering. Teknikerna beskrivna ovan begränsar hur ofta och omfattande SPF måste räknas om vid topologiförändringar, men visibileten mellan area 0 och övriga "normala" areas är oförändrad. För att ändra vilka rötter som propageras via LSAer mellan areas behöver man använda sig av olika areatyper. Det är enda sättet man kan filtrera och summera rötter i OSPF. Dessa varianter kallas för stubby areas.

Om ett area konfigureras som stub så kommer ABRer ej att annonsera LSA typ 4 eller LSA typ 5 in i areat. Externa rötter kommer alltså inte propageras in i dit. Tanken är att dessa ska täckas av en default-route, som per default kommer annonseras in i areat via ABR. Skulle en typ 5 mottas så kommer en stub-router att strunta i paketet och den kommer aldrig själv att skapa en typ 5.

Beroende på vilken typ av stubby-area man nyttjar kan även Typ 3 ej annonseras in i areat, vilket gör att areat enbart får en default-route injiceras. Det går även att använda sig av Not-so-stubby areas för att tillåta att externa routes injicera från inuti stub-areat. Dessa får en LSA Typ 7 inom areat. Typ 7 översätts till typ 5 av ABR. Se tabell:

Areatyp	**Stoppar typ 4/5?**	**Stoppar typ 3?**	**Tilllåter skapande av typ 7?**	**Konfiguration**
Stubby	Ja	Nej	Nej	area x stub
Totally stubby (S)	Ja	Ja	Nej	area x stub no-summary
Not-so-stubby area (NSSA)	Ja	Nej	Ja	area x nssa
Totally NSSA (NSSA-TS)	Ja	Ja	Ja	area x nssa no-summary

Jag tycker personligen om att använda av NSSA. Möjligheten att summera och filtrera samtidigt som flexibiltet finns för att kunna injicera ex. redistribuerade statiska rötter är väldigt trevlig. Om en default route ska annonseras i ett NSSA (ej NSSA-TS) så måste det manuellt konfigureras med `area x nssa default-information-originate`.

För att ett NSSA ska acceptera en default-route från en grannrouter i samband med VRF:er krävs kommandot `capability vrf-lite`. Se ex:

```
router ospf 4 vrf veeerf
 capability vrf-lite
```

Exempelkonfiguration och diverse förklaringar

```
interface Vlan2
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 AABBCCDDEEFF
 ip ospf dead-interval 3
 ip ospf hello-interval 1
 ip ospf 1 area 0
 ip ospf priority 100

router ospf 1
 router-id 1.3.37
 auto-cost reference-bandwidth 1048576
 area 0 filter-list prefix FILTER out
 area 2 nssa
 area 2 range 10.100.0.0 255.255.0.0
 area 2 nssa default-information-originate no-summary
 redistribute static route-map ROUTE_MAP_OSPF
 passive-interface default
 no passive-interface Vlan2
 distribute-list prefix OSPF_IN in
 default-information originate

router ospf 2 vrf ospf-vrf
 router-id 3.4.5.6
```

```
 auto-cost reference-bandwidth 1048576
 nsf ietf restart-interval 140
 capability vrf-lite
 area 10 nssa
 passive-interface default
 no passive-interface Vlan3
!
```

Interface

Det går att aktivera OSPFv2 för ett interface på två sätt. Det ena är som i exemplet ovan, att aktivera det direkt på interfacet. Det andra är att matcha den IP adress som finns på interfacet i OSPFv2 processen. Ex. har Vlan2 IPv4 10.10.10.10 kan interfacet läggas till i OSPFv2 processen genom kommandot network `10.10.10.10 0.0.0.0 area 0`. Nollorna är wildcard, alltså matchar 4 nollor enbart det här interfacet. Aktiveras OSPFv2 via interfacet, som i exemplet ovan, kommer secondary addresses även att annonseras som stubnätverk. Det går att exkludera secondaries genom att lägga till `secondaries none` på slutet av kommandot.

För att aktivera autentisering måste både en autentiseringsmetod och själva nyckeln specifiseras. Den konfiguration som är möjlig att göra i OSPFv2 processen är enbart att sätta default-autentisering för interface i areat. Inga lösenord kan konfigureras där. Kommandona i OSPFV2 processen är `area x authentication` för okrypterad text och `area x authentication message-digest` för MD5 kryptering.

Prioriten är för valet av Designated Router. Hög prioritet är bättre än låg. Default är 1.

OSPFv2 Process

`auto-cost reference bandwidth` ändrar default-kostnaden för interface. Värdet bör matcha på samtliga routrar i OSPFv2-domänen. Cost går även att sätta manuellt på interface.

`area 0 filter-list prefix FILTER out` filtrerar baserat på prefix lista vilka rötter som får lämna area 0 och installeras i övrigt anslutna areas på ABR. Det går även att konfigurera i andra riktningen, in, alltså vilka rötter som får installeras in i ett area.

`distribute-list prefix OSPF_IN in` filtrerar baserat på prefix lista vilka rötter som får installeras från LSDB till RIB. LSDB modifieras ej. Det går även att specifisera distribute-lists per interface.

`area 2 range 10.100.0.0 255.255.0.0` är ett kommando som summerar vilka rötter som finns inom ett area. I det här fallet kommer area 0 enbart känna till /16 masken i area 2 och inte samtliga specifika prefix. Det går även att lägga till `not-advertise` efteråt vilket gör att varken de mindre prefixen eller summeringen annonseras.

Virtual Link

Vid händelse att en OSPFv2 domän är helt trasig eller felbyggd och direktanslutning från area 0 till ett annat area ej är möjligt finns virtual links. De kan ej byggas någon form av stub-area. Det gör att en router i exempelvis area 5 tror sig vara direktansluten till area 0 även om trafiken går genom area 10. Det är inte ett tunnelprotokoll utan paketen routas som vanligt. Konfigurationen är enkel, i OSPFv2 processen konfigureras `area 10 virtual-link 10.20.30.40`. Det som specifiseras är transit-areat och destinationsroutern router ID.

Det går att nå samma resultat med GRE-tunnlar, även här över stub-areas. Dock så behöver man då tänka på att det är en tunnel och minst 24 bytes måste skyfflas undan åt GRE. Rekommendationen (riktiga rekommendationen är bygg inte OSPFv2 såhär...) är att använda virtual-links om man verkligen måste. GRE om man är utan val.

Autentisering är möjlig för virtual-links. Tillgängliga kommandon är `area x virtual-link 1.2.3.4 authentication null`, `area x virtual-link 1.2.3.4 authentication authentication-key x` och `area x virtual-link 1.2.3.4 authentication message-digest message-digest-key x md5 x`.

Autentisering med SHA-HMAC

Olika autentiseringsmetoder beskrivs ovan. IOS stöd för autentisering med SHA-HMAC med hjälp av key-chains. Det gäller även virtual-links. Send-liftime och accept-life-time går att specifisera för att byta ut nycklar periodvis. Nyckeln med högst key ID används om flera val finns. För skickade paket signerar routern paketen med giltig nyckel med högst key ID. För mottagna paket matchas det key ID som anges i mottagna paketet. Det går att kombinera en key-chian med MD5 kryptering med klassisk OSPFv2 MD5 autentiseringssyntax.

Exempelkonfiguration:

```
key chain OSPF
 key 1
   cryptographic-algorithm hmac-sha-256
    key-string p4ssw0rd1337

interface GigabitEthernet0/1
 ip ospf authentication key-chain OSPF

router ospf 1
 area 10 virtual-link 10.20.30.40 key-chain OSPF
```

TTL Security

OSPFv2 har stöd för TTL-security. Samtliga pakets TTL sätts då till 255 när de skickas. Om ett paket mottags med mindre TTL än 255 så slängs det, då har paketet routats på vägen och är inte direktanslutet. Detta för att skydda mot att någon illasinnat skapar ett OSPFv2 paket som skickas via unicast. Kan användas som överbelastningsattack mot ens routers CPU. TTL Security kan konfigurerasper interface med kommandot `ip ospf ttl-security` eller i en OSPFv2 process med kommandot `ttl-security all-interfaces`. Om det konfigureras per OSPFv2 process kan det stängas av per interface med `ip ospf ttl-security disable`. Det går att lägga till `hops x` på slutet av kommandot för att tillåta lägre TTL-värde. `ip ospf ttl-security hops 50` skulle ex. tillåta att routern processar ett paket med TTL 205.

SPF Throttiling

Det går att ställa in hur ofta SPF processen ska köras för att räkna ut bästa väg til destinationer. Inledningsvis så ska det här aldrig behöva ändras i miljöer av vanlig storlek. Per default så körs SPF 5 sekunder efter mottagen LSA. Kommer en uppdaterad LSA efter den här SPF-körningen så ökar väntetiden till 10 sekunder. Det är för att undvika att köra SPF medans man fortfarande mottager LSAer konstant.

Praramterarna för en SPF körning är spf-start, spf-hold och spf-max-wait. Start parametern indikerar väntetid innan en SPF uträkning när nätverket har varit stabilt under tid. Hold parametern indikerar väntan mellan efterföljande SPF körningar och dublerar sitt värde efter varje efterföljande SPF körning. max-wait parametern definierar den absoluta maxtiden mellan SPF-körningar, så hold parametern kan aldrig överstiga max-wait. Den definierar också hur länge ett nätverk ska vara stabilt för att flyttas tillbaka till värdet i spf-start.

Se följande för demonstrering:

1. LSA mottages sekund 0. Routern planerar SPF att köras om 10 sekunder.
2. Ny LSA mottages sekund 2. LSA sparas i LSDB och kommer att inkluderas i SPF-körningen.
3. Sekund 10 körs SPF och spf-hold värdet sätts till 15 sekunder. Om en LSA mottages inom dessa 15 sekunder så kommer nästa SPF körning köras sekund 25. Nätverket anses vara stabilt enligt max-wait värdet efter 100 sekunder efter SPF körning, alltså sekund 110.
4. En eller flera LSAer mottags mellan sekund 10 och sekund 25.
5. SPF körning körs sekund 25. spf-hold värdet dubbleras till 30 sekunder, alltså kommer nästa körning för mottagna LSAer ske sekund 55. Nätverket anses vara stabilt enligt max-wait sekund 125.
6. Inga LSAer tas emot kommando 30 sekunder. Väntetiden för SPF körning sätts tillbaka till spf-start värdet, alltså 10 sekunder. Nätverket anses inte stabilt än då sekund 125 ej har nåtts och spf-hold värdet är fortfarande 30 sekunder.
7. Sekund 80 tas en LSA emot. SPF uträkning planeras till sekund 90.
8. Sekund 90 körs SPF. Väntetiden till nästa SPF körning blir dubblerad från tidigare, alltså nu 60 sekunder. Om nya LSAer tas emot de kommande 60 sekunderna kommer SPF körning att ske sekund 150. Nätverket anses stabilt om 100 sekunder, sekund 190.
9. Inga LSAer tas emot de kommande 60 sekunderna. Vänteintervallen sätts tillbaka till spf-start, 10 sekunder, spf-hold är fortsatt 60 sekunder.

10. Inga nya LSAer har mottagits till sekund 190. Nätverket anses vara stabilt och spf-hold är åter 15 sekunder.

SPF throttling ställs in i OSPF-processen med kommandot `timers throttle spf spf-start spf-hold spf-max-wait`, värdena ska såklart bytas ut. Vilka värden som används kan ses med `show ip ospf | inc SPF`.

```
catalyst#show ip ospf | inc SPF
 Initial SPF schedule delay 50 msecs
 Minimum hold time between two consecutive SPFs 200 msecs
 Maximum wait time between two consecutive SPFs 5000 msecs
```

LSA Throttling

Precis som ovan är det här nånting man helst inte skruvar på. Det ska finnas goda skäl för att ändra på de fördefinierade värdena.

Med LSA Throttling kan man ställa in hur ofta en LSA skapas på nytt från en router. LSA Throttling använder sig av ungefära samma värden som SPF Throttling, `start-interval`, `hold-interval` och `max-interval`. När en LSA inte har uppdaterats fler gånger än max-interval värdet säger och den behöver skapas på nytt så återskapas LSAn och skickas till grannar aefter start-interval värdet. Väntetid sätts enligt värdet i hold-interval. Om samma LSA behöver uppdaterats på nytt inom väntetiden så väntar routern med det tills att väntetiden enligt hold-interval har gått ut. Hold-interval dubbleras varje gång LSAn behöver uppdateras inom väntetiden. Väntetiden kommer aldrig att överstiga värdet i max-interval. Om LSAn ej behöver uppdateras när väntetiden har gått ut kommer väntetiden att sättas enligt värdet i start-interval men hold-interval är fortfarande kvar på det högre värdet. Behöver LSAn skickas på nytt under väntetiden, men innan max-interval värdet har nåtts, kommer nästa väntetid att bli enligt det höga hold-interval värdet, fast dubblerat såklart. Verkligen samma beteende som i SPF Throttling.

Defaultbeteende i Cisco IOS är att en uppdaterad LSA skickas direkt och en till uppdatering av denna väntar 5 sekunder. Start-interval blir alltså 0, hold-interval är 5000 millisekunder och max-interval är 5000 millisekunder. Alltså är maxtiden en LSA behöver vänta på att skickas 5 sekunder. LSA throttling konfigureras i OSPFv2 processen med `timers throttle lsa all start-interval hold-interval max-interval`, med utbytta värden såklart. Routerns värden går att se med `show ip ospf | inc LSA`.

```
catalyst#show ip ospf | inc LSA
 Initial LSA throttle delay 50 msecs
 Minimum hold time for LSA throttle 200 msecs
 Maximum wait time for LSA throttle 5000 msecs
 Minimum LSA arrival 100 msecs
 LSA group pacing timer 240 secs
```

Det går även att ställa in så att en router ignorerar LSAer som mottags för tätt intill varandra genom att justera LSA arrival värdet med kommandot `timers lsa arrival milliseconds` i OSPFv2 processen. Om två eller flera likadan LSAer mottags millisekunder mellan varandra så accepteras den första och den andra slängs.

Incrimental SPF

ISPF är att sätt att låta SPF enbart räkna om de delar av topologin som förändras vid LSDB-uppdateirng. Det kan minska CPU-lasten och snabba på konvergenstiden. Kan vara bra att använda sig av i större topologier. Aktiveras med kommandot `ispf` i OSPFv2 processen.

```
catalyst#show ip ospf | inc Incremental
 Incremental-SPF disabled
```

OSPFv2 Prefix Suppression

I stora nät kan en stor del av innehållet i LSDB vara länknät (transit networks). Det kan vara helt ointressant för en router i Kiruna att känna till ett länknät i Kapstaden när det egentligen är är stub network som är intressant, alltså subnätet som hostar siter anslutna mot. RFC 6860 definierar en metod för att dölja eller bli väck med dessa prefix från LSDB. I en typ 1 LSA så definieras en routers anslutna nätverk. Dessa kan vara P2P, transit networks, stub networks eller virtual links. Prefix Suppression gör då att en router ej annonserar de som är definierade som transit network. Typ 2 LSAer är lite meckigare. Routern som skapar typ LSAn kommer här att sätta nätmasken till 255.255.255.255, som då är helt ogiltig för en Typ 2 LSA. Routrar som förstår RFC 6860 kommer ej att installera den här LSAn. Routrar som inte förstår RFC 6860 kommer att installera en host route.

OSPFv2 Prefix Suppression kan aktiveras per process eller per interface. I OSPFv2 processen är kommandot `prefix-suppression`, vid aktivering kommer routern att suppressa allt förutom loopbacks, sekundära adresser och prefix på passiva interface. Vill man stänga av prefix-suppression för ett visst interface är kommandot `ip ospf prefix-suppression disable`. Vill man aktivera prefix-suppression per interface är kommandot `ip ospf prefix-suppression`.

