# Universitatea „Politehnica” din București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

*Ingineria traficului în arhitecturi de MPLS cu Segment-Routing*

Proiect de diplomă

Prezentat ca cerință parțială pentru obținerea titlului de Inginer în domeniul *Electronică și Telecomunicații*

Programul de studii de licență *Tehnologii și sisteme de telecomunicații*

Conducători științifici: Absolvent

*Conf. dr. ing. Marțian Alexandru Dancă Anamaria-Alexandra Ing. Talvan Ion, Orange Romania SA*

*Anul 2022*

*Cuprins*

[Lista figurilor 9](#_bookmark0)

[Lista tabelelor 11](#_bookmark1)

[Lista acronimelor 11](#_bookmark2)

[Introducere 15](#_bookmark3)

1. [Ce este MPLS și care a fost motivația inițială pentru a-l dezvolta? 17](#_bookmark4)
   1. [Beneficiile MPLS 17](#_bookmark5)
   2. [Structura headerului MPLS 19](#_bookmark10)
   3. [Label Stacking 20](#_bookmark12)
   4. [Label Switch Router (LSR) 20](#_bookmark13)
   5. [Label Switched Path (LSP) 20](#_bookmark15)
   6. [Label Distribution Protocol 22](#_bookmark18)
   7. [Label Forwarding Instance Base 22](#_bookmark19)
   8. [Control Plane și Data Plane 23](#_bookmark20)
   9. [MPLS Payload 23](#_bookmark21)
   10. [Redirecționarea pachetelor etichetate 23](#_bookmark22)
   11. [Comportamentul TTL pentru pachetele etichetate 25](#_bookmark26)
   12. [MPLS Traffic Engineering 27](#_bookmark31)
       1. [Prezentare generală a modului de operare MPLS-TE 28](#_bookmark33)
       2. [Rutarea și costul unui TE-LSP 29](#_bookmark34)
       3. [Calcularea căii tunelului TE 29](#_bookmark35)
       4. [RSVP – TE 30](#_bookmark36)
2. [Segment-Routing MPLS 33](#_bookmark38)
   1. [Beneficiile Segment-Routing 33](#_bookmark39)
   2. [Noțiuni de bază 34](#_bookmark40)
   3. [Tipuri de SID 34](#_bookmark41)
   4. [Mecanism de redirecționare SR 36](#_bookmark47)
3. [Segment-Routing IPv6 39](#_bookmark50)
4. [Implementare 43](#_bookmark55)
   1. [MPLS – Traffic Engineering: Link Protection, Node Protection 43](#_bookmark56)
   2. [MPLS – Traffic Engineering: MPLS-VPN 48](#_bookmark69)
   3. [MPLS – Traffic Engineering : MPLS-TE si MPLS VPN 53](#_bookmark80)
   4. [Segment-routing MPLS-Traffic Engineering: Topology-Independent Loop-Free Alternate (TI-LFA).57](#_bookmark91) [4.5 Segment-routing MPLS-Traffic Engineering: Policy 62](#_bookmark101)
   5. [Segment-routing MPLS-Traffic Engineering: Path Computation Element 65](#_bookmark108)

[Concluzii 71](#_TOC_250000)

[Bibliografie 73](#_bookmark116)

## Lista figurilor

[Figură 1. 1. Beneficiile MPLS [3] 17](#_bookmark6)

[Figură 1. 2. BGP-Free în rețea MPLS [3] 18](#_bookmark7)

[Figură 1. 3. Modelul Overlay [3] 18](#_bookmark8)

[Figură 1. 4. Modelul Peer-to-Peer [3] 19](#_bookmark9)

[Figură 1. 5. Structură Header [3] 19](#_bookmark11)

[Figură 1. 6. Label Stack [3] 20](#_bookmark14)

[Figură 1. 7. Label Switched Path [3] 21](#_bookmark16)

[Figură 1. 8. LSP Imbricat (nested LSP) [3] 21](#_bookmark17)

[Figură 1. 9. Operațiuni cu etichete [3] 23](#_bookmark23)

[Figură 1. 10. CEF și LFIB Lookup [3] 24](#_bookmark24)

[Figură 1. 11. Eliminare etichetă [3] 25](#_bookmark25)

[Figură 1. 12. Mecanismul de propagare TTL între header-ul de IP și etichetele MPLS [3] 26](#_bookmark27)

[Figură 1. 13. Mecanismul de propagare TTL în cazul operațiunilor Swap, Push și Pop [3] 26](#_bookmark28)

[Figură 1. 14. Mesajul ICMP "Time Exceeded" [3] 26](#_bookmark29)

[Figură 1. 15. Mesajul ICMP "Time Exceeded" în Rețeaua MPLS [3] 27](#_bookmark30)

[Figură 1. 16. MPLS TE Head End Router [3] 27](#_bookmark32)

[Figură 1. 17. Stilul RSVP SE[3] 31](#_bookmark37)

[Figură 2. 1. Tipuri de SID [9] 34](#_bookmark43)

[Figură 2. 2. Prefix SID-bazat pe căile de redirecționare [9] 35](#_bookmark44)

[Figură 2. 3. Adjacency SID- bazat pe căile de redirecționare [9] 35](#_bookmark45)

[Figură 2. 4. Adjacency SID + Node SID - bazat pe căile de redirecționare [9] 35](#_bookmark46)

[Figură 2. 5. SR Policy în Segment-Routing [11] 37](#_bookmark48)

[Figură 2. 6. Algoritmul RSVP-TE versus Algoritmul SR nativ [10] 38](#_bookmark49)

[Figură 3. 1. Beneficiu SRv6 [12] 39](#_bookmark51)

[Figură 3. 2. Redirecționarea pachetului SRv6 [12] 40](#_bookmark52)

[Figură 4. 1. Topologia de rețea utilizată pentru exemplificarea MPLS – Traffic Engineering 43](#_bookmark57)

[Figură 4. 2. Secvență din configurația echipamentelor R2 și R4 44](#_bookmark58)

[Figură 4. 3. Tabela de redirecționare MPLS 44](#_bookmark59)

[Figură 4. 4. Configurația Tunelului 1 45](#_bookmark60)

[Figură 4. 5. Starea Tunelului T1 45](#_bookmark61)

[Figură 4. 6. Backup Tunnel 100 - Link Protection 46](#_bookmark62)

[Figură 4. 7. Calea tunelului T1 46](#_bookmark63)

[Figură 4. 8. Configurația Tunelului de backup 100 47](#_bookmark64)

[Figură 4. 9. Trecerea de la starea "ready" la "active" odată cu eșuarea link-ului protejat 47](#_bookmark65)

[Figură 4. 10. Calea de backup - T100 – Link Protection 47](#_bookmark66)

[Figură 4. 11. Backup Tunnel 101 - Node Protection 48](#_bookmark67)

[Figură 4. 12. Configurația Tunelului de backup 101 42](#_bookmark68)

[Figură 4. 13. Calea de backup - T101 - Node Protection 48](#_bookmark68)

[Figură 4. 14. MPLS VPN 49](#_bookmark70)

[Figură 4. 15. Interfețele cu rol de redirecționare a traficului din VRF-uri 50](#_bookmark71)

[Figură 4. 16. Configurația routerelor PE1 și PE2 – secțiunea vrf 51](#_bookmark72)

[Figură 4. 17. Rutele pentru VPN afișate în tabela de redirecționare MPLS 51](#_bookmark73)

[Figură 4. 18. Tabelele de rutare ale clienților A și B 52](#_bookmark74)

[Figură 4.19. Conectivitate între site-urile clientului A 52](#_bookmark75)

[Figură 4. 20. Conectivitate între site-urile clientului B 52](#_bookmark76)

[Figură 4. 21. Lipsa conectivității dispozitivelor ce aparțin clienților diferiți 52](#_bookmark77)

[Figură 4. 22. Calea de comunicație între site-urile clientului A 52](#_bookmark78)

[Figură 4. 23. Calea de comunicație între site-urile clientului B 52](#_bookmark79)

[Figură 4. 24. Ilustrarea tunelurilor T10 si T11 54](#_bookmark81)

[Figură 4. 25. Calea de transmitere a traficului prin T10, respectiv T11 54](#_bookmark82)

[Figură 4. 26. Ilustrarea tunelurilor de backup – T20, T21, T22, T23 55](#_bookmark83)

[Figură 4. 27. Informații privind tunelurile de backup T20 și T22 ale routerelor PE1 și PE2 55](#_bookmark84)

[Figură 4. 28.Calea de transmitere a traficului prin tunelurile T10 și T11 într-un scenariu de bună funcționalitate 55](#_bookmark85)

