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.

Revision #2
Created 26 April 2025 12:03:15 by Jehrlander
Updated 26 April 2025 12:03:43 by Jehrlander