Ett bra användningsområde av prefix suppression är nät där man använder sig av GRE-tunnlar. Då riskerar man inte att det blir recursive routing, dvs att man lär sig tunnelns endpoint-adresser via tunneln.

Stub Router

Det är skillnad på stub router och ett stub area. En stub router är en router som under viss tid, eller permanent, ej kan bli en transit router. Det vill säga att den inte används för att skicka paket vidare till en annan router. Det kan vara bra vid underhåll, ex. om en ASBR kör både OSPF och BGP, alltid annonserar en default route via OSPF, men vid omboot tar BGPn längre tid på sig att

konvergeras vill man ej att trafik går via denna router. ASBRn kommer när den är en stub router att annonsera sina egna LSAer med en infinite metric cost (16,777,215) för samtliga transit-adjacencies. Dess direktanslutna stubnät annonseras som vanligt.

Det här går att ställa in i OSPFv2 processen. Det går att göra med `max-metric router-lsa on-startup announce-time`, då kommer routenr att annonsera transit LSAer med infinite metric tills att announce-time värdet har nåtts. För just BGP finns kommandot `max-metric router-lsa on-startup wait-for-bgp` vilket gör att annonsering blir med infinite metric tills att BGP har signalerat att den är klar eller 10 minuter har passerats.

Graceful Restart

Graceful restart (GR) tillåter en router att fortfarande skyffla trafik medans OSPFv2 processen är nere. Konfigureras i routingprocessen med kommandot `nsf`. Detta då Cisco utvecklade en egen variant, Non Stop Forwarding, innan GR implementerades. Cisco IOS stöder båda varianter.

När GR används så signalerar en router att den äri restart mode. Direktanslutna granar är i helper mode. Grannarna struntar under GR tiden att dfe inte mottar några Hellos och justerar varken LSDB eller routingtabell därefter. Om en DR signalerar att den är i restart mode så förlorar den ej rollen som DR.

```
catalyst#show ip ospf | inc NSF
 IETF NSF helper support enabled
 Cisco NSF helper support enabled
```

Graceful Shutdown

Graceful shutdown används när man stänger av OSPFv2 genom att konfigurera `shutdown` i OSPFv2 processen eller `ip ospf shutdown` på interfacenivå. Routern tar då ned OSPFv2 grannskap (samtliga eller för specifiserat interface), skickar ut egenskapade LSAer med LS Age satt till LSA MaxAge (default 3600 sekunder) för att tömma domänen, skickar Hello-paket till grannar med DR/BDR fälten satta till 0.0.0.0 och en tom grannlista och til sist slutar skicka och ta emot OSPFv2 paket.

Återaktivering sker genom att ta bort konfigurationen ovan.

Default route

En default route kan injiceras i OSPFv2 processen genom kommando `default-information originate [ always ]`. Utan nyckelordet always på slutet så krävs det att en defaultroute redan finns i routingtabellen.

Om en default route ska annonseras i ett NSSA (ej NSSA-TS) så måste det manuellt konfigureras med `area x nssa default-information-originate`.

För att ett NSSA ska acceptera en default-route från en grannrouter i samband med VRF:er krävs kommandot `capability vrf-lite`. Se ex:

```
router ospf 4 vrf veeerf
 capability vrf-lite
```

NSF & NSR

NSF aktiveras i OSPFv2 processen med `nsf ietf`, om inte redan aktiverat. Verifiering med `show ip ospf x nsf`, där x är process ID.
Timers kan sättas i OSPFv2 processen med `nsf ietf restart-interval x`. Det går att stänga av en routers möjlighet att vara NSF helper med `nsf ietf helper disable`.

NSR aktiveras via kommandot `nsr` i OSPFv2 processen.

Max LSA & redistribute maximum-prefix

Sedan IOS-XE 17.11.1 (källa här) är funktionerna Max LSA och Redistribute maximum-prefix aktiverade per default. I tidigare releaser är de avstängda. Max LSA är aktiverat med värdet 50 000 per default och redistribute maximum-prefix är aktiverat med värdet 10 240 per default. max-lsa begränsar hur många LSAer en OSPF process kan ta emot. redistribute maximum-prefix begränsar hur många prefix en process kan redistribuera.

Syns enbart i konfigen om man har manuellt ändrat värdena:

```
router ospf 1
 max-lsa 50001
 redistribute maximum-prefix 10241
```

OSPFv3

Open Shortest Path First version 3 (OSPFv3) är en efterträdare till OSPFv3. Det stora med v3 är att utbyten av LSAer innehållande IPv6-prefix är möjligt. Cisco tillåter även att man byter ut LSAer innehållande IPv4-prefix, men det gör inte alla andra leverantörer. RFC 5340 definierar OSPFv3. IPv6 måste vara aktiverat på routern för att OSPFv3 ska fungera med kommandot `ipv6 unicast-routing`.

Det finns en äldre syntax och en nyare syntax för OSPFv3. Ex. vid aktivering av OSPFv3 på interface med den gamla syntaxen är `ipv6 ospf 1 area 0`. Nya syntaxen är `ospfv3 1 ipv6 area 0`. Jag kommer att använda nya syntaxen här.

Många koncept är samma i OSPFv3 som i OSPFv2. Den här artikeln antar att man känner till OSPFv2.

Skillnader mot OSPFv2

- Inget network-kommando finns. Aktivering av interface sker på interface-nivå.
- Då det går att ha flera IPv6-adresser på ett interface så annonseras samtliga adresser vid aktivering
- Om inga IPv4-adresser finns på en router måste router ID (RID) sättas manuellt
- Nya LSAer, link-local, area och AS
- Det går att använda instanser på en länk. På ett multiaccessnät kan man då dela upp router 1 och 2 i instans 1 och router 3 och 4 i instans 2. Används instanser måste de matcha mellan routrar som ska vara grannar. Instansnummer är mellan 0 och 255.
- OSPFv3 använder begreppet länk istället för nätverk
- Grannskap bildas över IPv6 link-local adresser. Detta gäller även om det är LSAer med IPv4 som ska bytas ut. Undantaget är virtual links då en global IPv6 adress används, stöd för virtual-links finns alltså enbart över routad IPv6
- Autentisering sker ej via OSPFv3 utan via inbyggda autentiseringsmöjligheter i IPv6

LSAer

Många LSAer är samma som i OSPFv2. En del har bytt namn och LSA typ 8 samt typ 9 har lagts till.

LSA	Namn	Beskrivning	Floodas var?

1	Router LSA	Beskriver en router och dess länkar till grannar inom ett area	Inom ett area
2	Network LSA	Skapas av DR och representerar multiaccess transit-nätverket samt koppling till grannar	Inom ett area
3	Inter-Area Prefix LSA	Skapas av Area Border Router (ABR) för att beskriva nätverk som finns i ett annat area	Inom ett area
4	Inter-Area Router LSA	Skapas av ABR så att anslutna areas kan hitta till en ASBR	Inom ett area
5	Autonomous System External LSA	Skapas av ASBR för att beskriva injicerade rötter	Autonomous System
6	x	x	x
7	NSSA LSA	Skapas av en ASBR inom ett Not-So-Stubby-Area (NSSA) för att beskriva injicerade rötter	Inom ett area
8	Link LSA	Beskriver link-local adresser. Skickas om det finns mer än 1 router på en länk.	Lokalt på ett Ethernet-segment
9	Intra-Area-Prefix LSA	Associerar IPv6 prefix med ett transit nätverk genom att peka på en Network LSA och associerar IPv6 prefix med en router genom att peka på en router LSA.	Inom ett area

Funktionalitet har flyttats från Typ 1 och 2 LSAer till Typ 8 och 9. Typ 1 och typ 2 innehåller inte längre information om adresser. Prefix-information återfinner istället i LSA typ 9. Själva topologin finns kvar i Typ 1 och 2 men ej anslutna nätverk.

Då topologi-information och adress-information är separerad i OSPFv3 är protokollet mer effektivt. Om en interface-adress ändras skickas enbart en uppdaterad Link LSA (enbart lokalt) och en nya Intra-Area-Prefix LSA. Då topologin är oförändrad behöver inte SPF köras, enbart prefixen uppdateras i LSDB.

Autentisering

OSPFv3 använder sig av IPv6 inbyggda stöd för autentisering. Autentisering konfigureras på interface-nivå. Det finns två typer man kan använda sig av. Den ena är via IPv6 Authetication

Header (AH), vilket enbart innebär autentisering, och den andra är Encapsulating Security Payload (ESP, vilket innebär autentisering men även kryptering av trafik.

Exempel med AH:

```
interface GigabitEthernet0/1
 ospfv3 authentication ipsec spi 256 sha1 40-TECKEN-HEX
```

Exempel med ESP:

```
interface GigabitEthernet0/1
 ospfv3 encryption ipsec spi 1337 esp aes-cbc 256 64-TECKEN-HEX sha1 40-TECKEN-HEX
```

Kommandot innehåller:

- Security Paramter Index (SPI) som ska vara unik för routern och ska matcha på båda sidor
- Kryptering för esp, i det här fallet aes-cbc 256, samt ett lösenord. Lösenordslängden varierar beroende på kryptering och innehållet ska vara hexadecimalt (får alltså innehålla: ABCDEF123456789)
- Autentiseringsmetod, i det här fallet ovan sha1, följt av hexadecimalt lösenord med varierande längd beroende på vald algoritm

Det går även att implementera autentisering med Authentication Trailer enligt RFC 7166. IPv6 inbyggda IPSec stöd används då ej. Det konfigureras på interfacenivå med kommandot `ospfv3 authentication key-chain KEY-CHAIN-NAME`. Den metoden innebär ingen möjlighet till kryptering av trafik. Nyckel IDn och strängar måste matcha.

I nya releaser av IOS-XE (verifierat i 17.12.3) så kan man inte använda sig av service password-encryption med OSPFv3 encryption. Se mer information här.

Adressfamiljer

OSPFv3 har stöd för att transportera LSAer för flera flera adressfamiljer. Det är det som gör att OSPFv3 kan vara bärare för både IPv4 och IPv6.

Olika instans IDn har mappats till adressfamiljer:

Instans ID	**Adressfamilj**
0	Base IPv6 Unicast
1-31	IPv6 unicast beroende på lokal policy
32	Base IPv6 multicast

33-63	IPv6 multicast beroende på lokal policy
64	Base IPv4 unicast
65-95	IPv4 unicast beroende på lokal policy
96	Base IPv4 multicast
97-127	IPv4 multicast beroende på lokal policy
128-191	Otilldelat
192-255	Reserverade för framtiden

Om inget instans ID tilldelas manuellt kommer routern att automatiskt välja 0 för IPv6-prefix och 64 för IPv4-prefix. Även om man bara använder ett process-ID så delas ingen information mellan instanserna.

I OSPFv3 Hello paket, DBD paket och LSAer finns en AF-bit definierad som sätts enligt instans ID-mappningen ovanför.

Konfiguration av adressfamiljer sker enligt exempel:

```
router ospfv3 1
 router-id 1.3.3.7
 auto-cost reference-bandwidth 1048576
 !
 address-family ipv6 unicast
  area 0 range 2001:DB8:0:1337::/63
  passive-interface default
  no passive-interface Vlan1337
  prefix-suppression
 exit-address-family
 !
 address-family ipv4 unicast
  passive-interface default
  no passive-interface Vlan1337
 exit-address-family

interface Vlan1337
 ospfv3 network broadcast
 ospfv3 hello-interval 1
 ospfv3 dead-interval 3
 ospfv3 priority 5
 ospfv3 bfd
 ospfv3 1 ipv6 area 0
```

```
 ospfv3 1 ipv4 area 0


! Notera instans ID:t nedanför. Den första delen är för IPv4 och andra för IPv6


catalyst#show ospfv3 interface vlan 1337
Vlan1337 is down, line protocol is down
  Link Local Address FE80::BEE7:12FF:FE91:164D, Interface ID 157 (snmp-if-index)
  Internet Address 10.248.254.2/24
  Area 0, Process ID 1, Instance ID 64, Router ID 1.3.3.7
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DOWN, Priority 1, BFD enabled
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
Vlan1337 is down, line protocol is down
  Link Local Address FE80::BEE7:12FF:FE91:164D, Interface ID 157 (snmp-if-index)
  Area 0, Process ID 1, Instance ID 0, Router ID 1.3.3.7
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DOWN, Priority 1, BFD enabled
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
```

Värt att notera om övriga inställningar ovan, ex. hello-intervall, network type och timers, är att de gäller samtliga adressfamiljer.

Prefix Suppression

OSPFv3 har stöd för prefix suppression, det vill säga möjligheten att inte skicka med informaton om transit link prefix. I OSPFv3 så döljs dessa från typ 8 och typ 9 LSAer. Aktiveras per process med kommandot `prefix suppression` eller per interface med `ospfv3 prefix-suppression`. Om man använder sig av adressfamiljer går det att konfigurera ovanför adressfamiljerna, vilket aktiverar det per adressfamilj, eller inuti adressfamiljerna.

Graceful Shutdown

Fungerar som i OSPFv2 men med en lite annorlunda process. Följande sker när man slår in `shutdown` i processen:

- Hello-paket skickas med router prioritet 0. DR/BDR ansvaret försvinner, om ett sådant finns
- Slutar acceptera mottagna Hellos
- Flushar alla skapade LSAer förutom en typ 1 LSA
- Floodar sin typ 1 LSA med cost satt till 65,535
- Efter Dead-interval tar slut och alla grannar anses nere flushas ens egna typ 1 LSA
- Slutar skicka och processa OSPFv3 paket

Den största skillnaden är alltså att OSPFv3 väntar på att Dead interval är slut innan alla grannar är nere. Det tar lite längre tid jämfört med OSPFv2.

Graceful Restart (NSF)

OSPFv3 använder termen Graceful Restart för Non Stop Forwarding (NSF). Aktiveras genom `graceful-restart` i OSPFv3 processen alternativt i adressfamiljskonfläge. Timers går att sätta med `graceful-restart restart-interval 300`. Helper-funktion går att avaktivera med `graceful-restart helper disable`.

NSR går att aktivera med kommandot `nsr` i OSPFv3 processen.

IS-IS

Intermediate System-to-Intermediate System (IS-IS) är ett populärt routingprotokoll för underlays i större nät, exempelvis Internetleverantörer. IS-IS är ett link-state routing protocol. Protokollet byggdes för OSI (Open System Interconnection) protokollet, alltså ej för IP, men transporterar sedan länge IP bra.

IS-IS

Protokollet definierades först i en ISO/IEC standard som heter 10589:2002.
IS-IS använder sig precis som OSPF sig av Dijkstras Shortest Path First (SPF) algoritm för att räkna ut bästa väg till ett nätverk.
Network Service Access Point (NSAP) adressering används för att identifierar routrar, area-tillhörighet och grannskap. Två hierarkier finns, Level 1 (intra-area) och Level 2 (inter-area). Man kan jämföra Level 2 med area 0 i OSPF och Level 1 med övriga areas.
Meddelanden utbyts i IS-IS via Type-Length-Values (TLV). Dessa skickas som MAC-multicast, protokollet är alltså ej beroende av något annat än datalagret. TLVerna kan innehålla olika typer av adressfamiljer (AFI), stöd finns alltså för både IPv4 och IPv6.

RFC 1195 från år 1990 är första definitionen av hur IS-IS ska fungera i IP-nät och kallas här för Integrated IS-IS.