[Figură 4. 29.Calea de transmitere a traficului prin tunelurile de backup T20 și T22 în cazul defecțiunii unui link.55](#_bookmark86) [Figură 4. 30. Ilustrarea Tunelului T2 55](#_bookmark87)

[Figură 4. 31. Configurația Tunelului T2 50](#_bookmark88)

[Figură 4. 32. Starea Tunelului T2 56](#_bookmark88)

[Figură 4. 33. Calea de transmisie a traficului de-a lungulul tunelului T2 56](#_bookmark89)

[Figură 4. 34. Tabela FIB – Tunnel 2 56](#_bookmark90)

[Figură 4. 35. Topologia de rețea utilizată pentru exemplificare Segment-Routing MPLS Traffic-Engineering 58](#_bookmark92)

[Figură 4. 36. Configurație Procotol IS-IS 59](#_bookmark93)

[Figură 4. 37. Calea de transmitere a traficului de la xr1 către xr6 59](#_bookmark94)

[Figură 4. 38. Apariția unei căi de backup după activarea TI-LFA 60](#_bookmark95)

[Figură 4. 39. Apariția căilor de backup în tabela de redirecționare mpls după activarea TI-LFA 60](#_bookmark96)

[Figură 4. 40. Vizualizarea etichetelor impuse 60](#_bookmark97)

[Figură 4. 41. Protection coverage 60](#_bookmark98)

[Figură 4. 42. Ilustrarea unei căi post-convergență în cazul unor costuri diferite pe link-uri 61](#_bookmark99)

[Figură 4. 43. Informații privind calea de backup TI-LFA în cazul unor costuri diferite pe link-uri 61](#_bookmark100)

[Figură 4. 44. Ilustrarea unor SR-Policy configurate explicit și dinamic 63](#_bookmark102)

[Figură 4. 45. Configurația SR-Policy 63](#_bookmark103)

[Figură 4. 46. Interfețele lui xr1 59](#_bookmark104)

[Figură 4. 47. MPLS Forwarding Table 64](#_bookmark104)

[Figură 4. 48. Caracteristicile Politicilor SR configurate 64](#_bookmark105)

[Figură 4. 49. MPLS Forwarding Table după modificarea costului interfeței Gi0/0/0/3 64](#_bookmark106)

[Figură 4. 50. Caracteristicile SR Policy 1 după modificarea costului interfeței Gi0/0/0/3 65](#_bookmark107)

[Figură 4. 51. Topologia de rețea controlată de SR-PCE 67](#_bookmark109)

[Figură 4. 52. Configurația pentru dispozitivele PCE și PCC 67](#_bookmark110)

[Figură 4. 53. PCE’s peer database 67](#_bookmark111)

[Figură 4. 54. PCC’s peer database 68](#_bookmark112)

[Figură 4. 55. Informații topologice deținute de SR-PCE 68](#_bookmark113)

[Figură 4. 56. Informații din baza de date topologică a routerului SR-PCE 69](#_bookmark114)

[Figură 4. 57. Informații din baza de date topologică a routerului SR-PCE referitoare la politicile SR configurate 70](#_bookmark115)

## Lista tabelelor

[Tabel 2. 1. Tipuri de SID – mod de configurare și utilizare [9] 34](#_bookmark42)

[Tabel 3. 1 Diferențe între modurile de funcționare SRv6 [12] 41](#_bookmark53)

[Tabel 3. 2. Diferențe între implementările SR-MPLS și SRv6 [12] 41](#_bookmark54)

## Lista acronimelor

|  |  |
| --- | --- |
| *Acronim* | *Denumire în engleză* |
| *ASN* | Autonomous System Number |
| *ATM* | Asynchronous Transfrer Mode |
| *AToM* | Any Transport over MPLS |
| *BGP* | Border Gateway Protocol |
| *BGP-LS* | Border Gateway Protocol Link-State |
| *BSID* | Binding Segment ID |
| *CE* | Customer Edge Router |
| *CSPF* | Constrained Shortest Path First |
| *eBGP* | External Border Gateway Protocol |
| *ECMP* | Equal-Cost Multi-Path routing |
| *EGP* | Exterior Gateway Protocol |
| *EIGRP* | Enhanced Interior Gateway Routing  Protocol |
| *E-LSR* | Edge Label Switch Router |
| *ERO* | Explicit Route Object |
| *EVPN* | (Ethernet VPN) |
| *FEC* | Forwarding Equivalence Class |
| *FIB* | Forwarding Information Base |
| *FR* | Frame Relay |
| *FRR* | Fast Reroute |
| *GPS* | Global Positioning System |
| *iBGP* | Internal Border Gateway Protocol |
| *ICMP* | Internet Control Message Protocol |
| *IGP* | Interior Gateway Protocol |
| *ILM* | Incoming Label Map |
| *IP* | Internet Protocol |
| *IPv4* | Internet Protocol version 4 |
| *IPv4* | Internet Protocol version 6 |
| *IS-IS* | Intermediate Systems to Intermediate  Systems |
| *ISP* | Internet Service Provider |
| *LDP* | Label Distribution Protocol |
| *LFA* | Loop-Free Alternante |
| *LFIB* | Label Forwarding Instance Base |
| *LIB* | Label Information Base |

|  |  |
| --- | --- |
| *LSP* | Label Switched Path |
| *LSR* | Label Switch Router |
| *MP* | Merge Point |
| *MP-BGP* | Multiprotocol Border Gateway Protocol |
| *MPLS* | Multi Protocol Label Switching |
| *MPLS-TE* | MPLS Traffic Engineering |
| *MPLS-VPN* | Multi Protocol Label Switching -Vitual  Private Networks |
| *NHFLE* | Next Hop Label Forwarding Entry |
| *NHOP* | Next-Hop Bypass Tunnel |
| *NLPI* | Network Level Protocol Identifier |
| *NNHOP* | Next-Next-Hop Bypass Tunnel |
| *OaM* | Operation and Maintenance |
| *OSI* | Open Systems Interconnection |
| *OSPF* | Open Shortest Path First |
| *P* | Provider Router |
| *PCALC* | Path Calculation |
| *PCC* | Path Computation Client |
| *PCE* | Path Computation Element |
| *PCEP* | Path Computation Element Protocol |
| *PE* | Provider Edge Router |
| *PLR* | Point of Local Repair |
| *PPP* | Point to Point Protocol |
| *QoS* | Quality of Service |
| *RD* | Route Distinguisher |
| *RIB* | Routing Information Base |
| *RLFA* | Remote Loop-Free Alternante |
| *RRR* | Routing with Resource Reservation |
| *RSVP* | Resource Reservation Protocol |
| *RSVP-TE* | Resource Reservation Protocol - Traffic  Engineering |
| *RT* | Route Target |
| *SDN* | Software Define Network |
| *SE-RSVP* | Shared Explicit Resource Reservation  Protocol |
| *SID* | Segment ID |
| *SLA* | Service-Level Agreement |
| *SPF* | Shortest Path First |
| *SR* | Segment-Routing |
| *SRD* | Segment Routing Domain |
| *SRGB* | Segment Routing Global Block |
| *SRH* | Segment-Routing Header |
| *SRLB* | Segment Routing Local Block |
| *SRLGs* | Shared Risk Link Groups |
| *SR-MPLS* | Segment Routing-Multi Protocol Label  Switching |
| *SR-PCE* | Segment Routing -Path Computation  Element |

|  |  |
| --- | --- |
| *SR-TE* | Segment Routing Traffic Engineering |
| *SRv6* | Segment-Routing IPv6 |
| *SRv6-BE* | SRv6 Best Effort |
| *TCP* | Transmission Control Protocol |
| *TDP* | Tag Distribution Protocol |
| *TE-LSP* | Traffic Engineering - Label Switched Path |
| *TED* | Traffic Engineering Database |
| *TI-LFA* | Topology-Independent Loop-Free  Alternate |
| *TTL* | Time To Live |
| *VPN* | Virtual Private Network |
| *VRF* | Virtual Routing Forwarding |
| *WAN* | Wide Area Network |
| *XTC* | XR Traffic Controller |

## Introducere

Nevoia de a primi și transmite informații a reprezentat motivul ce a stat la baza perfecționării modului prin care oamenii pot comunica. Comunicarea online a devenit semnificativ mai importantă în ziua de azi pentru întreaga lume. Schimbul rapid de informații între diferiți destinatari a devenit posibil cu ajutorul răspândirii internetului și dezvoltării facilităților pe care acesta le oferă. Internetul oferă o mare varietate de beneficii cum ar fi: transmiterea unui mesaj text, trimiterea unui e-mail, conferințe video, accesul la informație, capabilități de dezvoltare, GPS, servicii bancare, cumpărături, work from home și multe altele.