Adressering

Mycket terminologi i IS-IS kommer från OSI protokollet. Här kallas en host för End System (ES) och en router för Intermediate System (IS). System är en nätverksnod, Circuit är ett interface och Domain är ett Autonomous System (AS). Kommunikation mellan två ES kommer i två former, connectionsless-mode och connection-mode. Connectionless fungerar på samma sätt som IP. I OSI så användes protokollet ConnectionLess-mode Network Protocol (CLNP) på samma sätt som IPv4/6 används idag. Spår av CLNP syns fortfarande i IS-IS. `show clns` används för att verifiera IS-IS relaterade parametrar i IOS-XE. Connection-mode kommunikation använde en version av protokollet X.25.

Kommunikation mellan två ES kräver adressering. Adresseringen, för både connection mode och connectionless, använder sig av Network Service Acces Point (NSAP) adressering. En NSAP-adress tilldelas hela nätverksnoden och ej till specifika interface.
Se adressformat:

image.png

NSAP adressen består av två delar. Initial Domain Part (IDP) och Domain Specific Part (DSP).

IDP består av Authority and Format Identifier (AFI) och Initial Domain identifier (IDI). AFI värdet, en oktett mellan 00 till FF, indikerar formatteringen på resterande adressfält. IDI har en varierande längd beroende på adressformatet indikerat av AFI. Den kan till och med vara tomt. Tillsammans indikerar dem routing-domänen som noden är i.

Hur DSP ser ut beror på formatet indikerat av AFI. Den består av en High-Order Domain Specific Part (HO-DSP), av varierande längd, som identifierar vilken del (area) av domänen noden är i. Fältet kan delas upp i subfält. System ID är en unik identifierare för noden. NSAP tillåter fältet att vara mellan 1 och 8 oktetter lång men samtliga implementationer av NSAP låser fältet i 6 oktetter lång.
SEL fältet, även kallad NSAP Selector eller NSEL, är ett 1 oktett långt fält som identifierar en specifik service i eller ovanför nätverkslagret på destinationsnoden som ska processa datagrammet. Går at jämföra med TCP/UDP.

AFI	**Beskrivning**	**IDI längd och innehåll**	**HO-DSP längd och innehåll**
39	Användning av data landskod (ISO 3166)	2 oktetter, numerisk landskod enligt ISO 3166	10 oktetter, areanummer
45	Användning av internationella telefonnummer (ITU-T E.164)	8 oktetter, internationella telefonnummer enligt E.164	4 oktetter, areanummer
47	Användning av internationell kodväljare (ISO 6523)	2 oktetter, internationell organisationskod enligt ISO 6523	10 oktetter, areanummer
49	Lokalt definierat format (privat adressering)	Formellt sett ej på plats	Mellan 0 och 12 oktetter, areanummer

I dagens integrated IS-IS konfigurationer så används alltid AFI 49. Den minsta storleken på en NSAP adress är 8 oktetter och den största är 20 oktetter. Om SEL-oktettens värde är 0, alltså ingen specifik service hänvisas till, hänvisar man till själva noden. En NSAP adress med SEL oktetten satt till 0 kallas för Network Entity Title (NET) och det är adressen som konfigureras på noden, vilket är ett måste i IS-IS.

Adressen består av fält med hexadecimala tecken separerade med punkter.
Exempelvis: `49.0000.1337.5555.6666.00`. 49 är här adressfamiljen, 0000 är area-numret, 1337.555.666 är system ID:t och 00 är SEL värdet, vilket indikerar att adressen är en NET.

Som tidigare nämnt får inte interface någon NSAP-adress. Interface använder L2-adresser på samma sätt som TCP/IP, forwarding sker till en grannes L2-adress. I OSI-nätverk så kallas L2-adresser för Sub Network Point of Attachment (SNPA). Cisco numrerar sina interface i IS-IS med Local Circuit ID, som ökas med 1 för varje interface som läggs till i IS-IS.

Routingnivåer i OSI nätverk

Routing i OSI nätverk använder sig av nivåer (levels) för att beskriva i vilken omfattning routingen sker.

Nivå	Förklaring
Level 0	Routing mellan två ES noder på samma länk, eller mellan en ES och dess närmaste IS
Level 1	Routing mellan ES noder inom ett area i en domän
Level 2	Routing mellan ES noder i olika areas i en domän
Level 3	Routing mellan ES noder i olika domäner

Level 0 routing berör hur en host (ES) hittar sin närmaste router (IS) och vise versa genom att skicka Hello paket. ES skickar ES Hello (ESH) och IS skickar IS Hello (ISH). Går att jämföra med Router Advertisements/Neighbor Advertisements i IPv6.

Level 1 routing berör intra-area routing, alltså routing mellan ES noder som är medlemmar i samma area. Ett area är en samling routrar som i ett link-state routing protocol har detaljerad och komplett visibiltet till hela areats topologi. IS noder samlar in vilka ES noder de har anslutna till sig och annonserar desssa till sina IS grannar.

Level 2 routing berör inter-area routing inom samma domän. Mellan L2 IS så annonseras inte listan med anslutna ES. Area prefix annonseras mellan L2 IS. Om en L1 IS får ett paket som ska till ett ES i ett annat area kommer L1 IS att skicka paketet till närmaste L2 IS oavsett vilket destinationsarea det gäller. Paketet kommer sedan att skickas mellan L2 IS tills att det kommer till rätt area där det åter hanteras av L1 IS. Man kan säga att L1 routing är routing baserat på system ID och L2 routing är routing baserat på area prefix.

Level 3 routing berör routing mellan domäner. I OSI skulle det hanteras av Inter Domain Routing Protocol (IDRP, ISO 10747). I moderna nätverk så kan BGP transportera information om NSAP-adresser (MP-BGP support for CLNS, address-family nsap (unicast)).

En L1 route kommer att föredras över en L2-route. Sist i hackordningen finns External routes (redistributed).

Metrics, nivåer och grannskap

IS-IS metrics tilldelas till individuella interfaces (links). Den ursprungliga specifikationen för IS-IS definierar fyra typer av metrics:

Metric	Förklaring

Default	Måste stödjas av samtliga IS-IS implementationer. Använder vanligtvis länkens bandwidth. Ett högre metric värde är en långsammare länk.
Delay	Delay-värdet på länken
Expense	Pengakostnaden för att skicka trafik över länken
Error	Bitfel på länk

Idag används främst enbart Default-metricen. Ciscos IS-IS implementation ger alla interface en default-metric, 10, oavsett deras bandbredd. Det är en skillnad från OSPF där OSPF tar interfacets bandbredd till metricen. Istället kan man sätta en metric manuellt på interfacet med kommandot `isis metric x [ level ]`. I den ursprunliga IS-IS specifikationen var metric-värdet 6 bitar stort, vilket tillåter värden menllan 1 och 64, och den kompletta path-metricen var 10 bitar stort, vilket tillåter värden mellan 1 och 1023. Idag behövs större spelrum och därmed finns wide metrics, vilket tillåter 24 bitar för interface metric och 32 bitar för hela path metricen. De ursprunliga metrics:en kallas nu för narrow metrics. IS-IS har en administrative distance (AD) på 115.

IS-IS delar upp sina databaser och routingprocess för de olika nivåerna som finns. Ett IS konfigurerad för enbart L1 kan enbart bilda grannskap med IS som också har L1. För att då kunna transportera trafik mellan areas så finns IS som hanterar både L1 och L2. Två stycken IS som hanterar L1 och L2 kan bilda två grannskap, ett för L1 och ett för L2, så länge area numret för L1 matchar.

För varje level aktiverad på ett IS kommer IS:et att skapa och flooda en Link State PDU (LSP), vilket går att jämföra med en OSPF Link State Update innehållande en eller flera Link State Advertisements. Ett IS använder både L1 och L2 LSPer för att beskriva dess grannskap.

Pakettyper

Det finns fyra definierade IS-IS paket:

- Hello packet
- Link state PDU (LSP)
- Complete Sequence Numbers PDU (CSNP)
- Partial Sequence Number PDU (PSNP)

Hello Packets

IS-IS Hello (IIH) används för att hitta grannar, märka tapp av grannar, verifiera tvåvägskommunikation, skapa och behålla grannskap samt välja en Designated IS (DIS, som en DR i OSPF). På broadcast-interface används olika Hello-paket för L1 och L2 grannskap. På P2p interface används ett enda L1L2 Hello (kallas även P2P Hello). Per default skickas Hello paket var 10 sekund men går att modifiera med interface-kommandot `isis hello-intervall x [ level ]`. Hold-timern är ett värde skapart av Hello multiplier gånger hello-intervallen. Standardvärdet för Hello multiplier är 3 vilket leder till en hold time på 30 sekunder. Multipliervärdet går att ändra med interfacekommandot `isis`

`hello-multiplier x [ level ]`.

Hello-intervall och hello-multiplier behöver **inte** matcha mellan IS för att bilda grannskap.

Värdena på en DIS (Designated IS) är alltid 3 gånger mindre än konfigurerat värde. En DIS skickar alltså Hello-paket var 3.333 sekund enligt defaultvärdet. Det är för att hitta avbrott på en DIS snabbare.

Link State PDUer

En Link State Protocol Data Unit (LSP) använder för att annonsera routinginformation. De går att jämföra med OSPF Link State Update innehållande en eller flera LSAer. Det finns i IS-IS ingen skillnad på olika typer av LSPer. LSPerna innehåller olika Type-Length-Values (TLV) vilket beskriver nätverksinnehållet. IS-IS LSPer kan identifieras av ett ett nummer som består av tre delar:

1. System ID från IS som skapade LSP (6 oktetter från ISets NET adress)
2. Pseudonode ID vilket skiljer ISet i sig från LSPer för multiaccessnät där ISet är DIS (1 oktett)
3. LSP Number vilket betcknar fragmentsnumret för den här LSPn (1 oktett). Kallas även för Fragment eller Fragment Number

Tripletten av information ovan kallas för LSPID. I LSPer som beskriver själva routern är Pseudonode IDt satt till 0. För varje routing-nivå ett IS deltar i så skapas en LSP.

För att skilja mellan olika versioner av samma LSP så används sekvensnummer. Sekvensnumret är ett 32 bitar stort värde vilket ökas med 1 vid varje förändring som börjar vid 0x00000001 och slutar vid 0xFFFFFFFF. Skulle maxvärdet nås så måste ISet som skapade LSPn startas om eller byta System ID, men det tar minst 136 år att nå taket.

Varje LSP har en livstid, Remaining Liftetime. När en LSP skapas är den 1200 sekunder (20 minuter) och minskar konstant. IS-IS routrar förnyar sina självskapade LSPer var 15 minuter. Om en LSPs remaining lifetime når 0 så kommer en router ta bort LSPns innehåll från link-state databasen, behålla headern och annonseras den tomma LSPn med en Remaining Lifetime satt till 0. Att annonsera en LSP med Remaning Lifetime kallas LSP purge. Hela LSPn kommer först rensas efter ZeroAgeLifetime har gått ut efter detta, den är 60 sekunder lång. Cisco routrar kommer dock att behålla den tomma LSP headern i 20 minuter.

Eftersom att IS-IS paket enkapsuleras direkt som L2 frames där MTUn kan vara begränsad finns fragmenteringsmetoder för LSPerna. Om placering av samtliga TLVer i en enda LSP skulle överstiga MTUn så kommer routern istället att dela upp TLVerna i flera LSPer. Fragmenteringsnumret ökar för varje LSP med start på 0. Övriga routrar som mottar de fragmenterade LSPerna kommer ej att öka fragmenteringsnumret utan floodar LSPerna som dom är. En konsekvens av denna regel är att **MTU måste vara samma** inom en IS-IS floodingdomän, alltså inom ett area eller mellan samtliga L2-routrar. Om det ej är möjligt måste IS-IS routrar manuellt konfigureras om för att behålla varje LSP som ej är större än den minsta MTUn.

I OSPF skapas olika LSAer beroende på innehåll, area, syfte und so weiter. I IS-IS skapas bara en (möjligtvis fragmenterad) LSP som innehåller all relevant information till routern:

- Grannskap till andra routrar och nätverk (likt typ 1 LSA i OSPF)
- Intra-area och inter-area prefix (likt typ 1, 2 & 3 LSAs)
- Externa prefix (lik typ 5/7 LSAs)

En IS-IS router skapar 1 enda LSA (per level) som beskriver den och dess anslutna nätverk. Är den DIS så kommer den att skapa en Pseudonode ID för varje netvärk den är DIS för.

En LSP har en unik identifierar för hela LSPn och kan enbart floodas, begäras, ackas, förnyas, åldras och flushas i sin helhet. Vid topologiförändringar, ex. ett nätverk som slutas redistribueras in i IS-IS, måste hel LSPn skapas om och flushas på nytt. Det kan anses ineffektivt jämfört med OSPF, där en ny LSA skickas om på nytt, men i IS-IS så behöver en router bara hålla koll på högst ett par-tre stycken LSPer som åldras och förnyas. Det är även lätt att implementera bärande av nytt data/nya funktioner i en LSP då datat är formatterat som TLV information. Skulle en router ta emot LSP innehållande TLVer den ej känner igen kommer den informationen att ignoreras.

Complete och Partial Sequence Number PDUs

CSNP och PSNP är paket som används för att synkronisera link-state databasen. Sekvensnummer används för att beskriva en serie av LSPID värden och ska ej förväxlas med sekvensnummer i individuella LSP paket.

CSNP paket går att jämföra med OSPF Database Description paket. Syftet med CSNP paket är att annonsera en komplett lista över LSPer i sändarens link-state databas. Mottagare av CSNP kan jämföra paketet mot sin egen link-state databas och antingen flooda en nyare eller saknad LSP till sändaren, eller be om en LSP som den själv saknar i sin databas.

Även här fragmenteras paketen till flera CSNP paket om de skulle överskrida MTUn. På P2P länkar byts CSNP paket ut mellan IS under första grannskapsbildandet. På broadcastnätverk skickas CSNP paket periodvis av DIS.

PSNP paket går att jämföra med OSPF Link State Request och Link State Acknowledgement paket. Med hjälp av PSNP kan en router be om en viss LSP eller acka mottagande av LSPn. En PSNP kan begära eller acka flera LSPer. PSNP används på P2P länkar för att både be om LSPer och acka mottagande. På broadcastnätverk används PSNP paket för att be om LSP men ackar sker genom CSNP meddelanden skickade av DIS.

Nätverkstyper

I IS-IS finns bara två nätverkstyper: broadcast och point-to-point (P2P). Det går att köra IS-IS över ex. HDLC, PPP, GRE, Frame Relay och ATM men det finns inga specifika nätverkstyper för dessa interface. I IS-IS finns följande gannskapslägen:

- Down. Första läget efter att IS-IS aktiveras på interface. Inga IIHs har mottagits från granne.
- Initializing. IIHs har mottagits från granne men det är inte säkerställt att tvåvägskommunikation är etablerad, dvs att även grannen mottager IIHs
- Up. IIHs är mottagna och det är säkert att grannen tar emot IIHs

IS-IS över P2P

Över P2P interface förväntar sig IS-IS enbart en granne. I den första implementationen av IS-IS ansågs ett grannskap vara uppe vid första mottagande av IIH från grannen, men då ingen verifiering skedde kunde det leda till problem. Numera frinns ett tre-vägs handskap för IS-IS över P2P-länkar.

Varje IS-IS interface får ett Local Circuit ID. Vilket Local Circuit ID ett interface har fått kan kontrolleras med `show clns interface [ interface ]`. På P2P länkar används Local Circuit ID i IIH paket för att märka identitetsförändringar på andra sidan länken. På broadcastlänkar används Local Circuit ID som Pseudonode IDt om routern är DIS på interfacet. Därför behöver Local Circuit IDt bara vara unikt på riktigt på en broadcastlänk. Samma nummer kan återanvändas på broadcast och P2P-länkar. Local Circuit IDt är 1 oktett stor, begränsningen blir då 256 interface per IS. För att hantera mer än 256 interface finns Extended Local Circuit ID som är 4 oktetter lång.