Orice informație comunicată online este livrată destinatarului aproape instantaneu. Procesul de comunicare și modul în care aceasta se realizează este personalizabil ținând cont de nevoile și cerințele fiecărui utilizator. Nu este nicio îndoială cu privire la rolul vital și esențial pe care comunicarea online îl are în procesul de apropiere al oamenilor, comunităților și societății în general. Granițele de timp și distanță au devenit irelevante și nesemntificative mulțumită comunicării online. Aceasta susține aproape fiecare domeniu al industriei și afacerilor, fie că este vorba de artă și cultură, sport, medicină, agricultură sau chiar politică, comunicarea online ajută foarte mult la luarea mai rapidă a deciziilor și îmbunătățirea productivității pe toate planurile. [1]

Pentru a îndeplini cerințele societății și pentru a crea conectivitatea dintre dispozitive s-au dezvoltat diferite tehnologii de transport a informației. La cel mai simplu nivel, un router stabilește comunicații de baza prin interfață, se bazează pe o metodă de adresare care identifică în mod unic fiecare dispozitiv și organizează dispozitivele individuale într-o structură de rețea. Acestea sunt esențiale pentru a îmbunătăți capacitățile rețelelor, sunt considerate punct de control principal în mediul de rețea, deținând responsabilitatea pentru calitatea și securitatea serviciului, monitorizare și eficiență. Creșterea constantă a cerințelor, nevoia de viteze mai mari și de extindere a rețelelor locale au determinat dezvoltarea de noi tehnologii de trasport pentru atingerea unor noi niveluri de performanță. [2]

## Scopul lucrării

Pornind de la ideea dezvoltării tehnologiilor de trasport a informației, lucrarea își propune să analizeze protocolul MPLS (Multi Protocol Label Switching), implementarea acestuia în rețelele tradiționale și implementarea cu Segment-Routing. Se vor pune în evidență diferențele între cele două metode de implementare, avantajele și dezavantajele acestora.

Pentru a contribui la realizarea obiectivului și la o ușoară înțelegere, lucrarea este structurată în 3 capitole:

* + 1. Capitolul 1 conține noțiunile care stau la baza întregii lucrări și anume: protocolul MPLS și elementele sale componente, analiza avantajelor acestuia și noțiuni de inginerie de trafic.
    2. Capitolul 2 prezintă informații despre implementarea Segment-Routing pe un plan de date MPLS, noțiunile de bază și mecanismul de redirecționare.
    3. Capitolul 3 cuprinde aspectele generale care stau la baza implementării Segment-Routing pe baza planului IPv6, diferențe între SR-MPLS și SR-IPv6, mecanism de redirecționare.
    4. Capitolul 4 prezintă implementarea propriu-zisă a unor topologii de rețea simulate în mediul EVE-NG având ca scop punerea în evidență a caracteristicilor implementării tradiționale MPLS și MPLS Segment-Routing.

## 1. Ce este MPLS și care a fost motivația inițială pentru a-l dezvolta?

Multi Protocol Label Switching este un mecanism eficient de încapsulare, care utilizează etichete,

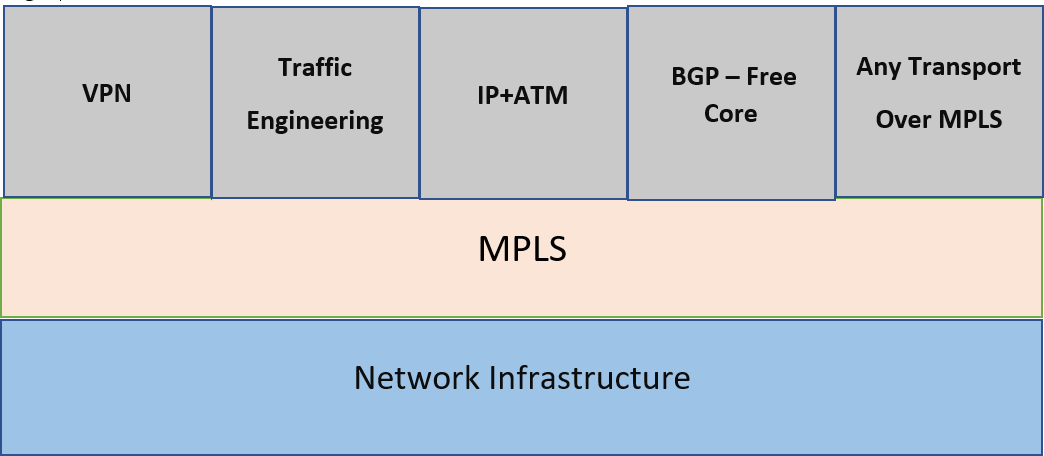
„labels”, atașate pachetelor IP pentru transportul datelor de la un nod la altul într-o rețea de telecomunicații. Acesta oferă un mecanism foarte scalabil care facilitează topologia, permițând routerelor de bază să comute pachete pe baza unor headere simplificate.

Tehnica comutării etichetelor nu este nouă întrucât Frame Relay și ATM au utilizat-o pentru a muta frame-uri și celule de-a lungul rețelei. O diferență dintre acestea două este aceea că FR permite orice lungime pentru dimensiunea frame-urilor, în timp ce ATM impune o lungime fixă celulei (header de 5 bytes și sarcină utilă de 48 de bytes). Asemănarea dintre FR și ATM este aceea că valoarea etichetei se schimbă la fiecare hop din rețea, spre deosebire de rutarea pachetelor IP unde routerul nu schimbă adresa destinație atunci când primește un pachet, aceasta rămânând aceeași de-a lungul rețelei. Și în MPLS decizia de redirecționare a pachetelor se ia doar pe baza conținutului acestei etichete, fără a fi necesară examinarea pachetului în sine, astfel încât principalul beneficiu este reprezentat de accelerarea fluxurilor de trafic și eliminarea necesității interogărilor complexe din tabelele de rutare.

Înainte de MPLS, cele mai populare protocoale WAN (Wide Area Network) erau ATM (Asynchronous Transfrer Mode) și Frame Relay. Rețelele WAN au fost eficiente din punct de vedere al costurilor, fiind capabile să transporte diferite protocoale. Rețelele private virtuale , Virtual Private Network (VPN), au fost create prin închirierea link-urilor de ATM/FR și sunt cunoscute și sub numele de overlay networks (rețele suprapuse), iar în ziua de azi mulți dintre clienți folosesc serviciile MPLS VPN pentru beneficiile pe care acesta le oferă. [3]

### Beneficiile MPLS

MPLS oferă mai multe beneficii dintre care putem enumera:



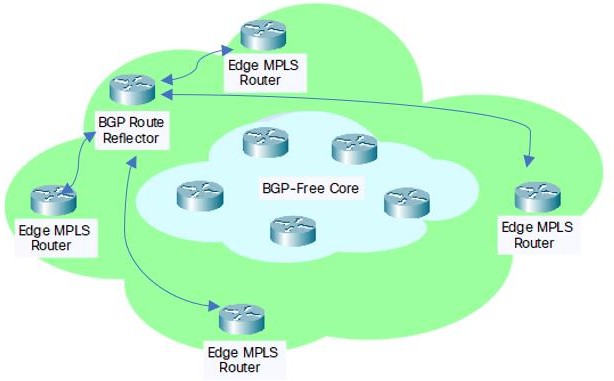
*Figură 1. 1. Beneficiile MPLS [3]*

* + - *Utilizarea unei singure infrastructuri de rețea unificate*

Unul dintre motivele pentru care IP a devenit cel mai utilizat protocol în întreaga lume este acela că poate transporta multe tehnologii, nu doar date, ci și telefonie. Utilizând MPLS cu IP se pot extinde posibilitățile a ceea ce putem transporta prin rețea. MPLS este denumit „multiprotocol” deoarece poate transporta mai multe tehnolgii precum: IPv4, IPv6, Ethernet, Point to Point Protocol (PPP) și alte tehnologii de Layer 2. Caracteristica prin care orice frame de Layer 2 este transportat peste MPLS backbone se numește Any Transport over MPLS (AToM). Nu este nevoie ca routerele care transportă traficul AToM să cunoască sarcina utilă, ele au nevoie doar de informația etichetei pentru a redirecționa pachetele. În esență, MPLS oferă o metodă simplă de comutare a mai multor protocoale într-o singură rețea. [3]

* + - *BGP-Free Core*

O rețea de service provider trebuie să redirecționeze traficul și către destinații care sunt externe acesteia, astfel prefixele IP externe trebuie să existe în tabela de rutare a fiecărui router, astfel toate routerele din rețeaua furnizorului de servicii trebuie să ruleze BGP (Border Gateway Protocol). Un ISP (Internet Service Provider) care are 200 de routere în rețeaua de bază (core network) trebuie să ruleze BGP pe toate acestea. Daca MPLS este implementat în rețea, numai routerele de margine (edge routers), să presupunem 50 de routere, trebuie să ruleze BGP, cele din nucleul rețelei transmițând pachete etichetate, fiind scutite de sarcina rulării BGP și având nevoie de mai puțină memorie.