Trevägshandskakningen baseras på att vardera routers P2P länk annonseras ett adjacency state TLV i sina IIh paket som innehåller följande fält:

- Adjacency Three Way State, status för grannskapet sett från den sändande routern
- Extended Local Circuit ID, IDt för sändande routers interface
- Neighbor System ID, ID på grannskapsroutrar vars IIHs har mottagit
- Neighbor Extended Local Circuit ID, värdet från grannarnas ID i mottagna IIH paket

Tidigare implementationer använde sig enbart av Adjacency Three Way State. Logiken bakom tidiga implementation av trevägshandskakningen är enligt följande när båda routrar anser grannskapet vara Down:

1. Router A mottagar IIH från Router B där Adjacency Three Way State är satt som nere. Router A kan höra Router B, men vet ej om Router B hör Router A. Router A skickar sitt IIH paket med Adjacency Trhee Way State satt till Initializing.
2. Router B mottagare IIH från Router A med Adjacency Three Way State satt till Initializing. Router B vet att tvåvägskommunikation fungerar. Därför skicka den Adjacency Three Way State satt till Up.
3. Router A mottager IIH från Router B med Adjacency Three Way State satt tiill Up. Router A vet att våvägskommunikation fungerar. Router A skickar IIH med Adjacency Three Way State satt till Up och handskakningen är över.

Det här är defaultbeteende som Cisco använder på P2P interface och kan konfigureras i interfacekonfig med `isis three-way-handshake cisco`.

Det finnsf all där ovan kan misslyckas, exempelvis om paket skulle switchas till fel routrar i ATM eller fibernät där RX/TX är separata. För att lösa detta lades Extended Local Circuit ID, Neighbor System ID och Neighbor Extended Local Circuit ID till. Router A skulle här vid första IIH populera Extended Local Circuit ID fältet med sitt sändande interface. Router B skulle svara med sitt eget Extended Local Circuit ID fält populerat med sitt sändande interface, Router A kommer placera Router Bs system ID i Neighbor System ID och kopiera Router Bs Extended Local Circuit ID till Neighbor Extended Local Circuit ID fältet. Därmed kommer Router B att veta vid mottagande av nästa IIH att det är rätt enheter som kommunicerar med varandra. Skulle senare ett IIH paket komma som ej innehåller fälten tas emot kommer det att droppas tyst. Den här metoden konfigureras på interface med `isis three-way-handshake ietf`. Den är även bakåtkompatibel med metoden ovanför.

När grannskapet är uppe kommer routrarna att försöka synkronisera sina link-state databaser. Samtliga LSPer markeras för flooding över P2P interfacet och CSNP paket skickas till varandra. Skulle CSNP paket bytas ut innan första LSP-utbytet kan LSPer som redan är kända avmarkeras för flooding. Ser en granne att en LSP är nyare eller ukänd kommer den även att be om LSPerna via PSNP. Då flooding per default kommer att ske är det här dock inte nödvändigt. Varje LSP som skickas över en P2P länk måste ackas med PSNP eller CSNP paket. IS-IS specifikationerna säger att CSNP paket ej ska bytas ut periodvis över P2P länkar så därmed ackas LSPer med PSNP. Vissa leverantörer implementerar periodvisa CSNPs över P2P länkar men Cisco gör ej det per default, det går dock att aktivera med interfacekommandot `isis csnp-interval x [ level ]`.

IS-IS över Broadcastlänkar

På Ethernetnät enkapsuleras IS-IS paket i IEEE 802.2 Logical Link Control (LLC) formatterade frames. Destination Service Access Points (DSAP) och Source Service Access Point (SSAP) fälten sätts till 0xFE. Samtliga L1 IS-IS paket skickas till multicast MAC-adressen 0180.c200.0014 och alla L2 IS-IS paket skickas till multicast MAC-adressen 0180.c200.0015.

Grannar hittas genom att skicka och ta emot IIH paket. I IIH paketen listas en routers samtliga grannar genom dess SNPA (L2-adress, MAC-adress) som har hittats på det interfacet. Om en router mottager ett IIH paket med sin egen SNPA är tvåvägskommunikation verifierad och grannskapet kan flyttas till Up. Om paketet inte innehåller routerns egna SNPA flyttas grannskapet till Initializing.

En DIS väljs per broadcastnätverk. Valet av DIS baseras på följande:

- Högst prioritet på interfacet
- Vid lika prioritet väljs den router med högst SNPA
- Om SNPA ej går att jämföra väljs den router med högst system ID

Prioritet är ett värde mellan 0 och 127. Sätts per interface med kommandot `isis priority x [ level ]`. Samtliga prioriteter deltar i valet och är preemptive, en ny router på samma segment med högre prioritet kommer att ta över DIS-rollen. Samtliga grannar på ett segment har full kommunikation med varandra, ingen motsvarighet till OSPFs DR/OTHER finns i IS-IS. Det DIS är asnvarig för är att hjälpa routers på segmentet att synkronisera och att representera broadcastsegmentet genom

Pseudonode i LSP.

Synkroniseringen sker genom att DIS skapar och skickar ett CSNP paket var tioende sekund (per default). Övriga routrar jämför CSNP paketet med sin egen databas. Innehåller CSNP från DIS en LSP med ett högre sekvensnummer än det en router har lokalt så laddas den ned från DIS genom att skapar en PSNP som begär LSPn. DIS kommer att ta emot PSNPn och flooda LSPn över segmentet. Skulle DIS CCNP indikera ett lägre sekvensnummer på en LSP, eller sakna en LSP som finns i lokala databasen, kommer en router att flooda LSPn på nätverket. DIS blir alltså bara en referenspunkt här och sköter inte all flooding vilket är skillnad från OSPF.

image.png

Genom annonsering av Pseudonode i LSP så minskar antalet länkar som annonseras. I exemplet ovan så skulle det blir 30 annonserade länkar utan pseudonode men bara 12 med pseudonode. Det minskar storleken på LSP som måste propageras som den är inom ett area utan att någon effekt förloras.

Areas

I IS-IS tilldelas areatillhörighet i NSAP adressen. En router kan ha upp till 3 olika NSAP adresser i en IS-IS instans, så länge de bara skiljer sig i area ID:t, men bygger här enbart en link-state database vilket gör att de konfigurerade areas smälter ihop. För att ansluta till andra areas krävs L2 (och L1/L2) routrar, som har möjlighet att skapa grannskap med routrar i andra areas. Kom ihåg att två separata link-state databaser byggs om en router är L1/L2. L2 routrar går alltså att jämföra med Area 0 i OSPF.

image.png

En L1/L2 router annonserar samtliga direktanslutna IP-nätverk samt alla L1 rötter från sitt eget area. Det sker alltså en injicering från L1 databasen till L2 databasen. Ett L1 area går att jämföra med ett OSPF Not So Stubby-Totally Stubb (NSSA-TS) area. De ser rötter inom ett eget area, kan redistribuera in rötter i areat och mottager en default-route från L1/L2-routern. Per default är L1/L2 läget igång i Cisco IOS, går att ändra på såklart.

Anledningen till att använda flera areas idag är för routesummering. I IS-IS sker routesummering vid L1/L2 routrar med kommandot `summary-address x/x` i router isis-processen.

Autentisering

Stöd finns för klartextlösenord och MD5. Välj MD5. Det finns tre sätt att autentisera paket i IS-IS:

1. Area password för L1, autentisering för samtliga routrar i ett area (kom ihåg att LSP floodas oförändrad, här behövs lösenordet!)
2. Domain password för L2, för att autentisera LSPer mellan L2 routrar

3. Autentisering för Hellos (interfacenivå)

Exempelkonfiguration:

```
! Interfaceautentisering, IIH LAN & P2P, L1/L2. Specifisera inte level-1/2 så gäller det för båda
! För P2P ska ej L1/L2 specas
interface GigabitEthernet0/1
 isis auth mode { text | md5 } { level-1 | level-2 }
 isis auth key-chain x { level-1 | level-2 }

! LSP / CSNP / PSNP
! Genom att ändra L1/l2 blir det area-autentisering eller domänautentisering
router isis
 auth mode { text | md5 } { level-1 | level-2 }
 auth key-chain x  level-1 | level-2 }
```

Interfaceautentisering kan slås på gradvis medans area/domänautentisering måste matcha på samtliga routrar.

IPv6

Samtliga regler för IPv4 gäller för IPv6. De transporteras i LSPer som TLVer. Samma IS-IS process kan transportera IPv4 TLVer såsom IPv6 TLVer.

Passiva interface

Genom att aktivera `passive-interface` på ett interface som inte är med i IS-IS processen så kommer det nätet att annonseras in i IS-IS. Det är så man kan annonsera ex. klientnät.

Vill man bygga en riktigt liten routingtabell där ingen trafik är ämnad för länknäten kan man kombinera detta med `advertise passive-only`. Blir samma effekt som prefix-suppression i OSPF. Väldigt schysst om man ex. använder IS-IS som underlay i MPLS eller EVPN-nät.

Circuit Type

I sin grundkonfiguration så försöker en Cisco IOS router som kör IS-IS att skapa grannskap över både L1 och L2. En router som är konfigurerad som bara L1 eller L2 kommer naturligtvis inte göra det. Vill man specifisera vilken typ av grannskap som ska bildas över ett visst interface kan man definiera `isis circuit-type { level-1-only | level-2-only }` i interfacekonfen.

Exempelkonfiguration

```
router isis
 net 49.1337.1111.2222.3333.00
 is-type level-1-2
 metric-style wide
 log-adjacency-changes all
 summary-address 10.13.137.0 255.255.255.0
 passive-interface Vlan1337
 advertise passive-only
 !
 address-family ipv6
  summary-prefix 2001:DB8::/32

interface GigabitEthernet0/1
 ip router isis
 isis authentication mode md5
 isis authentication key-chain ISIS-KEY
 isis circuit-type level-2-only
 isis metric 100 level-2
 ipv6 router isis
```

Design

Det finns tre övergripande designmetodier för IS-IS.

Den första är Platt (Flat) IS-IS. Här används enbart L2-routrar. Är att föredra när man börjar bygga ett nät. Det blir enbart en databas. Skulle man behöva lägga till areas så går det att konvertera utvalda routrar till L1/L2. En nackdel är att ingen summering av prefix går att utföra.

Designalternativ två är Three-tie campus/hierarchical design. Core/backbone är L2 routrar. Dist-routrar är L1/l2 och ev. routrar efter dist är enbart L1. Summering sker i distarna.

Designalternativ tre är hybrid IS-IS. Det finns ingen ren backbone (enbart L2) utan en collapsed core används som är L1/L2 routrar. Summering sker då i collapsed core.

NSF & NSR

Är Non Stop Forwarding ej aktiverat så går det att aktivera med `nsf ietf` i IS-IS processen. Verifiering med `show isis nsf`. NSF är en IETF standard och innebär samarbete mellan direktasnslutna routrar.

Flera timers finns associerad med NSF. Dessa värden går att konfigurera. T3 motsvarar restart-time i andra protokoll. Interface specifiserar hur länge den väntar på att interface kommer upp innan restart/switchover är klar. Interval är för att säkerställa att NSF omstarter inte sker för ofta och är en hold-down timer mellan omstarter.

```
router isis
 nsf interval 7
 nsf interface wait 15
 nsf t3 manual 60
 nsf lifetime 60
 nsf interface-timer 5
 nsf interface-expires 3
```

Non Stop Routing är en lokal funktion som inte signalerar någonting till grannar. Aktiveras genom `nsf cisco` i IS-IS processen. Togs fram före NSF, NSF bör föredras.

Show kommandon

show isis hostname	Se hostname (NSAP adress)
show isis database [detail]	Se LSDB
show isis neighbors [detail]	Se grannar
show clns interface [interface]	Se information om interface
show clns	Global CNS information (ex. NET)
show clns is-neighbors [detail] show clns neighbors [detail]	Grannar, väldigt lika outputs

Border Gateway Protocol (BGP)

BGP är ett routingprotokoll som används för transport av prefix över Internet. Det har även stöd för många adressfamiljer (AFI), vilket gör att det är en populär bärare av olika routingdata.

Det finns intern BGP (iBGP) och extern BGP (eBGP). En grupp routrar tillhörande samma organisation, eller funktion inom en organisation, grupperas enligt Autonomous Systems (AS). En BGP session mellan routrar med samma AS-nummer är extern BGP och sesioner mellan routrar med samma AS-nummer är intern BGP. En router kan enbart ha 1 AS-nummer. Extern BGP har administrative distance (AD) 20 och intern BGP har AD 200.

Routrar etablerar kommunikation med varandra via TCP 179. Routrar behöver alltså ej vara direktanslutna med varandra för att skapa grannskap. Routinginformation byts ut via Network Layer Reachability Information (NLRI) paket som innehåller prefix och dess path attributes.

Path Attributes

Pre-checks innan path selection algoritm

1. Next hop reachability (NEXT_HOP). Det måste finnas en väg till next-hop i routingtabllen för att routen ska installeras i RIB.
2. iBGP synchronization. Den här funktionen är avstängd per default i samtliga moderna implementationer per default. Men om aktiverad måste det finnas en exakt matchande route från ett IGP (eller statisk route) för att BGP routen ska installeras. i RIB Om OSPF används måste OSPF och BGP router IDs (RID) matcha för att routen ska anses synkroniserad.
3. Prebestpath cost-community. Om pre-bestpath point of insertion (POI) är med i ett prefix via extended community så kommer den att föredras över över andra path attributes.

Path Attributes

Vid val mellan samma prefix finns en lång lista som BGP utvärderar från toppen till botten. Vid första path attribute som vinner så installeras den routen till RIB.

Vissa path attributes måste finnas och stödjas medan vissa inte måste det. Följande varianter av path attributes finns:

- Well-Known Mandatory. Måste stödjas i samtliga implementationer av BGP och måste vara med i varje NLRI. Exempel på detta är ORIGIN, AS_PATH och NEXT_HOP
- Well-Known Discretionary. Måste stödjas i samtliga implementationer av BGP men måste ej vara med i samtliga NLRI. Exempel på detta är LOCAL_PREF.
- Optional Transitive Attributes. Måste ej stödjas av samtliga BGP implementationer. Dessa värden följer med i NLRI paket till andra BGP-sessioner.
- Optional Non-Transitive Attributes. Måste ej stödjas av samtliga BGP implementationer. Dessa värden stannar lokalt och skickas ej vidare via andra BGP sessioner.