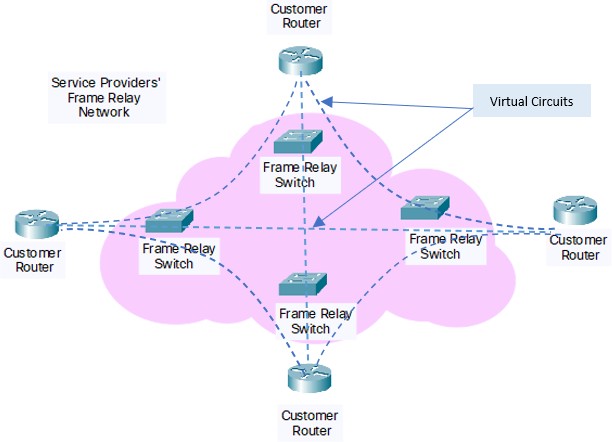


*Figură 1. 2. BGP-Free în rețea MPLS [3]*

* *Modelele peer-to-peer și overlay pentru MPLS VPN*

Când vorbim de VPN (Virtual Private Network) ne referim la o rețea privată într-o infrastructură comună unde toate site-urile clienților să se poată interconecta și să fie complet separate de alte VPN-uri. De obicei, VPN-ul aparține unei singure companii și are mai multe site-uri interconectate. Furnizorii de servicii pot implementa două modele majore de VPN pentru a oferi servicii clienților: modelul overlay VPN și modelul peer-to-peer.

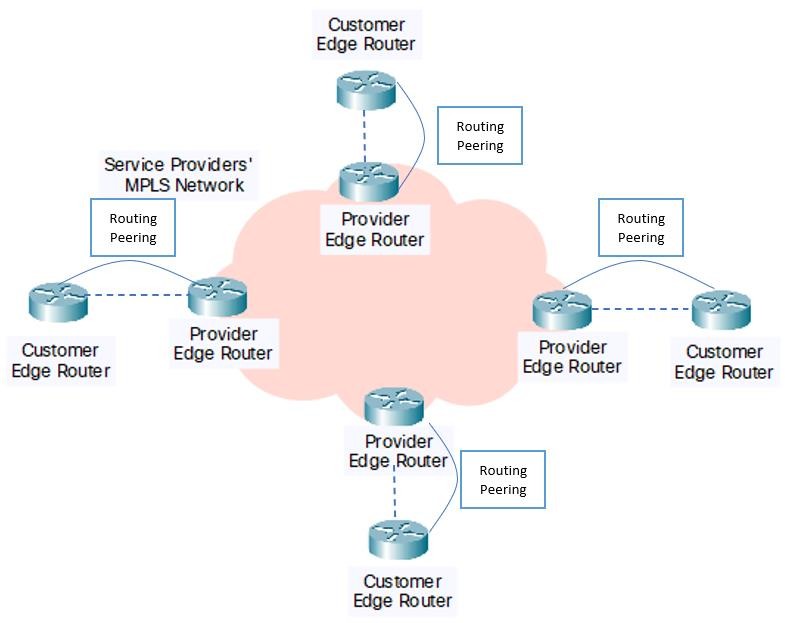
În modelul overlay VPN furnizorul oferă link-uri point-to-point sau circuite virtuale prin rețea între routerele clienților. Routerele clienților formează legături între ele direct prin link-urile point-to- point sau prin circuitele virtuale, nu se formează niciun peering de rutare între un router client și un router de service provider. Astfel, routerele de service provider nu văd niciodată rutele clienților. [3]



*Figură 1. 3. Modelul Overlay [3]*

În modelul peer-to-peer VPN routerele de service provider transportă datele clienților pe întreaga rețea și participă și la rutare. Cu alte cuvinte, routerul de service provider se conectează direct cu routerele

clienților. Înainte de apariția MPLS, overlay VPN era implementat mult mai frecvent deoarece VPN peer- to-peer necesita multe resurse și modificări de configurare la multe site-uri. MPLS VPN a făcut modelul peer-to-peer mult mai ușor de utilizat, eliminând necesitatea creării multor circuite virtuale sau configurării unor filtre de pachete, astfel încât să fie păstrată confidentialitatea între VPN-uri. MPLS asigură această confidențialitate utilizând conceptul de VRF, Virtual Routing Forwarding, informațiile de rutare de la diferiți clienți fiind păstrate separat, iar pachetele redirecționate doar pe baza informațiilor de etichetă, nu și a celor din headerul de IP. În MPLS VPN router-ul client este numit și Customer Edge Router (CE), iar router-ul de service provider este numit Provider Edge Router (PE).



*Figură 1. 4. Modelul Peer-to-Peer [3]*

Așadar, comparând Figurile 1.3 și 1.4 , putem observa că în primul caz, figura 1.3, fiecare CE se conectează cu n-1 alte CE-uri, considerând n numărul total de CE-uri, în timp ce în figura 1.4 fiecare router CE se conectează doar cu un singur PE. Astfel, furnizorul de servicii trebuie să furnizeze doar legătura dintre PE și routerele CE. Utilizând MPLS VPN traficul circulă direct, optim, între toate site- urile clienților întrucât acestea sunt interconectate, formează o rețea completă (fully meshed design of links). [3]

* + Ingineria traficului

Ideea care stă la baza ingineriei traficului este utilizarea optimă a infrastructurii de rețea, inclusiv link-urile care sunt subutilizate deoarece nu se află pe calea preferată de redirecționare. Protocoalele de rutare dinamică aleg cea mai bună cale ca fiind cea cu costul cel mai mic. Implementând ingineria traficului în rețeaua IP/MPLS se asigură o fluidizare a traficului și se evită congestiile de rețea. În capitolul „MPLS Traffic Engineering” vom prezenta mai în amănunt acest aspect.

### A picture containing rectangle Description automatically generatedStructura headerului MPLS

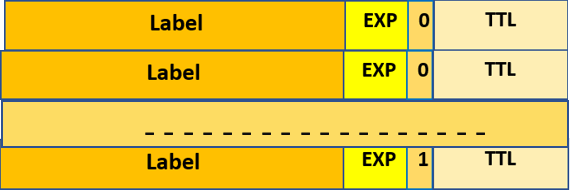
*Figură 1. 5. Structură Header [3]*

În Figura 1.5 avem reprezentarea headerului de MPLS. Eticheta MPLS este alcătuită din 32 de biți. Primii 20 de biți sunt identificatori pentru valoarea etichetei, următorii 3 biți sunt experimentali sau

pot reprezenta Class of Service, fiind utilizați pentru calitatea serviciului (QoS - Quality of Service). Bitul 24 reprezintă Bottom of Stack, acest bit ia valoarea 1 numai atunci când eticheta se află înaintea headerului de IP, altfel acesta ia valoarea 0. Atunci când pachetele dețin mai multe anteturi MPLS, acest bit permite unui LSR să recunoască ultimul header MPLS înainte de header-ul IP. Ultimii 8 biți sunt utilizați pentru TTL, Time to Live, având aceeași semnificație cu câmpul TTL din headerul de IP. Valoarea TTL indică perioada de timp în care un pachet poate exista în cadrul unei rețele, se decrementează cu 1 la fiecare hop, evitându-se astfel buclele ce pot fi formate în rețea din cauza livrării la infinit a unui pachet între routere. Atunci când valoarea acestui câmp devine 0, pachetul este eliminat din rețea. [3]

### Label Stacking

Un router compatibil cu MPLS (LSR - Label Switch Router) ar putea avea nevoie de mai multe etichete pentru a ruta un pachet în rețea, de aceea a fost necesară gruparea etichetelor într-o stivă. Prima etichetă din stivă poartă numele de „top label” , iar ultima de „bottom label”. Se poate observa din Figura

1.6 că bitul S are valoarea 0 pentru toate etichetele, cu excepția ultimei etichete unde valoarea este 1, fiind ultima din stivă.

### Label Switch Router (LSR)

*Figură 1. 6. Label Stack [3]*

Un LSR este un router specific MPLS, care efectuează rutarea numai pe baza etichetei. Când un LSR primește un pachet, folosește label-ul inclus în antet pentru a lua decizia cu privire la next-hop. Înainte ca pachetul să fie redirecționat, vechea etichetă va fi îndepărtată din antet și înlocuită cu una nouă care va oferi noi informații pentru redirecționare.

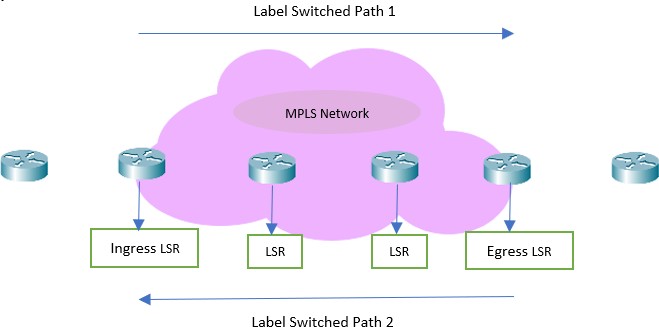
Există mai multe tipuri de LSR într-o rețea MPLS:

* + 1. Edge LSR (E-LSR) – este routerul care procesează atât pachete etichetate, cât și neetichetate
    2. LSR de intrare (Ingress LSR) – este routerul care primește un pachet neetichetat și care înserează o stivă de etichete în fața header-ului de IP, este considerat și un punct de plecare pentru o anumită cale LSP (Label Switched Path)
    3. LSR de ieșire (Egress LSR) - este routerul care primește pachetele etichetate, elimină eticheta și trimite mai departe pe legătura de date pachetul respectiv
    4. LSR intermediar – este routerul care efectuează schimbarea etichetelor (label swapping)

*Observație:* Un router de intrare LSR poate fi și un router de ieșire/ intermediar pentru orice alte LSP-uri. Așadar, rolul LSR-urilor este specific fiecărui LSP. [3]

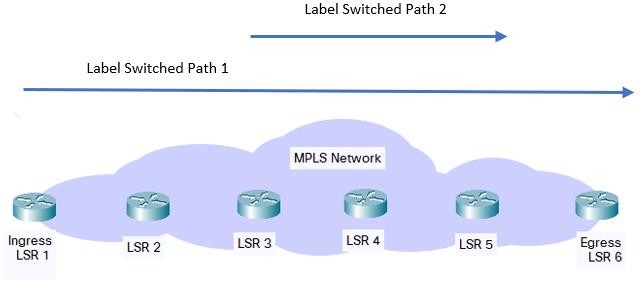
### Label Switched Path (LSP)

Când vorbim de LSP ne referim la calea prin rețeaua MPLS pe care o parcurg pachetele comutate de LSR-uri sau la o parte din această cale. În figura 1.7 avem o reprezentare a unei rețele MPLS, săgeata indicând calea LSP. Fluxul de pachete etichetate în direcția opusă, de la dreapta spre stânga, care se trimit între aceleași Edge LSR-uri, reprezintă o altă cale LSP.



*Figură 1. 7. Label Switched Path [3]*

De asemenea, routerul de intrare al unei LSP nu trebuie neapărat să fie routerul care etichetează pentru prima dată un pachet. Așa cum se poate observa și din figura 1.8, LSP1 se întinde pe toată lățimea rețelei MPLS, iar routerul 1 este cel care adaugă prima etichetă pachetului. LSP2 începe de la router 3 și se termină la router 5. Astfel, atunci când pachetul ajunge la router 3, va intra în LSP2 și totodată va primi o nouă etichetă. În acest moment putem spune că este vorba despre un „Nested LSP”. Apare și fenomenul de Label Stacking pe care l-am menționat anterior, iar stiva conține în acest moment două etichete: top label este corespunzător pentru LSP2, iar bottom label pentru LSP1.



*Figură 1. 8. LSP Imbricat (nested LSP) [3]*

Așa cum am menționat anterior, LSR de intrare adaugă prima etichetă pachetului în stivă (imposes the bottom label), urmând că top label să fie schimbat la fiecare hop din rețea. LSR-urile intermediare au funcția de swapping, acestea schimbă top label-ul, numit și incoming label, cu o altă etichetă (outgoing label), transmițând pachetul pe outgoing link. În final, LSR de ieșire elimină eticheta (pops the label) și trimite pachetul fară nicio etichetă pe link-ul de ieșire. Ca acest proces să se desfășoare cu succes a fost nevoie de un mecanism prin care routerele să își transmită informații despre asocierea etichetelor cu ficare prefix IGP (Interior Gateway Protocol -OSPF, IS-IS, EIGRP) pentru a duce la bun sfârșit redirecționarea pachetelor. Este important să menționam că aceste etichete sunt locale pentru fiecare pereche de routere adiacente, etichetele nu au un înțeles global peste întreaga rețea. Așadar, a apărut nevoia unui protocol de distribuție a etichetelor (LDP - Label Distribution Protocol). [3]

Există două metode prin care se poate realiza distribuția etichetelor:

1. Atașând etichetele unui protocol IP deja existent
2. Implementând un protocol de distribuție a etichetelor separat

Vom analiza cele două variante:

1. Această primă metodă prezintă un avantaj și anume acela că nu este nevoie de un nou protocol pentru a rula pe LSR-uri, dar un dezavantaj privind faptul că fiecare protocol IP trebuie extins astfel încât să suporte operațiunile necesare de imposing, swapping și popping, iar acest lucru nu este tocmai simplu. De asemenea, această metodă ar determina ca tabela de rutare și distribuția de etichete să fie în

sincronizare permanentă, ceea ce ar însemna că nu putem avea o etichetă în cazul unui prefix lipsă și invers.

Pentru protocoalele distance-vector, cum ar fi EIGRP, mecanismul este simplu, fiecare router generează un prefix în tabela sa de rutare căruia îi va corespunde o etichetă. Pentru protocoalele link- state, cum ar fi OSPF/ IS-IS, mecanismul este dificil din cauza faptului că fiecare router generează link- state updates, mesaje de update prin care acestea își comunică rutele. O arhitectură MPLS are nevoie ca fiecare router să asocieze corect eticheta cu prefixul IGP, chiar dacă routerul respectiv nu a generat acel prefix. Este nevoie de o îmbunatățire mare pentru ca protocoalele link-state să poată ajunge la aceste performanțe. [3]

Border Gateway Protocol (BGP) este un protocol de tipul EGP (Exterior Gateway Protocol) care poate suporta atât prefixe, cât și etichete în același timp. Este folosit în principal pentru rețelele virtuale private, MPLS VPN.

1. Această metoadă prezintă marele avantaj că este independentă de protocolul de rutare. Indiferent de ce protocol de rutare se utilizează, distribuția etichetelor se face separat de distribuția prefixelor. Există 3 tipuri de protocoale de distribuție a etichetelor:

i. TDP – Tag Distribution Protocol – proprietar Cisco

îi. LDP – Label Distribution Protocol - cel mai des utilizat

iii. RSVP – Resource Reservation Protocol – utilizat pentru MPLS-TE

### Label Distribution Protocol

Pentru fiecare prefix IGP din tabela de rutare, fiecare LSR creează o legătură locală între prefix și etichetă, urmând ca mai apoi să trimită aceste legături tuturor vecinilor săi. Toate aceste legături primite (received bindings) devin legături la distanță (remote bindings). Aceste corespondențe prefix-etichetă sunt stocate într-o tabelă specială numită Label Information Base (LIB). Tabela de rutare oferă informații despre adresa IPv4 a next-hop-ului. [3]

LDP îndeplinește 4 funcționalități majore:

* Descoperirea LSR-urilor care rulează LDP
* Stabilirea și întreținerea sesiunii
* Anunță corespondențele etichetelor
* Transmiterea mesajelor de notificare pentru a semnala evenimente semnificative pentru peer-ul LDP

Când două LSR-uri rulează LDP și partajează una sau mai multe legături între ele, se descoperă reciproc prin mesaje „Hello”. Următorul pas este să stabilească o sesiune printr-o conexiune TCP (Transmission Control Protocol) , astfel încât LDP să anunțe corespondența dintre prefix și etichetă între cele două routere.