1. **Weight.** Optional non-transitive. Cisco propertiär som konfigureras per router och är ej transitive, dvs att värdet stannar lokalt på routern och annonseras ej till andra. Högst weight vinner. Per default får varje lokalt injicerad route till BGP weight 32768.
2. **Local-preference**. Well-Known Discretionary. Högre vinner, defaultvärde är 100 per prefix. Värdet sprider sig inom ett AS men ej utanför ett AS. Används ofta för att tilldela prefix, eller samtliga prefix från en viss granne, ett högre värde för att influera routingväg för utgående trafik.
3. **Accumulated IGP (AIGP)**. Tillåter BGP att lägga till IGP metric till BGP next-hop med remote AS metric värde.
4. **Locally originated**. Rötter som har skapats lokalt vinner över andra. Lite redundant i Cisco-världen då weight slår denna.
5. **AS-path**. Varje gång ett paket lämnar ett AS läggs AS-numret till i NLRIn. Den väg med minst AS-nummer vinner. Det går att prependa, alltså lägga till sitt eget (eller annat) AS-nummer flera gånger för annonserade prefix.
6. **Origin**. Mandatory, transitive, well known. Hackordningen är IGP > EGP > Unknown. IGP är för rötter som har injiceras i BGP med `network` kommandot, EGP är legacy och unknown är för rötter redistribuerade in i BGP.
7. **Multi-exit discriminator (MED)**. Optional, non-transitive. Används när en router har flera anslutningar till ett annat AS för att påverka utgående trafik.
8. **Neighbor type**. eBGP föredras över iBGP.
9. **IGP metric till BGP next-hop**. Bästa väg till next-hop vinner.
10. **IGP cost-community**. Optional, non-transitive. Om IGP är punkten för pre-bestpath point of inseration (POI), vilket har annonserats via extended communities, kommer prefixet att vinna.
11. **Multipath**.
12. **Äldsta routen (endast eBGP)**. Den route som är äldst enligt `show bgp afi safi x.x.x.x` vinner. Tanken är att den mest stabila routen är bäst.
13. **Always compare RID**. Om always compare rid är konfigurerat kommer routen som kommer från granne med lägst router ID att vinna.
14. **Kortast cluster-length (endast iBGP)**. NLRI med kortast klusterlista (från route-reflektorer) vinner.
15. **Lägst peer IP**. Sista valet. Den granne med lägst IP, enligt TCP-sessionerna, vinner.

Accumulated IGP (AIGP)

AIGP utvärdes efter Local Preference. Mellan AS kommer ett AS att skriva AIGP värdet till en NLRI genom att kopiera IGP metricen.

AIGP läggs på ett prefix genom en route-map. I exemplet nedan används network-statement för att injicera nätverket i BGP. Det går att definiera AIGP metric manuellt men i exemplet nedanför så tas metricen från den IGP som används. AIGP behövs aktiveras per granne.

```
route-map AKTIVERA_AIGP permit 10
 set aigp-metric igp-metric

router bgp 1337
 address-family ipv4
 network 1.3.3.7 mask 255.255.255.255 route-map AKTIVERA_AIGP
 neighbor 7.3.3.1 aigp
```

Avaktivera eller ändra hänsyn till PAs

Det går att konfigurera BGP så att den lokala uträkningen struntar i vissa path attributes (PA) för att hitta bästa rutten eller ändrar hur den använder PAt. Här finns några utvalda kommandon:

```
! Aktivera kontroll av router ID
bgp bestpath compare-routerid

! Strunta i AS-path
bgp bestpath as-path ignore

! Strunta i IGP metric för next-hop
bgp bestpath igp-metric ignore

! Strunta i cost-community
bgp bestpath cost-community ignore

! Jämför alltid MED
bgp always-compare-med

! Finns ingen MED, anse den som sämst
bgp bestpath med missing-as-worst
```

Graceful Restart

BGP kallar Non Stop Forwarding (NSF) för Graceful Restart (GR). GRs funktion är att tillåta forwarding och upprätthållande av grannskap under en switchover. BGP gör detta genom att använda en End-of-RIB (EoR) markerare. EoR signalerar att BGP UPDATE meddelanden kommer att avta. Istället för att vänta på fler uppdatering kan en BGP router direkt köra best-path val när samtliga peers har indikerat EoR.

Per default har IOS-XE GR avstängt. GR aktiveras i BGP-processen med kommandot `bgp graceful-restart`. Vid aktivering måste samtliga grannar tas ned hårt, ej soft reset. `clear bgp ipv4 unicast *`. GR kan avaktiveras mot en specifik granne med `neighbor 1.3.3.7 ha-mode graceful-restart disable`.

BGP GR specifiserar två timers, restart-time och stalepath-time. restart-time börjar räkna ned när en TCP session misslyckas mellan peers under en RP (route-processor) switchover. Timern kontrollerar hur länge BGP väntar på ett BGP OPEN meddelande från grannen som utför RP switchover. När en router påbörjar RP switchover så markeras samtliga rötter från den grannen som "stale" men trafik kan fortfarande forwardas genom den. stalepah-time kontrollerar hur länge trafik kan skyfflas till grannen innan de rensas. Vanligtvis är stalepath-time-timern längre än restart-time timern. Värdena konfigureras med `bgp graceful-restart restart-time x` och `bgp graceful-restart stalepath-time x` där x är sekunder.

RPKI

RPKI (Resource Public Key Infrastructure) är en säkerhetsfunktion för BGP-annonseringar. Routrar som kollar efter RPKI kontrollerar att ett ASN får annonsera ett nätblock. Stämmer inte AS-numret med nätblocker kommer inte routen att installeras. Kombinationen nät och ASN inom RPKI kallas för ROA (Route Origin Authorizations).

Det finns olika varianter av RPKI. Hosted RPKI innebär att någon annan, ex. LIR, ansvarar för certifikatshanteringen. ROA administreras via leverantörens portal.

Kontroll av RPKI går att utföra här: https://rpki.cloudflare.com/?view=validator

Additional Paths (add-path)

Per default annonserar BGP enbart den bästa vägen till ett prefix. Det går att annonsera samtliga vägar från en Route-Reflector. Se exempelkonfiguration:

```
router bgp 1337
 template peer-policy IPV4
  advertise diverse-path backup
 address-family ipv4
  bgp additional-paths select backup
  bgp additional-paths install
```

För icke RRs går annonsering och/eller mottagning av additional paths att aktivera och avaktivera med kommandon i exemplet nedanför. Det går att aktivera för en hel adressfamilj eller per granne/peer-template.

```
router bgp 1337
 address-family ipv4
  maximum-paths ibgp 4
  bgp additional-paths receive
  bgp additional-paths send
  bgp additional-paths select best 3
  neighbor 1.3.3.7 additional-paths disable
  neighbor 7.3.3.1 advertise additional-paths best 2
```

För att förtydliga är add-path enbart aktuellt för iBGP.

RT-Filter

I en iBGP topologi används oftast route-reflectors (RR). En RR kommer att annonsera samtliga NLRIer till samtliga RR-klienter. En router kommer ex. enbart att installera VPNv4 rötter om Route Targets (RT) för en viss VPNv4 route i viss VRF finns definierad lokalt. Det gör att en router som inte har samtliga RTs installera, som kanske inte heller ska eller behöver ha alla RTs installerade, kommer att slänga många rötter. Med RT-filter kan man filtrera vad som ska annonseras från en RR.

```
! Aktivering av RT-filter
router bgp 1337
 address-family rtfilter unicast
  neighbor 1.3.3.7 activate
  neighbor 1.3.3.7 send-community extended
```

Verifiering av rtfilter med `show bgp rtfilter unicast all { detail }`.

DMZ Link Bandwidth

DMZ Link bandwidth är en teknik som tillåter unequal coast load sharing i BGP. DMZ länkar är eBGP länkar som ansluter olika AS, även kända som transit links. Information om bandbredd skickas inom ett extended community som är non-transitive.

```
router bgp 1337
 address-family ipv4
  bgp dmzlink-bw
  maximum-paths ibgp 4
  neighbor 7.3.3.1 dmzlink-bw
```

Multicast Source Discovery Protocol (MSDP)

MSDP är en teknik för att stödja multicast-transport över flera AS. Grannar ska vara aktiverade i adressfamiljerna `address-family ipv4 multicast` och/eller `address-family ipv6 multicast`. Inom ett AS måste multicast såklart vara konfigurerat, PIM, RPs und so weiter. Protokoll såsom Bootstrap Router och AutoRP ska ej annonseras till andra AS.

I MSDP är tanken att RPs i ett AS ansluter mot RPs i andra AS. Konfigureras med `ip nsdp peer 1.3.3.7 connect-source Loopback0 remote-as 1337`. Verifiering med `show ip msdp summary`. Specifikt peer kan kontrolleras med `show ip msdp peer 1.3.3.7`. TCP 639 används för MSDP-kommunikationen. MD5 autentisering konfigureras med `ip msdp password peer 1.3.3.7 LOOOSENORD`. Timers konfigureras med `ip msdp keepalive 1.3.3.7 15 35`, där 15 är keep-alive och 35 är dead-timer.

Autentisering

IOS-XE verkar ha en gräns på 25 tecken för password authentication.

Multiprotocol Label Switchning (MPLS)

MPLS är en teknik för att routa trafik baserat på etiketter (labels) istället för på vad som finns i RIB.

Begreppet MPLS är väl lite överanvänt. Används av vissa så snart de har en transport mellan två platser från en (I)SP, oavsett vilken teknik SPn använder. MPLS konfigureras oftast (alltid?) med MP-BGP för att kunna bära olika adressfamiljer, ex. VPNv4/6, för att etablera L2 eller L3VPN.

En router kommer att tilldela varje känt prefix med en etikett. Dessa etiketter kommer routrarna sedan att dela med sig av, via Label Distribution Protocol (LDP), till varandra. Det gör att routrarna känner hur varandra har märkt prefixen i deras routingtabeller och kan då göra forwardingbeslut, alltså vilket interface paketet ska skyfflas ut på, baserat på vilken label ens granne har märkt paketet med.

När MPLS kom var det mycket snabbare än traditionell routing av paket. Idag är skillnaden inte lika stor. Men MPLS minskar forwarding overhead på en router och gör dem därmed fortfarande mer effektiva.

MPLS i kombination med BGP är vad som skapar magi.

Man kan säga att MPLS består av följande komponenter:

- Label Information Base (LIB)
- Label Forwarding Information Base (LFIB)
- Någon distribution av labels (ex. LDP)
- Label-switched path (LSP)
- Label-switched router (LSR)

image.png

Skulle en router ta emot ett paket som inte är märkt med någon etikket på ett MPLS interface så kommer det hanteras enligt FIB. Vid utskick på ett interface som är aktiverat för label switching kommer en tagg att läggas till.

Labels

Forwarding equivalence class (FEC) är en samling prefix som behandlas på samma sätt. De kan ha samma next-hop, samma utgående interface och möjligtvis samma kö-policies.

Labels sprids mellan routrar via LDP. Ett äldre protokoll, Tag Distribution Protocol (TDP), har funnits men är nu gammalt och dåligt.

Label Distribution Protocol (LDP)

När MPLS & LDP är aktiverat kommer LSRer att dela med sig av varandras etiketter. Om Router 1 vet att Router 2s etikett för ett visst prefix är 1337 kommer Router 1 att sätta label 1337. Router 2 kommer då direkt förstå vilket interface paketet ska ut på efter mottagande och vilken etikett Router 2 ska använda för att nästa router ska forwarda paketet.

LDP har möjlighet till statisk tilldelning av etiketter.

Metoder för distribuering av etiketter:
Downstream on Demand (DoD). Varje LSR ber om en etikett för en FEC (forwarding equivalence class). Det finns enbart en etikett per FEC. Det här används för Label Controlled (LC) ATM interface.
Unsolicited downstream (UD). Varje LSR distribuerar en etikett för alla IGP rötter till samtliga LSR-grannar utan att de har blivit ombedda. LIB kommer att visa bindningar från varje granne. Det används på alla interface förutom LC-ATM.

Metoder för behållande av etiketter:
Liberal Label Retention (LLR). Samtliga etiketter sparas i LIB för en FEC. Enbart etiketten från nedströms LSR, enligt FIB; installeras i LFIB. Resten sparas för backup för användning av Fast Reroute (FRR). Det är bra för HA och används på amma interface förutom LC-ATM.
Conservative Label Retention (CLR). Enbart etiketten från nedströms LSR sparas i LIB. Bra för att spara minne och används enbart på LC-ATM interface.

Metoder för kontroll av Label switched path (LSP):
Independent. Varje LSR skapar en lokal bindning för en FEC. Inga andra LSR är inblandade. En svaghet är att vissa LSR:er skyfflar paket innan LSP är igång. Används i de flesta Cisco plattformar.
Ordered. LSR skapar en binding för en FEC om den inser att den är egress LSR eller om den tar emot en etikettbindning från next-hop för denna FEC. Allokerar enbart etiketter för direktanslutna rötter elelr IGP rötter för vilka den har tagit emot en etikettbindning från next-hop LSR. Används på Cisco ATM switchar.

LDP hittar grannar genom att skicka hello-paket till all-routers multicast (224.0.0.4) över UDP 646 med interfacets IP som source. När grannar har hittat varandra skapas en TCP sessioner över port 646. TCP-sessionen kommer dock att sourcas från LDP router ID.

Grannskap kan verifiera med `show mpls ldp neighbor x.x.x.x`. Interface och dess grannskap kan också ses med show `mpls ldp discovery [ detail ]`.

Vilka etiketter som har blivit tilldelade kan ses med `show mpls ldp bindings`. Det går att kolla på labels från en viss granne med `show mpls ldp bindings neighbor x.x.x.x`, där x är grannens router ID. `debug mpls ldp bindings` om man vill se aktion i realtid.

Aktivera LDP

Aktivera först MPLS/LDP globalt i routern med `mpls ip`.

LDP kan aktiveras per interface eller per routingprocess. LDP aktiveras per interface när kommandot `mpls ip` slås. Trots att inte LDP nämns specifikt så är det så LDP aktiveras.För OSPF och ISIS kan MPLS aktiveras per process, se exempel:

```
router ospf 1
 mpls ldp autoconfig area x


router isis
 mpls ldp autoconfig
```

Det är nog en bra idé att aktivera per IGP så man inte missar något. Verifiering sker med `show mpls interfaces [ interface ] [ detail]`. Behöver man exkludera ett interface från autokonfig så kan man slå `no mpls ldp igp autoconfig` i interfacekonfläge.

Timers

Hello intervall och hold-time kan konfigureras med kommandona `mpls ldp discovery hello interval x` och `mpls ldp discovery hello holdtime x`, där x är antalet sekunder. Default är 5 sekunder för Hello och 15 sekunder för Holdtime. Värdena behöver inte matchas utan det lägsta värde väljs under förhandling. Holdtime måste alltid vara minst 3 gånger så stort som Hello time. Förändring av värden gäller enbart nya sessioner och inte redan etablerade.

Förhandlade Hello-timers och Holdtime-timers går att se med kommandot `show mpls ldp neighbor x.x.x.x detail`, pipa efter time om man vill.

Det finns en inbyggd skydssmekanism, backoff timer, för att stoppa LSRer från att försöka bli grannar konstant när de inte är kompatibla. Per default är backoff timer vid försöka misslyckande 15 sekunder och max 120 sekunder. Timers kan justeras med `mpls ldp backoff x x`, där x är sekunder, och verifieras med `show mpls ldp backoff al`l.

Router ID

Router ID i LDP måste vara en routad adress då sessioner etableras via den. Används med fördel en loopback. Router ID kan specifiseras för LDP med kommandot `mpls ldp router-id { 1.3.3.7 |`

`GigabitEthernet0/1 } [ force ]`. Det går alltså att hänvisa till ett interface för att få router ID:t. Utan force på slutet kommer router ID:t att väljas om när process startas om eller router startas om. Med force väljs det direkt och LDP grannskap startar om. Specifiseras inget automatiskt så tas högsta IP på loopback. Finns ingen loopback tas högsta IP på fysiska interface.

Det går att source:a en LDP session från ett interface med interfacekommandot `mpls ldp discovery transport-address interface`. Kan vara nödvändigt om trafik över interfacet inte får komma från Loopback, av någont anledning.

Direktanslutna nät och Penultimate Hop Popping

När ett paket kommer till den sista LSR i LSP, alltså den router som har nätet direktanslutet, så skulle den LSRn behöva göra två lookups. Den första i LFIB, då paketet kommer med en label, och sedan i FIB för att forwarda paketet ut på rätt interface. För att lösa detta finns Penultimate Hop Popping (PHP).

Direktanslutna nät kommer att annonseras via LDP med en implicit-null label (imp-null, label 3). När grann LSR:er sedan ska forwarda trafik till den sista LSRn så kommer de inte att lägga till någon etikett på paketet så att sista LSR bara behöver göra en lookup i FIB.