### Label Forwarding Instance Base

LFIB este o tabelă folosită pentru a redirecționa pachetele etichetate. Această tabelă este populată cu etichetele de intrare și de ieșire pentru un prefix IP având în vedere cea mai bună cale. Eticheta de intrare este cea care provine din legătura locală a unui LSR specific, iar eticheta de ieșire este cea care provine de la o legătură la distanță aleasă de LSR dintre legăturile disponibile pentru un anumit prefix. LIB (Label Information Base) este tabela ce menține corespondența dintre prefixul IP și etichetele de la toți vecinii. În cazul ingineriei de trafic MPLS, etichetele sunt distribuite de RSVP, iar în cazul MPLS- VPN de BGP. [3]

### Control Plane și Data Plane

Arhitectura MPLS cuprinde 2 componente:

1. Control Plane : planul de control care generează și menține informații de rutare și etichetare Aici întâlnim :
   * RIB (Routing Information Base): generată cu ajutorul protocoalelor de rutare
   * LDP (Label Distribution Protocol): alocă etichete, creează LIB și stabilește LSP
   * LIB (Label Information Base): generată de LDP și utilizată pentru controlul etichetelor [4]
2. Data Plane: se mai numește și Forwarding Plane și are rolul de redirecționare a pachetelor IP și a pachetelor MPLS

Aici întâlnim:

* + FIB (Forwarding Information Base): generată pe baza informațiilor de rutare obținute din RIB și e utilizată pentru redirecționarea pachetelor IP
  + LFIB (Label Forwarding Information Base): creată de LDP pe un LSR și utilizată pentru redirecționarea pachetelor MPLS; conține două tabele secundare: ILM (Incoming Label Map) și NHFLE (Next Hop Label Forwarding Entry)

NHFLE e utilizată pentru a redirecționa pachetele MPLS. Un NHFLE specifică informații precum: ID-ul tunelului, interfață de ieșire (outbound interface), next hop, eticheta de ieșire (outgoing label), operațiunea etichetei (label operation). ILM este utilizat pentru a face corespondența între fiecare etichetă de intrare (incoming label) și un grup de NHLFE. ILM specifică ID-ul tunelului, interfața de intrare (inbound interface), eticheta de intrare (incoming label) și label operation. [4]

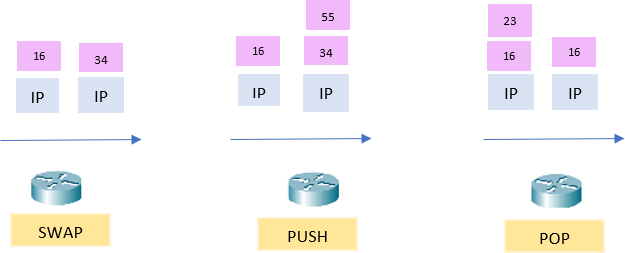
### MPLS Payload

Labelul MPLS nu are Network Level Protocol Identifier (NLPI). Acest câmp este prezent în toate frame-urile de Layer 2 pentru a indica ce protocol de Layer 3 este folosit. Așadar, cum pot routerele LSR să știe care este sarcina utilă MPLS?

Pe de-o parte, majoritatea LSR-urilor nu au nevoie de această informație. Routerele intermediare, de exemplu, au nevoie doar de informația privind eticheta de sus pentru a comuta pachetul. Pentru ca redirecționarea să se facă corect, LSR-ul respectiv trebuie să dețină ambele legături: cea locală și cea de la distanță pentru top label.

Pe de altă parte, LSR-ul de ieșire care elimină toate etichetele de deasupra pachetului trebuie să știe care este sarcina utilă MPLS deoarece trebuie să redirectioneze încărcătură utilă mai departe. Are nevoie de valoarea câmpului Network Level Protocol Identifier din frame-ul de ieșire. [3]

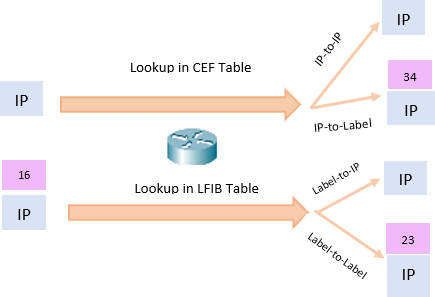
### Redirecționarea pachetelor etichetate

Operațiunile posibile de etichetare sunt swap, push și pop.

*Figură 1. 9. Operațiuni cu etichete [3]*

LSR-ul știe cum să redirecționeze pachetul privind eticheta de sus a pachetului primit și intrarea corespunzatoare in LFIB. LSR determină ce operațiune trebuie efectuată și care este următorul hop la care trebuie să fie trimis pachetul. După cum se poate observa și din figura 1.9, operațiunea de „swap” se referă la înlocuirea etichetei de sus cu o altă etichetă, operațiunea de „push” este procesul de schimbare a etichetei de sus cu nouă etichetă și adăugarea unor etichete suplimentare, iar operațiunea de „pop” se referă la eliminarea etichetei de sus, pachetul fiind trimis cu stiva de etichete rămasă sau ca pachet neetichetat. [3]

*IP Lookup vs Label Lookup*

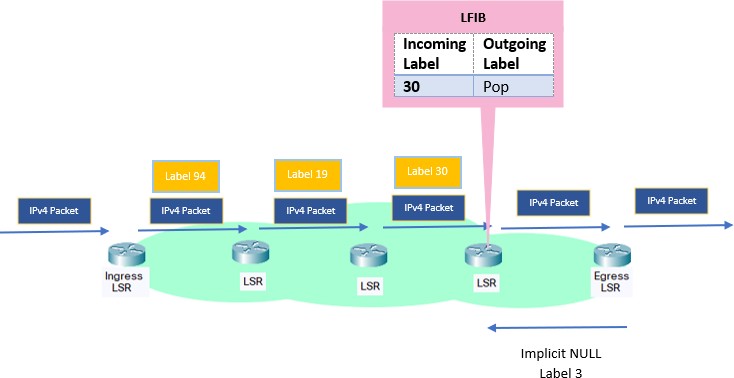
Când un router primește un pachet IP, se face IP lookup, asta înseamnă că pachetul este căutat în tabelul CEF (Cisco Express Forwarding), iar când primește un pachet etichetat, căutarea se face în LFIB. Routerul știe daca primește un pachet IP sau un pachet etichetat uitându-se în antetul de Layer2.

*Figură 1. 10. CEF și LFIB Lookup [3]*

După cum se poate observa în 1.10, atunci când un pachet IP ajunge la un LSR de intrare și este redirecționat în urma etichetării, procesul se numește redirecționare IP-to-Label. Dacă un LSR primește un pachet deja etichetat, acesta poate elimina eticheta (Label-to-IP) sau poate trimite mai departe pachetul cu o altă etichetă (Label-to-Label).

Există situații în care un LSR poate primi un pachet al cărei etichetă să nu fie stocată în LFIB. În aceasta situație, LSR-ul are două opțiuni: scoate etichetele și încearcă să trimită pachetul sau pur și simplu îl elimină din rețea. Opțiunea de a elimina pachetul este corectă deoarce LSR nu știe ce se află în spatele stivei de etichete, nu știe daca e un pachet IPv4, IPv6 sau altceva. De asemenea, există etichete rezervate, de la 0 la 15. Aceste etichete nu pot fi utilizate pentru expedierea pachetelor obișnuite. Eticheta 0 este eticheta NULL explicită pentru IPv4, 2 este eticheta NULL explicită pentru IPv6, , iar 3 este eticheta NULL implicită. 1 este eticheta de alertă a routerului, în timp ce 14 este eticheta de alertă OAM (Operation and Maintenance). [3]

Eticheta NULL implicită este utilizată pe LSR de ieșire pentru a anunța penultimul LSR că este necesar să elimine o etichetă din stiva de etichete. Este necesară utilizarea acestei metode întrucât LSR- ul de ieșire va avea de făcut cu un label lookup mai puțin, iar acest lucru va spori eficiența redirecționării pachetelor. Implicit NULL semnalizează eliminarea unei singure etichete din stivă, iar eticheta corespunzătoare, eticheta 3, nu va fi văzută niciodată în stiva de etichete a unui pachet MPLS.



*Figură 1. 11. Eliminare etichetă [3]*

Atunci când se utilizează implicit NULL există posibilitatea ca pachetul să ajungă la LSR de ieșire neetichetat, astfel poate fi trimis mai departe fară bitul Experimental (EXP) care conține informația QoS. În situația în care este nevoie de informația din EXP nu putem utiliza eticheta 3. Explicit NULL ne oferă soluția pentru această problema. LSR de ieșire semnalează eticheta cu valoarea 0 penultimului router, primește apoi un pachet etichetat cu top label 0. Nu poate redirecționa pachetul deoarece în LFIB 0 este atribuit mai multor FEC (Forwarding Equivalence Class) asftel este nevoit să elimine eticheta NULL explicită, iar informația QoS se păstrează. FEC este un termen utilizat în MPLS pentru a descrie seturi de pachete cu caracteristici identice sau similare care pot fi transmise în același mod, pot avea aceeași etichetă MPLS.