Per default annonseras direktanslutna nät som imp-null med label 3, då poppas top label. Det går även att annonsera direktanslutna nät med explicit null med label 0, då poppas hela label stacken. Explicit null för IPv6 använder label 3. För att annonsera direktanslutna nät med explicit null används kommandot `mpls ldp explicit-null` i globalt konfläge.

Label 1 finns även, den indikerar att paketet ska processas av en mjukvarumodul.

Autentisering

Autentisering möjliggör MD5-skydd på TCP-sessionerna. Följande konfiguration konfigurerar ett globalt lösenord, fallback-lösenord, för samtliga LDP peers:

```
mpls ldp password fallback LDP_LOSENORD
mpls ldp password required
```

Utan required-kommandot anser routern är autentisering inte är tvingande. Verifiering sker med `show mpls ldp neighbor password`. Det går även att se med show `mpls ldp neighbor detail` att MD5-autentisering sker.

Autentisering kan även utföras per granne genom att specifisera grannens router ID i: `mpls ldp neighbor 1.3.3.7 password LDP_GRANNE_PASS`.

Det går även att använda key-chains med flera nycklar. För att tillåta flera nycklar under samma tid, när flera nycklar är giltiga, kan man tillåta det med `mpls ldp password rollover duration x`, där x är antal minuter man tillåter flera nycklar. Konfigurera sedan en key-chain. Konfigurera även en ACL som matchar grannens router ID. Kommandot för att aktivera det här mot en granne är sedan `mpls`

`ldp password option 1 for ACL_GRANNE_RID key-chain KEYCHAIN_MPLS_GRANNE`. Loggning för detta ska vara på per default men följande kommandon ökar loggningen: `mpls ldp logging password rollover` & `mpls ldp logging password configuration`. Skulle ett lösenord vara i rolloverprocessen kan man se det med `show mpls ldp neighbor password pending`. Byt ut pending mot current för att se nuvarande.

Session protection

Skulle ett interface gå ned så kommer alla etiketter som en LSR känner till via det interfacet att försvinna. I ett nät där IP-konnektivitet fortfarande finns mellan router ID (förutsatt att loopbacks använder) så kan man behålla TCP-kopplet, och därmed etiketterna, genom att konfigurera `mpls ldp session protection` i globalt konfläge. Det kan vara bra om det är stora databaser som byts ut och interfacet bara flappar.

När interfacet då går ned så kommer datatrafiken att routas om enligt nyuppdaterade LFIB men LSR behåller fortfarande labels associerade till interfacet som har gått ned i LIB. För att se hur forwarding ser ut enligt LFIB finns kommandot `show mpls forwarding-table`. Session protection gäller i 24 timmar, har länken inte kommit upp än så kommer TCP sessionen att rivas och etiketterna slängas. Tiden alla etiketter sparas kan modifieras med `mpls ldp session protection duration x`, minst 30 (sekunder) ogh höst oändligt. Kommandot gäller enbart lokalt på en LSR så rekommendationen är att timers är likadant överallt. Debugging går att göra med `debug mpls ldp session protection`. Session protection syns i `show mpls neighor x detail`.

Session protection appliceras med kommandot ovan globalt, men det går att begränsa med ACL. Det är rimligt för anslutningar där det bara finns 1 väg att inte applicera session protection. Skapas en ACL som nekar de routrar som ej ska matchas, applicera sedan session protection med `mpls ldp session protection for ACL_LDP_SESSIONPROTECTION`.

Det går även att använda targeted LDP för att nå samma resultat som session protection. `mpls ldp neighbor x.x.x.x targeted ldp`. Timers för targeted hellos kan sättas med `mpls ldp discovery targeted-hello { hello | holdtime } x`.

IGP Synchronization

IGP sync är en funktion som ska se till att IGPn och LDP har samma vy över nätet. Problem kan uppstå när ett nytt IGP grannskap bildas och routingtabell uppdateras men LDP är inte klar, ex. att etiketter inte har bytts ut än. Då kommer LSPn att gå sönder då trafik routas enligt RIB/FIB och ej enligt LIB/LFIB.

IGP Sync skyddar mot tillfällen då trafik kan block-hole:as och mot felkonfigureringar. IGP Sync gör att de nylärda rötterna ej installeras förrän LDP signalerar att etikettutbytet är klart. OSPF & IS-IS gör det genom att ge LSAn en maximalt hög metric (cost). IGP sync kan aktiveras för OSPF och IS-IS.

```
router ospf 1
 mpls ldp sync
```

```
router isis
 mpls ldp sync

! Går att stänga av per interface
interface GigabitEthernet0/1
 no mpls ldp igp sync
```

Verifiering sker med `show mpls ldp igp sync`. Debugging går med `debug mpls ldp igp sync`.

Per default kan IGP Sync vänta på att LDP blir klart för evigt. Holdtown tiden går dock att konfigurera med `mpls ldp igp sync holddown x`, där x är millisekunder. Det verkar dock inte finnas någon anledning till varför man ska modifiera den här tiden.

På interfacenivå kan man konfigurera hur länge routern ska vänta efter att en länk har flappat och LDP-grannskap har etableras innan synkronisering anses vara klar. Default är 0 sekunder. Modifieras med konfigurationen `mpls ldp igp sync delay x`, där x är sekunder.

Etiketttilldelning

Tilldelning av etiketter (labels) är processen när en router tilldelar en lokal label till ett prefix. Det går att modifiera vilka labels som används, vilka prefix som labels ges till, vilka labels som annonseras och tas emot... med mera.

Per default tilldelas labels till alla transit-länkar. Önskar man inte detta, och därmed minskar storleken i LIB, kan man stänga av detta.
Det går även att applicera prefix-listor och ACL:er för att specifisera vilka nätverk en router ska skapa labels för.

```
! Tilldela ej labels till transit-nätverk
mpls ldp label
 allocate global host-routes

! Tilldela labels till nät i prefix-lista
mpls lpd label
 allocate global prefix-list PREFIX_LISTA
```

Verifiera sedan om en label finns för ett IP-nät med `show mpls ldp bindings x`.

Att ställa in vad som annonseras i IOS-XE är lite bökigt. Först måste man ställa in att samtliga labels ej annonseras, det gör man med `no mpls ldp advertise-labels`. Kruxet här blir att man måste då specifisera för samtliga grannar vad för etiketter man ska annonsera, för nu annonseras inga labels alls. Skapa nu en ACL som specifiserar vilka labels (enligt IP-adresser) som ska annonseras och en ACL som definierar en granne (eller allt förutom en viss granne genom deny, match är det som

gäller). Kommandot blir sedan `mpls ldp advertise-labels for ACL_MED_PREFIX to ACL_MED_LDP_GRANNE`. Verifiering sker med `show mpls bindings IP NÄTMASK advertisement-acls`.

Filtrering av etiketter som installeras är enklare. Skapa en ACL som matchar det man önskar installera. Sedan är kommandot `mpls ldp neighbor x labels accepts ACL_MED_PREFIX`.

Vilket spann av etiketter en LSR ska använda konfigureras med `mpls label range 16 1048575`. I exemplet används maxvärdena. Verifiering med `show mpls label range`. Tilldelning ändras vid omstart eller att MPLS processen rivs och byggs igen med `no mpls ip` följt av `mpls ip`.

Statisk label-tilldelning är möjligt. Specifisering av vilken label som används lokalt för ett prefix görs med `mpls static binding ipv4 10.11.12.0 255.255.255.0 1337`, där 1337 är etikettnumret. För att specifisera vilken etikett som används vid forwarding, samt next-hop, sker med `mpls static binding ipv4 10.11.12.0 255.255.255.255.0 output 10.20.30.40 47740`. Lärda label-bindings från grannar används före de statisk konfigurerade vid forwarding.

LDP Implicit Withdraw

När LDP behöver ändra en label för ett prefix så skickas först ett label withdraw (LABEL-WITHDRAW) meddelande innan uppdateringen. Label withdraw innebär att informationen om etiketten inte längre är giltig och enny kommer annonseras. I äldre versioner av IOS så skickades ingen label withdraw utan när en uppdatering (LABEL_MAPPING) kom så hedrades den helt enkelt. Det äldre beteendet kan aktivera per granne med `mpls ldp neighbor x implicit-withdraw`.

Default route

Per default allockerar IOS-XE en imp-null label för en default-route. Önskar man lägga till en label för default-route så gör man det med det globala kommandot `mpls ip default-route`.

TTL Propagation

Först en förklaring över MPLS TTL beteenden:

1. Label swap. Den översta etikettens TTL minskar och TTL-värdet flyttas över till den nya etiketten. Precis som i IP forwarding.
2. Label push. Den översta etikettens TTL minskar och TTL-värdet flyttas till den utbytta etiketten och ev. extra etiketter. Det händer om en P router routar trafik i en TE-FRR tunnel och en till etikett läggs på mitt i LSP.
3. Label pop. Den översta etikettens TTL minskar och det nya värdet läggs till på en inre etiketten som nu syns. Det händer ej om det nya värdet är större än TTL på den inre etiketten.

Per default kommer en ingress LSR att kopiera IPv4 TTL eller IPv6 hop-limit till samtliga använda etiketter. Det gör att om ett paket skjuts in i ex. en L3 VPN med alltför låg TTL kan paketet slängas mitt i LSP. Det beteendet kan stängas av `no mpls ip propagate-ttl forwarded`. När en ansluten kund kör en traceroute nu kommer de enbart edge LSR med VPN-label, ej den översta etiketten. Slår man av

TTL Propagation helt med `no mpls ip propagate-ttl` kommer kunden inte att se någon av MPLS-routrarna längs LSP. Designmässigt är det bäst att slå av TTL propagatiion för skyfflad trafik vid ingress och egress LSR.

Graceful Restart, NSR

Stöd finns för graceful-restart. Aktiveras med globala konfigurationskommandot `mpls ldp graceful-restart`. Rekommendationen är att både LDP och de routingprotokoll man använder (OSPF/IS-IS/BGP) använder sig av graceful-restart samtidigt. Grace-period konfigureras med `mpls ldp graceful-restart timers neighbor-liveness x`. Hur länge forwarding sker utan grannskap konfigureras med `mpls ldp graceful-restart timers max-recovery x`. Hur länge etiketter sparas under GR konfigurereas med `mpls ldp graceful-restart timers forwarding-holding x`. Verifiering med `show mpls ldp graceful-restart`.

Non Stop Routing (NSR) aktiveras med `mpls ldp nsr`. Verifiering med `show mpls ldp nsr`.

MTU

MPLS mtu kan verifieras med `show mpls interface x detail`. MPLS lägger på en 4 byte stor shim-header. MPLS headern läggs på efter L2 headern men innan L3 headern. Därav måste L2 MTU vara större eller lika än MPLS MTU som i sin tur måste vara större eller lika MPLS MTU.

RSVP-TE

Traffic engineering (TE är en av de starkaste funktionerna i MPLS. TE tillåter att man styr trafik från source tilld estination baserad på olika attribut, exempelvis på bandbredd och länkfärger. OSPF och IS-IS kan transportera de här attributen. RSVP utökades även med RSVP-TE för att stödja MPLS-TE. RSVP använder IP protokoll 46, alltså ej UDP/TCP.

Se exempel för aktivering via IS-IS nedan. Loopback:en ska vara router ID för MPLS TE. TE aktiveras för en IS-IS level eller ett OSPF area.

```
router isis
 metric-style wide
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
```

Verifiering med `show mpls traffic-eng topology level-2 brief`. Brief kan bytas ut mot att kolla på ett specifik router ID för MPLS TE.

Verifieringar för att se grannar kan utföras med `show mpls traffic-eng link-management igp-neighbor`. Statistik går att se med `show mpls traffic-eng link-management statistics`.

Följande tre meddelanden, och dess innehåll, används för att signalera TE LSPer:

PATH. PATH är ett meddelande som skickas hop-för-hop från source till destination (source till destination kallas i MPLS för head till tail). PATH innehåller information om previous hop (PHOP) så att slutet av LSP känner till hur den ska skicka svarstrafik samma väg.
PATH innehåller i RSVP-TE följande objekt:

- **Label Request Object (LRO)**. Används för att begära TE etiketter längs LSP men bär inte några etikettvärden.
- **Explicit Route Object (ERO)**. Resultatet av patch calculation (PCALC) algoritm som informerar PATH meddelanden om vilken väg de ska transporteras.
- **Record Route Object (RRO)**. Används för att spara den väg PATH-meddelanden har tagit. Användbart när man använder loose-hop expansion för att förhindra signalerings-loopar.
- **Session Attribute Object (SAO)**. Innehåller information om sesionen, vilket läge/mode, nod/bandbredd protection flags samt fast-reroute (FRR)
- **Sender Tspec**. Innehåller information om bandbreddsreservering enligt genomsnittet.

RESV. Skickas hop-för-hop enligt PHOP-informationen från PATH. Följer alltid samma väg som PATH. Next-hop (NHOP) finns inuti RESV så att uppströmsnoder vet LSPerna korrekta väg. RESV innehåller följande objekt:

- **Label object**. Innehåller etikett-objektet, det faktiskt värdet för TE tunneln. Tillsammans med NHOP värdet definierar Label object LSP forwarding plane.
- **Record Route Object (RRO)**. Används för att spara vilken väg RESV-meddelandet har tagit på samma sättt som RRO i PATH.

PATHERR. Felmeddelande när något går fel med RSVP-signalleringen. Kan ex. vara att tillräcklig bandbredd ej finns tillgänglig eller att en länk har gått ned. När ett PATHERR meddelande tas emot kommer headend att räkna om LSP.

Tunnel-interface används för att utföra routingen. Den ska vara unnumbered med hänvisning till MPLS-TE RID lookback-interface.

```
mpls traffic-eng tunnels


interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 1.3.3.7
 tunnel mpls traffic-eng affinity 0x0 mask 0x0
 tunnel mpls traffic-eng path-option 10 dynamic
 tunnel mpls traffic-eng record-route
```

Record-route används för att märka loopar. Verifiering med `show mpls traffic-eng tunnels tunnel 1`. På en tunnel mitt i LSP, som inte har någon aning om vad tunnel 1 är för något, kan man slå `show mpls traffic-eng tunnels role middle`.

TE Attribut

Det finns olika attribut som kan tilldelas tunnlar. TE metric är ett vanligt attribut som kan tilldelas interfacfe. Den används ofta för att skapa två topologier i ett nätverk. En topologi som är baserad på IGPn, som används för dataflöden, och ett som är baserat på TE metric för ex. voice-trafik.

```
interface GigabitEthernet0/1
 mpls traffic-eng administrative-weight 5


interface Tunnel2
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 1.3.3.7
 tunnel mpls traffic-eng affinity 0x0 mask 0x0
 tunnel mpls traffic-eng path-option 10 dynamic
 tunnel mpls traffic-eng path-selection metric te
```

Ett annat attribut är affinity, även känk som link coloring. Det är ett fyra byte stort hexadecimalt värde där värdet representerar en färg.
Röd motsvarar 0001/0x1, Grön motsvarar 0010/0x2, Blå motsvarar 0100/0x4 och Orange motsvarar 1000/0x8. I konfigurationen så definieras Affinity. Om Affinity är satt till 0 så "bryr sig" routrarna om det. Om den ej är satt till noll så innebär det "ignorera".

För att konfigurera en tunnel att ex. traversera Gröna länkar när dessa finns sätts `tunnel mpls traffic-eng affinity 0x2 mask 0x2` på tunnelinterfacet.

Med `tunnel mpls traffic-eng affinity 0x0 mask 0x2` så instruerar man TE att ta vilken länk som helst som *inte* är grön. Notera att affinity är satt till 0.