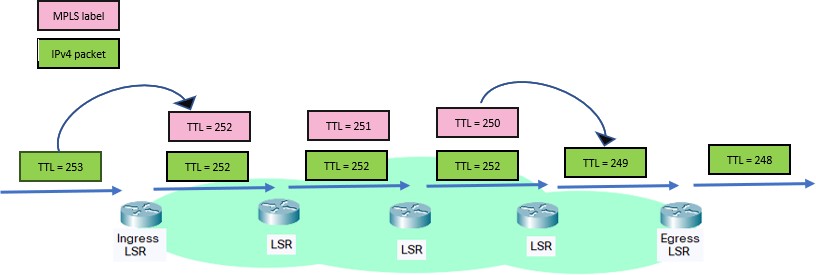
Eticheta 1 este eticheta de alertă a routerului și poate fi prezentă oriunde în stivă, mai puțin pe ultima poziție (bottom label). Când eticheta 1 este top label, aceasta anunță că pachetul respectiv necesită o analiză mai atentă. Eticheta 14, OaM, este utilizată pentru detectarea defecțiunilor, localizare și monitorizare a performanței. [3]

### Comportamentul TTL pentru pachetele etichetate

Time To Live (TTL) este un mecanism prin care un pachet este împiedicat să circule la infinit printr-o rețea. Valoarea intiala TTL este 255 și se decrementează la fiecare hop în rețea. Atunci când valoarea TTL devine 0, pachetul este eliminat, iar un mesaj ICMP tip 11, cod 0 (time exceeded) este trimis către sursa care a livrat pachetul respectiv.

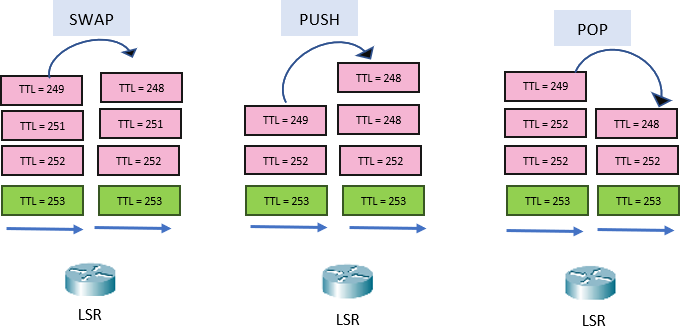
Internet Control Message Protocol (ICMP) este un mecanism care permite oricărui router să semnaleze sistemului sursă o situație anormală apărută în rutarea unui pachet utilizând mesaje de eroare, dar poate fi utilizat și pentru control atunci când un sistem testează dacă un alt sistem este capabil să recepționeze pachete trimițând mesaje de informare sau interogare. [5]

Având în vedere că etichetele MPLS sunt atașate pachetelor IP, mecanismul TTL se propagă de la headerul de IP către label stack și invers, astfel evitându-se buclele de rutare în MPLS cloud.

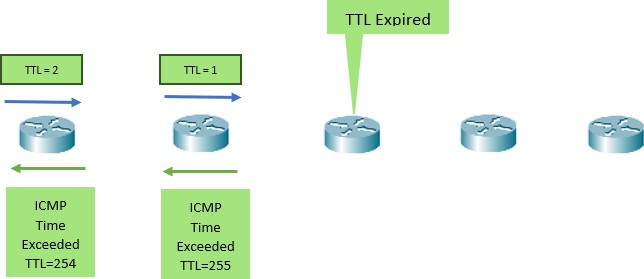


*Figură 1. 12. Mecanismul de propagare TTL între header-ul de IP și etichetele MPLS [3]*

Figura 1.12 arată comportamentul TTL în cazul IP-to-Label / Label-to-IP. Atunci când un pachet IP intră în MPLS cloud, valoarea TTL din headerul de IP este copiată în eticheta MPLS și apoi decrementată cu 1. Atunci când pachetul urmează să părăsească cloud-ul MPLS are loc procesul invers: valoarea TTL din eticheta MPLS este preluată și decrementată cu 1 în header-ul IP.



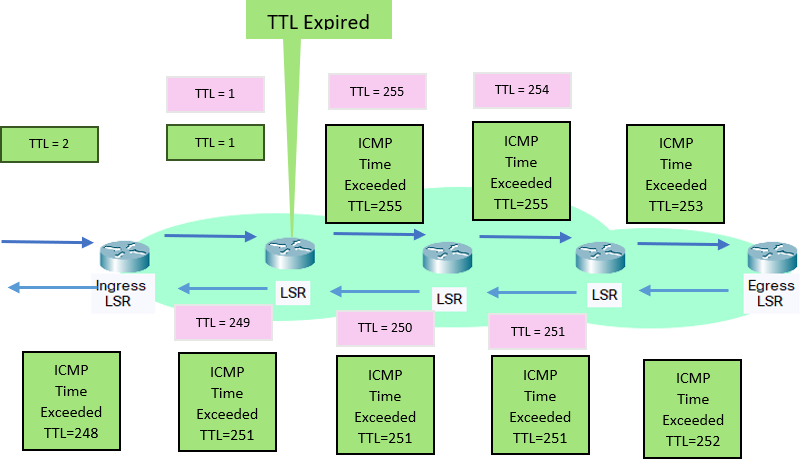
*Figură 1. 13. Mecanismul de propagare TTL în cazul operațiunilor Swap, Push și Pop [3]*

Figura 1.13 prezintă comportamentul TTL în cazul Label-to-Label. Dacă operațiunea „swap” este efectuată, valoarea TTL – 1 este copiată pe eticheta nouă. În cazul doi, operațiunea este „push” , valoarea TTL – 1 este copiată pe etichetele noi adăugate, iar în ultimul caz, „pop”, valoarea TTL – 1 este copiată în eticheta expusă, cu excepția cazului în care acea valoare este mai mare decât TTL-ul etichetei nou expuse. Un LSR intermediar nu poate modifica câmpul TTL din etichetele de bază sau din antetul IP. Un LSR poate modifica doar eticheta de sus din stiva de etichete a unui pachet. [3]

*Figură 1. 14. Mesajul ICMP "Time Exceeded" [3]*

Atunci când un router primește un pachet IP cu valoarea câmpului TTL egală cu 1, pachetul este eliminat din rețea, iar routerul transmite un mesaj ICMP către sursă, figura 1.14. După cum se observa și din figura 1.15, în MPLS lucrurile se întâmplă similar, diferență este că pachetul traversează întregul LSP

pentru a ajunge la routerul care a trimis pachetul. Acest lucru este necesar deoarece în unele situații LSR- ul care a generat mesajul ICMP nu are cunoștinte despre cum să ajungă la inițiatorul pachetului.



*Figură 1. 15. Mesajul ICMP "Time Exceeded" în Rețeaua MPLS [3]*

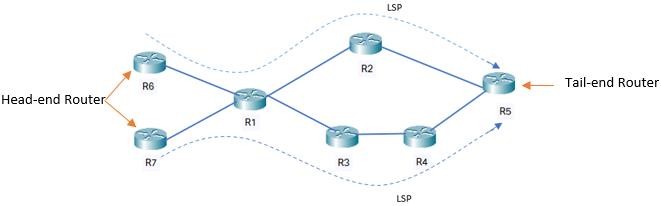
### MPLS Traffic Engineering

Rutarea în rețelele IP se bazează pe principiul costurilor minime întrucât este guvernată de necesitatea de a trimite traficul prin rețea cât mai repede posibil. Fiecare protocol IP are asociat un cost pentru fiecare link din rețea. Însumarea costurilor fiecărui link ce aparține unei căi prin rețea determină un cost al căii (cost path) care e utilizat pentru a redirecționa traficul. Fiecare pachet IP este redirecționat de fiecare router pe baza adresei IP de destinație, independent de modul în care pachetele IP au fost transmise înainte. De asemenea, modelul de redirecționare IP nu ține cont de capacitatea de lățime de bandă disponibilă a legăturii. Prin urmare, un router poate continua să redirecționeze traficul către un link, chiar dacă acel link elimină deja pachete din cauza lipsei de lățime de bandă. Rezultatul acestui comportament este că unele link-uri devin suprautilizate, în timp ce altele sunt subutilizate. Încărcătura de pe un link poate fi distribuită modificând costurile link-urilor pentru un anumit protocol de rutare, însă acest lucru este dificil având în vedere complexitatea rețelelor. [3]

Așadar, implementarea MPLS-TE este soluția pentru această problemă:

* asigură răspândirea eficientă a traficului în întreaga rețea, evitând legăturile suprautilizate
* ia în considerare lățimea de bandă configurată (statică) a legăturilor
* ia în considerare atributele legăturii
* se adaptează automat la modificarea lăţimii de bandă şi a atributelor de legătură

MPLS-TE permite o schemă TE în care routerul din capătul unui LSP (head-end router) să poată calcula cea mai eficientă rută prin rețea către routerul din celalalt capăt al LSP-ului respectiv. Acest lucru este posibil dacă routerele cunosc întreaga topologie a rețelei și lățimea de bandă rămasă pe link-uri. Datorită MPLS, redirecționarea este bazată pe sursă, nu pe destinație IP, se citește informația etichetei de intrare în LFIB și se înlocuiește cu o etichetă de ieșire.