Med `tunnel mpls traffic-eng affinity 0xE mask 0xE` så instruerar man TE att ta en väg som är Grön, Blå och Orange samtidigt. `tunnel mpls traffic-eng affinity 0xE mask 0xF` gör att vägen måste vara Grön, Blå och Orange men *inte* Röd.

TE kan begära bandbredd. Det konfigureras på tunnel-interfacet.

```
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 20000
```

Verifiering med `show ip rsvp sender filter session-type 7 tunnel-id x`. Går även att se med `show ip rsvp interface`.

Skulle man överallokera bandbreddstilldelning kommer nya tunnlar ej att kunna bildas.

Bandbredd går att automatiskt justera för TE-tunnlar beroende på uppmätt trafik. Funktionen tar medelvärde från dataöverföringshastigheten genom tunneln och periodvis justerar den signalerade bandbreddsreservationen. Funktionen aktiveras globalt med en intervall för periodvisa mätningen med en default på 5 minuter. Det går sedan att justera per tunnel.

```
mpls traffic-eng auto-bw timers frequency 60


interface Tunnel14
 tunnel mpls traffic-eng bandwidth 5000
 tunnel mpls traffic-eng auto-bw frequency 300 max-bw 10000 min-bw 2000
```

Det går att manuellt specifisera vilken väg paket ska ta genom explicit path där next-hop adresser specifiseras. Se exempelkonf:

```
ip explicit-path name EXPLICIT_PATH enable
 next-address 1.3.3.1
 next-address 1.3.4.1
 next-address 1.3.5.1
 next-address 1.3.6.1
 next-address 1.3.7.1

interface Tunnel10
 tunnel mpls traffic-eng path-option 10 explicit name EXPLICIT_PATH
```

Det går även att göra genom att specifisera TE IDs. Vid flera länkar mellan LSRer så kan vilken som av dessa användas.

```
ip explicit-path identifier 11 enable
 next-address 3.3.3.3
 next-address 9.9.9.9
 next-address 2.2.2.2
 next-address 10.10.10.10

interface Tunnel11
 tunnel mpls traffic-eng path-option 10 explicit identifier 11
```

Man kan även specifisera att trafiken måste gå över en viss router genom loose hop expansion. Vägen till denna router specifiseras inte.

```
ip explicit-path name PATH_1_6_11_LOOSE enable
 next-address loose 6.6.6.6
```

```
interface Tunnel13
 tunnel mpls traffic-eng path-option 10 explicit name PATH_1_6_2_11_LOOSE
```

Skicka in trafik i TE-tunnlarna

Man måste ju få in lite trafik i tunnlarna för att det finna TE-nätet man har byggt faktiskt ska komma till användning.

I ett MPLS nät utan TE så skyfflas trafiken helt beroende på labels. I det här fallet vill vi inte det först. Det enklaste man kan göra äratt statiskt routa trafik in i TE-tunneln. För att ex. nå en loopback på en annan LSR via TE-interfacet i globala routingtabellen kan man konfigurera `ip route 1.3.3.7 255.255.255.255 Tunnel10`. Verifiera att tunnel är uppe med `show mpls traffic-eng tunnels tunnel 10`.

Det går även att använda auto-route. Auto-route programmerar automatiskt en statisk route för tunnel-destinationen ut genom tunneln som den har konfigureras.

```
interface Tunnel 10
 tunnel mpls traffic-eng autoroute destination
```

Även policy routing går att göra med en route-map.

Det går även att göra så att Tunneln blir en P2P länk för IGP-protokollet med `tunnel mpls traffic-eng forwarding-adjacency holdtime 10000`.

Graceful restart

Aktiveras per interface med `ip rsvp signalling hello graceful-restart`. Följande justeringar kan också installeras:
`ip rsvp signalling hello graceful-restart mode full`, `ip rsvp signalling hello graceful-restart refresh interval 6000`, `ip rsvp signalling hello graceful-restart refresh misses 5` och `ip rsvp signalling hello graceful-restart send recovery-time 90000`.

MPLS-TE Fast Reroute (FRR)

FRR är ett sätt att få till high-availability i nätet. Om en länk eller nod längs TE-tunneln går ned så finns en pre-signalerad backupväg redo och trafik kan routas in i backuptunneln så snart felet märks. TE FRR terminologi följer:

PLR - Point of local repair. Head-end (början) längs backupvägen. När ett fel hitas så routar den paketen in i backuptunneln. Det är snabbare än att signalera tillbaka till den ursprunliga TE LSP head-end:en för omräkning.

MP - Merge point. Tail-end (slutet) av backupvägen. När trafik anländer på MP så routas trafiken enligt den vanliga TE vägen igen. PLR och MP arbetar ihop för att säkerställa att noder uppströms

och nedströms påverkas minimalt medans FRR är aktivt.

Det finns olika typer av skydd:

1. **Path protection**. En presignalerad backupväg från head till tail per path-option ifall fel uppstår någonstans längs LSP. Enkel att konfigurera och skyddar automatiskt samtliga hop längs vägen utan att behöva använda PLR. Konvergerar långsamt och skalar dåligt.
2. **NHOP protection (next-hop protection)**. Återställer trafik till en LSPs next-hop genom att routa runt länkar som gått ned. Skalar bättre är NNHOP om det finns många LSRer.
3. **NNHOP protection (next-next hop protection)**. Återställer trafik till en LSPs väg till next-hop efter next-hop. Bättre skydd ärn NHOP men skalar sämre. Kallas även för node protection.

Se exempelkonfiguration för NHOP protection. Här specifiserads vilka länkar som inte ska traverseras och specifiserar en backup-path på utgående interface för den ursprunliga TE tunneln. För att använda NNHOP protection så specifiseras en nod längre bort i exclude-address.

```
ip explicit-path name FRR_AVOID_NEXTHOPS enable
 exclude-address 1.3.3.7

interface Tunnel50
 tunnel mpls traffic-eng path-option 10 explicit name FRR_AVOID_NEXTHOPS

interface GigabitEthernet0/1
 mpls traffic-eng backup-path Tunnel30
```

För att märka att en länk gått sönder man kan använda RSVP hellos.

```
ip rsvp signalling hello

interface GigabitEthernet0/1
 ip rsvp signalling hello
 ip rsvp signalling hello refresh interval 5000
 ip rsvp signalling hello refresh misses 4
```

Verifiering med `show ip rsvp hello { instance } { summary }`.

Numera används helst BFD istället.

```
ip rsvp signalling hello bfd

interface GigabitEthernet0/1
 ip rsvp signalling hello bfd
```

Verifiering med `show ip rsvp hello bfd`.

Används OSPF i underlay så bärs LSAer för att bygga automatiska tunnlar. Aktivera med globala kommandot `mpls traffic-eng auto-tunnel backup`. Använd debug för att se dem byggas med `debug mpls traffic-eng auto-tunnel backup all`. Verifiering med `show mpls traffic-eng auto-tunnel backup`.

Segment Routing

Segment Routing (SR) är en ersättare till LDP och RSVP-TE. Individuella noder och des grannskap har segment IDn (SIDs) och varje segment har en label-bindning. Det tillåter trafik att traversa hela nätverket enkapsuleras inuti MPLS och individuella länkar kan bli valda genom att specifisera segment labels. En SR mapping server kan användas för att låta LDP och SR arbeta ihop.

Segment Routing Global Block (SRGB) får ej överlappa med den globala MPLS label allockeringen. SRGB spannet behöver inte vara unikt per router, alltså kan samma etikett återanvändas för samma prefix över flera routrar. Etikett per prefix-SID måste vara unikt i nätverket.

RSVP-TE kan användas samtidigt som SR. SR-TE är dock det som bör användas.

SR kan använda en mapping server (SR mapping server, SRMS) för att allokera prefix-SIDs. Utan en mapping server så allokeras prefix-SIDs lokalt baserat på SRGB av varje LSR.

Generalized MPLS

GMPLS är en förlängning till MPLS konceptet där ett path attribute som kan identifiera ett flöde kan specifiseras. GMPLS inriktas mot optiska nätverk. Kallas även för Multiprotocol Lambda Switching. GMPLS används i samband med Wavelength Division Mutliplexing (WDM) system.

GMPLS sätter upp "ljusvägar" ände till slut. Precis som med MPLS-TE (traffic engineering) kan man specifisera hur mycket bandbredd som ska tilldelas. GMPLS kan även inkludera information om vilka länkfärger soms ka tilldelas, L1 information såsom våglängd, vilka fiber i en kabel und so weiter. Om samma typ av våglängd tilldelas ett flöde från början till slut behöver inte signalen ändras av WDM-systemen flera gånger vilket ökar effektiviteten.

GMPLS stöder dubbelriktade (bidirectional) LSPer vilket inte stöds i vanliga MPLS nät. Med dubbelriktade LSPer minskar uppsättningstiden.

MPLS Transport Profile

MPLS-TP är en teknik för att justera det vanliga MPLS-bettende för att bättre emulera TDM-nätverk (Time-division multiplexing). MPLS-TP togs fram för stödja OAM (operations, administrations and maintenence) funktioner som finns i TDM och SONET/SDH-tekniker. MPLS-TP ersätter etikett-allokeringstekniker såsom SVP-TE, LDP, BGP och SR.

MPLS-TP saknar stöd för följande funktioner:

- Penultimate Hop Popping (PHP)
- ECMP (MPLS-TP kräver symmetrisk routing)
- Label merge

MPLS-TP har följande fördelar över traditionell IP/MPLS:

- IP behöver ej konfigureras
- Avancerad OAM
- Felrapportering. En del av OAM, meddelanden inkluderar Link Down Indiciator (LDI), Lock Report (LKR) och Alarm Indication Signal (AIS)

MPLS TP konfigureras per länk. Man behöver specifisera next-hop manuellt. Det kan antingen ske genom att specifisera next-hop mac, next-hop IPv4 eller genom att specifisera `medium p2p` för P2P-länkar. MAC-adress 0180.c200.0000 används som destination på P2P-interface, vilket vanligtvis används för STP. Länk-IDn måste konfigureras för samtliga metoder.

```
! Exempelkonfiguration

interface GigabitEthernet0/1
 description TX-MAC
 mpls tp link 1 tx-mac 0000.1337.0001

interface GigabitEthernet0/2
 description TX-IPv4
 mpls tp link 2 ipv4 10.3.3.7

interface GigabitEthernet0/3
 description P2P
 medium p2p
 mpls tp link 3
```

Verifiering av ovan sker med `show mpls tp link-numbers`.

MPLS-TP behöver ett statiskt konfigurerat spann för etiketter, då LSPer provisioneras manuellt. Varje router har en statisk range som är den dynamiskt tilldelade delat med 10. Dvs att med dynamisk range 7000-7999 är den statiska 700-799. Router IDt behöver ej vara routbart.

```
mpls label range 7000 7999 static 700 799
mpls tp
 logging events
 router-id 1.3.3.7
```

Verifiering sker med `show mpls tp summary` och `show mpls label range`.

BFD går att kombinera med MPLS-TP tunnlar.

MPLS-TP tunnlar liknar MPLS-TE tunnlar, men den kan ha två stycken slut. Begreppen "working" och "protect" används, vilket kommer från andra protokoll baserat på kretskoppling (circuit-based protocols). Konfigurationen ser ut som att man manuellt bygger LFIB då man identifierar lokala etiketter, fjärran etiketter och utgående interface. in-labels är LSPn från andra hållet, alltså vilka den andra routern ska använda för LSPer för utgående trafik. Konfigurationen måste spegelvänt matcha mellan routrar.

```
interface Tunnel-tp56
 no ip address
 no keepalive
 tp source 1.3.3.7 global-id 0
 tp destination 7.3.3.1 global-id 0
 bfd { template }
 working-lsp
  out-label 105 out-link 1
  in-label 707
  lsp-number 0
 protect-lsp
  out-label 105 out-link 2
  in-label 101
  lsp-number 1
```

Man måste manuellt konfigurera samtliga mittpunktsroutrar.

```
mpls tp lsp source 1.3.3.7 tunnel-tp 56 lsp working destination 7.3.3.1
 tunnel-tp 56
  forward-lsp
  in-label 705 out-label 607 out-link 0


mpls tp lsp source 1.3.3.7 tunnel-tp 56 lsp protect destination 7.3.3.1
 tunnel-tp 56
  forward-lsp
    in-label 105 out-label 301 out-link 3
```

Verifiering med `show mpls forwarding-table labels x - x`. Debugging går med `debug mpls tp lsp-ep` och `debug mpls tp event`. Använd `show mpls tp tunnel-tp 56` för att kolla på tunneln.

När transporten är klar enligt ovan så kan man bygga pseudowires över MPLS-TP.

Inter-AS MPLS

Det finns flera sätt att koppla ihop olika AS.

Ett är att skapa länknät per VRF mellan olika AS. Trafiken på länknäten mellan AS är här inte MPLS-enkapsulerad utan IP-skyfflad. Det är den enklaste lösningen med låg relativ komplexitet, ingen koordinering för RDs, RTs eller annan VPN-information behövs mellan leverantörer. Nackdelen är att det skalar dåligt då varje VRF kräver ett eget länknät och en egen BGP peering.

Det går att aktivera Carrier Supporting Carrier (CSC) på länknäten mellan AS för att få en hel LSP. Det görs i BGP-konfiguration genom `neighbor 1.3.3.7 send-label`.

Det går även att etablera VPNv4/v6 sessioner mellan ASBRer för att utbyte rötter. Sessionerna är via eBGP och finns i globala tabellen, därmed behövs bara en transit-länk. Tekniker såsom TE, mLDP och CSC kan då sträckas över AS-gränserna. De olika AS:en måste komma överens om exakta RT policies per kund. Skulle de inte komma överens om detta krävs RT-rewrites vilket skalar dåligt.

Ska multicast stödjas så bör `ip pim bsr-border` konfigureras på interface mellan AS så att RP (Rendezvous Point) information inte läcker mellan.

För att en ASBR ska informera ett annat AS om VPNv4/6 rötterna så måste den först kunna installera dem. Lösning 1 är att lokalt definiera samtliga VRFer och dess RTs, det skalar dåligt. Lösning två är att konfigurera ASBR som en route-reflector, det gör man genom att konfigurera den riktiga route-reflectorn som en route-reflector-client. VPN-rötter som kommer från en RR-client installeras i BGP tabellen. Den bästa lösningen är att instruera ASBR att spara samtliga VPN-rötter oavsett om RTs importeras eller nej. Det gör man genom att konfigurera `no bgp default route-target filter` i BGP VPNv4/6 kontext. Enbart VPN-rötter kommer att bytas ut trots att peering är via globala tabellen.

Aktivering av grannar är som vanligt, trots att det är eBGP:

```
router bgp 1337
 neighbor 7.3.3.1 remote-as 7331
 address-family vpnv4
  neighbor 7.3.3.1 activate
 address-family vpnv6
  neighbor 7.3.3.1 activate
```

L2VPN med samma teknik som ovan är väldigt komplext.Pseudo-wires (PW) kopplas ihop via multi-segment PWs (MSPW). Det görs statiskt men går även att göra med BGP auto-discovery.

Den tredje varianten för att koppla ihop AS är like CSC eller UMPLS men MPLS service label ändras ej. Ingen label swap sker för Vpnv4/v6 eller för PW. BGP labeled-unicast används mellan ASBRer,

vilket leder till att PE loopbacks läcks mellan AS. Det anses osäkert och kräver samarbete mellan AS men resultatet blir att MPLS tjänsterna är konsekventa från början till slut. En enda transit-länk behövs mellan AS.

Tjänster

Över MPLS kan olika tjänster levereras, ex. L3VPN, EVPN, VPWS.

Virtual Private Wire Service

VPWS är en teknik som skapar P2P-länkar mellan siter. E-LINE eller EoMPLS är delarav VPWS som gör det för Ethernet-förbindelser. Pseudowires (PWs) används för att skapa P2P förbindelserna. PWs signaleras via LDP och har control-word (CW) aktiverat.

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.

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

EVPN med VXLAN

BGP är kontrollplan och VXLAN är dataplan. RFC 7348. 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

BGP rötter / EVPN Route-Types

Ett antal typer av BGP rötter 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	
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 (3VNI*)
- 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.

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).

Ingress replication

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

Ingress replication-grannar 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
```

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-rötter 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.

Patchning IOS-XE

Patchning av IOS-XE burkar fungerar bra. Glöm bort all kunskap om att sätta `boot system variablar` för så ska det inte hanteras längre (men det går).

Bundle mode vs packages

Skulle man fortfarande peka ut filer direkt med boot parameter så kallas det här för bundle mode.

När man går över till install mode så skapas packages.conf filen automatiskt. Observera att man inte kan konvertera gradvis utan måste köra hela kommandot.

`*Oct 2 11:18:10.833: %INSTALL-3-OPERATION_ERROR_MESSAGE: R0/0: install_engine: Failed to install_add package bootflash:ir1101-universalk9.17.12.04.SPA.bin, Error: [1|install_add(ERR,)]: Booted in bundle mode. For Bundle-to-Install mode conversion, please use one-shot CLI - install add file <> activate commit`

Förfarande

Ladda upp önskad programvara till burken. SCP funkar bra. Använd sedan kommandot `install add file flash:cat9k_iosxe.17.12.03.SPA.bin activate commit`. Burken kommer nu att kopiera över filerna i .bin paketet till switchen samt till övriga medlemmar i stacken (om stackad). Den kommer sedan fråga om den får boota om, svara ja med Y.

Om man är för långsam med att svara Y kommer omstart inte ske. Däremot är switchen redo att installera det nya paketet. Återuppta med `install activate commit`. Efter omstart kan auto-abort-timer vara aktiverad. När den går slut så kommer switchen att boota tillbaka till gammal mjukvara. Verifiera med `show install active`. För att stoppa auto-abort-timer slå `install auto-abort-timer stop` i exec-läge.

Lager 1 (fysiskt)

Samlad artikel om lager 1 mumbojumbo som i alla fall för mig är lite mystiskt och magiskt. Inte för att det är koolt utan för att jag inte förstår det.

Feldetektering

Carrier delay är en timer som körs i mjukvaran för att identifiera när en länk går ned enligt lager 1. Per default är carrier delay-värdet 2 sekunder för att undvika flappar. När ett annat värde är konfigurerar så syns det i `show interface x` output:en.

```
interface GigabitEthernet0/45
 carrier-delay 5

c9300#show int gi 1/0/45 | inc Carrier
  Carrier delay is 5 sec

interface GigabitEthernet0/45
 carrier-delay msec 50

c9300#show int gi 1/0/45 | inc Carrier
  Carrier delay is 50 msec
```

Per default så släcks lampan på switchen när en länk går ned men det tar 2 sekunder innan en log genereras. Det är då IOS väntar med att notifiera andra processer innan carrier delay timern har gått slut.

På vissa plattformar finns även möjlighet att konfigurerea `carrier-delay up x`.

NETCONF & YANG

NETCONF är en IETF standard som står för network Configuration Protocol. Det är en standard för att konfigurera, ta bort och modifiera konfigurationer på nätverksenheter. YANG är en modell där nätverkskonfigurationer och lägen struktureras i en trädmodell. YANG används som datamodell för NETCONF. Man kan säga att YANG för NETCONF fyller samma roll som MIBs i SNMP.

NETCONF innehåller funktioner såsom "GET", "GET-NEXT" och "SET" m.m.. NETCONF har utöver det extra funktioner såsom testning av konfigurationer innan det tillämpas, möjlighet att konfigurera flera nätverksenheter samtidigt och bulk-get operationer som är mycket snabbare än SNMP. Att utföra transaktioner över ett helt nätverk är enklare då NETCONF hanterar felhantering och sekvensering. NETCONF använder sig av SSH. NETCONF meddelanden struktureras enligt XML.

LISP

Locator/ID Separation Protocol (LISP) är ett protokoll som stödjer både tunnling av data (UDP 5431) och har ett kontrollplan (UDP 5432) för att annonsera och begära routinginformation.

Om man jämför LISPs kontrollplan med traditionell routing kan man säga att traditionell routing innebär att routing-tabellen är populerad ifall trafik ska skyfflas någonstans. LISP fungerar istället på begäran. När trafik ska till en viss IP-adress så frågar routern en server var den IP-adressen befinner sig. Man kan säga att det är en routing-tabell på begäran, "on demand".

Termer

Term	**Förklaring**
RLOC (Routing location)	Var en host befinner sig, L3-enheten som hantear klienten.
EID (Endpoint ID)	En klient, /32 route
MS (Map server)	Server som håller koll på EID-till-RLOC mappingar. Map-requests forwards från Map resolver (MR) till MS. MS svarar direkt till ETR som registrerade EID och den ETRn svarar med en map-reply till frågeställande ITR.
MR (Map resolver)	Samtliga map-requests skickas från ITR:er till MR.
ALT	Virtuell nätverk som används för inter MR/MS kommunikation. Ovanlig. Ofta är MR/MS-rollerna på samma router
ITR (Ingress tunnel router)	Inkommande trafik går till en ITR. ITR frågar MR till vilken RLOC den ska routa trafik.
ETR (Egress tunnel router)	Utgående trafik, trafik från ITR går till ETR. ETR har registrerat sina subnät och hostar (EID-till-RLOC mapping) till MS. En router med anslutna subnät är både ETR och ITR beroende på vilket perspektiv man ser dem från, lättast är att kalla dem xTR.
PITR (Proxy ITR)	PITR hanterar ITR-funktionalitet för kommunikation som initieras utanför LISP-domänen med destination innanför.
PETR (Proxy ETR)	PETR hanterar ETR-funktionalitet för kommunikation som initieras inom LISP-domänen med destination utanför. PxTR används för att hänvisa till båda funktionerna.
Shared Model	RLOC finns i globala routing-tabellen och EIDer i olika VRF. Varje EID VRF har ett eget instance ID (IID).

Parallel Model	RLOC finns i samtliga VRFer.

Konfiguration

Nedan är ett konfigurationsexempel av LISP-konf på en xTR. Detabase-mapping är för en loopback-adress.

```
router lisp 1
 eid-table default instance-id 0
  database-mapping 1.3.3.7/32 IPv4-interface GigabitEthernet0/1 priority 10 weight 10
  ipv4 itr map-resolver 10.10.10.10
  ipv4 etr map-server 10.20.20.20 key LISP-PASSWORD
```

Nedan är konfigurationsexempel på MR/MS. En route pekas out som en site.

```
router lisp
 ipv4 map-server
 ipv4 map-resolver
 ipv6 map-server
 ipv4 map-resolver
 site SITE1337
  authentication-key LISP-PASSWORD
  eid-prefix instance-id 101 10.1.10.0/23 accept-more-specifics
  eid-prefix instance-id 102 10.10.10.10/32
  eid-prefix instance-id 102 2001:10:10:10::10/128
```

PCAP / Packet Capture i IOS-XE

Det är jätteenkelt att skapa en PCAP i en IOS-XE switch. Det finns många filter man kan använda. Här skriver jag hur man gör det per port utan speciella filter. Man är välkommen att fylla på i artikeln.

1. Identifiera vilken port användaren sitter på. (show mac address-table kan väl vara hjälpsam)
2. Definiera namn på monitor session, vilken port, riktning, och filter. Ex. `monitor capture PCAP-NAMN interface gigabitEthernet 3/0/20 both match any`
3. Definiera var PCAP ska sparas med `monitor capture dameware file location flash:PCAP-NAMN.pcap`
4. Starta insamling med `monitor capture PCAP-NAMN start`
5. När du är nöjd avsluta insamling med `monitor capture PCAP-NAMN stop`. **Om du inte vill fylla diskytan på switchen glöm inte detta.**
6. Ladda hem PCAP-fil med SCP och analysera. Det är viktigt att på din klient använda flaggan -O. Pga att du behöver använda scp och inte sftp protokollet.
7. Ta bort pcap fil i flash när du är klar

Verfiering sker med `show monitor capture`. Man kan kolla på en del av insamlingen med `show monitor capture NAMN buffer brief`. Har man inte specifiserat, eller inte kan specifisera, var den ska sparas kan man exportera ut PCAP-en till ex. bootflash med `monitor capture NAMN export bootflash:NAMN.pcap`.

Stacking C9000

Switchar i C9000 serien kan stackas. 9500 stackas med virtual-stackwise, en nyare version av VSS. Mindre switchar, ex. C9300, stackas med stackkablar.

En switchstack kan verifieras med show switch.

För att kontrollera stack-kablarna använd show switch stack-ports. För att se om det finns CRC-fel på stackkablarna använd details.

show switch stack-ports summary - sammanfattning av stack-portarnas status

show switch stack-ports detail - detaljer om CRC-errors m.m

switch x stack port x disable - Stäng av specifik port

switch x stack port x enable - Öppna av specifik port

Network Address Translation (NAT)

NAT är en samling av olika tekniker för att översätta en IP-adress till en annan.

Carrier Grade NAT (GGN)

Traditionella NAT-tabeller sparar outside local/global adresser per flöde för att bestämma den slutgiltiga adressen en host försöker att nå, alltså sparas source-adressen i ett returpaket. CGN sparar inte den här informationen. Det gör att mindre minne används och därmed kan NAT skala bättre. Outside local/global adresser är irrelevanta för en CGN-enhet, då den kan titta på desinationsadressen och port på ett returpaket och mappa det till ett NAT-entry.

När man aktiverar CGN i en IOS-XE router så försvinner möjlighet till outside-mappings, CGN ska alltså enbart användas på enheter där man ej har någon trafik som ska initieras mot en enhet bakom CGN.

```
ip nat settings mode cgn
no ip nat settings support mapping outside
```

Send

Det går att skicka text mellan användare inloggade i en Cisco IOS-XE enhet med send kommandot.

Man kan ange * för att skicka till samtliga. Det går att skicka till specifika användare genom att skicka till deras line-nummer, använder show users för att se vilka line-nummer som användas. Skriva meddelandet. Avsluta med Ctrl+Z. Avbryt med Ctrl+C.

Port-security

Exempelkonfiguration

```
switchport port-security violation restrict
switchport port-security mac-address sticky
switchport port-security mac-address sticky aaaa.bbbb.cccc
switchport port-security aging time 6
switchport port-security aging type inactivity
switchport port-security
```

Sticky MAC

Används sticky-mac får MAC-adressen enbart finnas på en port på switchen. Skulle den dyka upp på en annan port kommer den inte kunna kommunicera. Följande loggrad syns:

`Oct  1 08:58:12: %PORT_SECURITY-2-PSECURE_VIOLATION_MAC_MOVE: Security violation occurred, caused by MAC address aaaa.bbbb.cccc on port GigabitEthernet1/0/46 attempting to access port GigabitEthernet1/0/13`.

kron-jobb

Det går att köra kron-jobb i IOS-XE.

Här är ett bra exempel för att spara konfiguration varje natt:

```
kron occurrence EVERYNIGHT at 23:45 recurring
 policy-list WR-MEM
!
kron policy-list WR-MEM conditional
 cli wr mem
```

Logging

Persistent logging

Man kan spara ned samtlig syslog lokalt. Kan vara bra om en enhet förlorar möjlighet att skicka syslog till en server över upplänken.

Görs med `logging persistent url bootflash:/syslog`.

Proxy ARP

Proxy ARP är en funktion där routern svarar på samtliga ARP-frågor som gäller adresser utanför sitt subnät, om frågan skulle komma. Frågan kan komma om en host är konfigurerad med fel nätmask.

Proxy ARP är aktiverat per default.

```
c6806xl#show ip int vl 1137 | inc Proxy
  Proxy ARP is enabled
  Local Proxy ARP is disabled
```

Avaktiveras genom att konfigurera `no ip proxy-arp` på interfacet.

Local Proxy ARP är en funktion där routern svarar på ARP-frågor även inom ett subnät. Kan vara bra att använda vid ex. private VLAN om man ändå vill tillåta trafik inom ett subnät fast med ACLer. Aktiveras med `ip local-proxy-arp` på interfacet.

WLC 9800

9800-serien är Ciscos IOS-XE WLC. Alltså samma mjukvara för WLCn som för switchar och routrar. Riktigt najs.

Kommandon

Kommando	**Förklaring**
show wireless client <massor av val>	Få information om klienter. Här är några användbara val: • summary - Information om klienter, vilken AP, autentiseringsmetod etc • mac-address aabb.ccdd.eeff detail - Detaljerad information om klienten, sökning på mac-adress
show ap <massor av val>	Information om APor Användbara val: summary - Information om ap-namn, mac-adress, mjukvara etc
wireless client mac-address aaa.bbbb.cccc deauthenticate	Kasta ut klient från det trådlösa väldet

TACACS

VRF-aware tacacs

Enkel konfiguration för att konfa TACACS inom en VRF. Loopback1337 ingår då i den VRFn.

```
aaa group server tacacs+ ISE2T
 server-private 1.2.3.4 single-connection key elite1337
 server-private 1.3.3.7 single-connection key 1337elite
 ip vrf forwarding RESOURCE
 ip tacacs source-interface Loopback1337
```

Om man byter source-interface på TACACS-konfen så ta bort hela och lägg till på nytt. Om man inte gör det kan switchen fortfarande försöka source:a från det tidigare interfacet.

SNMP

SNMPv3

```
    - snmp-server view VIEW_ALL iso included
    - snmp-server group SNMPGROUP v3 priv read VIEW_ALL access ipv6 PERMIT-SNMP-V6 PERMIT-SNMP-V4
    - snmp-server group SNMPGROUP v3 priv context vlan- match prefix read VIEW_ALL access ipv6 PERMIT-SNMP-V6 PERMIT-SNMP-V4
    - "snmp-server user SNMPUSER SNMPGROUP v3 auth sha {{ snmpv3_password1 }} priv aes 128 {{ snmpv3_password2 }}"
    - "snmp-server source-interface traps {{ management_source_interface }}"
```

Lösenorden ska vara 32 tecken långa.

SVI

MAC-adress

Det går att konfigurera vilken MAC-adress en SVI ska använda:

```
interface Vlan1337
 mac-address 1337.c0ff.ee00
```

Byter man MAC-adress skickas automatiskt en GARP och en IPv6 RA med nya MAC-adressen.

SSH-nyckel på Cisco-enheter

Den här konfigurationen har testats mot en C2960CX med mjukvara 15.2(7)E10 och lokalt konto på switchen.

SSH-nyckeln man klistrar in kan inte klistras in på en gång utan behöver delas upp. Jag kollade hur mycket som lyckades klistras in i switchen och fortsatte sedan på nästa radbryt.

```
c2960#conf t
c2960(config)#ip ssh pubkey-chain
c2960(conf-ssh-pubkey)#username jehrlander
c2960(conf-ssh-pubkey-user)#key-string
! Klistra nu in SSH-nyckeln
exit
```

Verifiera konf med show run | sec ip ssh pubkey-chain och testa inloggning.

```
c2960#show run | sec ip ssh pubkey-chain
ip ssh pubkey-chain
  username jehrlander
   key-hash ssh-rsa 421337EE5018FD46DEE13374EA1F5FB marcus@jehrlander.net
```

Jag har utgått från konfigurationen här.