*Figură 1. 16. MPLS TE Head End Router [3]*

Pentru a ilustra acest concept am creat figura 1.16. Dacă dorim să trimitem trafic de la R6 către R5, iar această rețea rulează numai redirecționare IP, acest trafic va urma numai calea R1-R2-R5, indiferent de ce am configurat pe routerele R6 și R7. Dacă rețeaua rulează MPLS, putem configura cele două căi ca fiind două LSP-uri diferite, astfel încât să fie utilizate etichete disticte. Atunci când R1 primește pachetul, valoarea etichetei îi va indica dacă pachetul aparține LSP-ului cu R6 ca router de capăt, sau a LSP-ului cu R7. MPLS-TE poate fi implementat în orice rețea care are LSR-uri. Totuși, deoarce lățimea de bandă și alte atribute ale link-urilor trebuie să fie cunoscute de LSR-urile de capăt ale LSP- urilor, protocolul de rutare care trebuie utilizat în punctele terminale ale MPLS-TE trebuie să fie unul de tip link-state. Conceptul de link-state se referă la modalitatea de populare a tabelei de rutare: fiecare router identifică echipamentele vecine direct conectate, își alcătuiește o tabelă de rutare stabilind cele mai bune căi și în final face schimb de informație cu celelalte routere, astfel încât fiecare router din rețea va avea o tabelă cu toate rutele existente și va cunoaște topologia rețelei în aria respectivă. Prin urmare, head-end LSR își poate da seama cum să dispună LSP-ul proiectat pentru ingineria traficului. Acest LSP se mai numește și „MPLS-TE tunnel”. Tunelul MPLS-TE este unidirecțional, întrucât și LSP-urile sunt unidirecționale, și este configurat doar pe head-end LSR, nu și pe tail-end LSR (LSR-ul unde se termină tunelul TE). Implementarea TE-tunnel previne congestia în rețea și oferă întregului trafic caracteristicile de care are nevoie. [3]

Pentru ca MPLS TE să funcționeze sunt necesere următoarele elemente de bază:

* Constrângeri pentru link-uri – se referă la cât trafic poate suporta fiecare link și ce tunel TE îl poate utiliza
* Distribuția informațiilor TE – prin protocolul link-state activat
* Un algoritm pentru a calcula cea mai bună cale de la head-end LSR la tail-end LSR (PCALC)
* Un protocol de semnalizare RSVP (Resource Reservation Protocol) – semnalizează tunelul TE prin rețeaua MPLS
* O modalitate de redirecționare a traficului către tunelul TE

### Prezentare generală a modului de operare MPLS-TE

MPLS-TE a fost denumit intial Routing with Resource Reservation (RRR). Acest nume indică faptul că principalul motiv pentru utilizarea MPLS-TE este direcționarea traficului în funcție de constrângeri și resurse. Când vorbim de resurse, ne referim la lățimea de bandă a link-urilor și unele atribute ale acestora pe care administratorul le specifică și le configurează pe link-uri. Atributele sunt anunțate de protocolol link-state, OSPF / IS-IS. Acestea două au fost extinse pentru a prelua informații specifice TE. Când se configurează TE-tunnel pe un LSR, acesta devine head-end LSR. Sunt necesare specificații cu privire la tail-end LSR și constrângerile care trebuie să fie respectate. Aceste informații sunt stocate într-o bază de date TE, iar routerul le trimite prin protocolul link-state implementat. Cu aceste informații se calculează cea mai scurtă rută care respectă toate constrângerile impuse de la head-end LSR la tail-end LSR. Calculul căii se face prin algoritmul PCALC (Path Calculation) sau prin CSPF (Constrained Shortest Path First), SPF modificat astfel încât să ia în considerare constrângerile impuse. Algoritmul PCALC rulează pe head-end LSR și realizează corespondența între cerințele de lățime de banda și atributele tunelului TE cu cele de pe link-uri, iar dintre toate căile posibile o alege pe cea mai scurtă. LSR-urile intermediare au nevoie de etichetele de intrare și de ieșire pentru LSP-ul asociat cu tunelul TE respectiv. Acestea pot învața etichetele numai dacă etichetele sunt semnalizate. RSVP-TE ( Resource Reservation Protocol - Traffic Engineering) este protocolul de semnalizare al MPLS care se ocupă cu semnalizarea tunelului TE de-a lungul căii. Mesajul RSVP PATH este trimis de la head-end LSR către tail-end LSR, purtând o solicitare pentru o etichetă MPLS. Acesta este direcționat cu ajutorul Explicit Route Object (ERO) care specifică fiecare hop din rețea pe care mesajul trebuie să-l parcurgă de-a lungul tunelului. Mesajul RSVP RESV se trimite în sens invers și poartă eticheta MPLS pe care fiecare LRS de-a lungul căii o va utiliza pentru redirecționarea traficului pe tunelul respectiv. În plus,

mesajul RESV indică LSR-urilor intermediare să rezerve resurse, lățime de bandă, pe link-urile pe care tunelul TE le utilizează. [3]

După cum am menționat anterior, MPLS-TE se bazează pe redirecționarea traficului pe baza unor constrângeri. Protocoalele OSPF și IS-IS au necesitat anumite schimbări pentru a suporta informația specifică ingineriei traficului. Informațiile TE ale unui link sunt:

* + - * Metrica TE – este un parametru utilizat pentru a construi o topologie, poate fi diferită de costul OSPF sau metrica IS-IS a link-ului
      * Lățimea de bandă maximă - lățimea de bandă totală a conexiunii
      * Lăţimea de bandă maximă rezervabilă - lățimea de bandă a link-ului disponibilă pentru TE
      * Lăţime de bandă nerezervată - este diferența dintre lățimea de bandă maximă și orice lățime de bandă care este deja rezervată
      * Grup administrativ - camp de 32 de biți, administratorul poate seta individual fiecărui bit o semnificație cum ar fi: un link are un delay mai mic de 100ms sau un link are o anumită viteză, specificând astfel resursele și caracteristicile disponibile

### Rutarea și costul unui TE-LSP

Atunci când traseul unui tunel TE se calculează, rezultatul este cea mai scurtă cale posibilă, iar mai mulți factori sunt luați în considerare. Atributele link-urilor TE sunt esențiale întrucât fiecare link din rețeaua MPLS are mai multe caracteristici care trebuie comunicate către head-end router pentru ca acesta să ia decizia cu privire la disponibilitatea utilizării acelui link pentru tunelul TE. Un link activat pentru ingineria traficului poate avea următoarele caracteristici:

* + - * Lăţimea de bandă maximă rezervabilă – lățimea de bandă maximă rezervabilă în pool-ul global, cel pe care îl folosesc toate tunelurile TE obișnuite
      * Indicatori de atribut (Attribute Flags) – indică resursele link-ului, capacitatea sau politicile administrative; indică dacă un tunel cu nevoi specifice de resurse poate traversa acel link
      * metrica TE – are ca valoare implicită costul IGP al link-urilor, însă poate fi modificată
      * Shared Risk Link Groups (SRLGs) – caracteristică a link-urilor care indică faptul că acestea au fibră comună, iar dacă aceasta cedează mai multe link-uri sunt afectate, deci sunt supuse aceluiași risc
      * Maximum Reservable Sub-Pool Bandwidth - reprezintă o fracțiune din lățimea de bandă globală a pool-ului

De asemenea, există și o serie de atribute care caracterizează un tunel MPLS-TE precum:

* + - * Destinația tunelului: ID-ul tail-end LSR către care ar trebui direcționat tunelul TE
      * Lăţimea de bandă dorită
      * Configurarea și menținerea priorităților
      * Reoptimizare
      * Opțiuni de cale

### Calcularea căii tunelului TE

Stabilirea căii tunelului prin rețea depinde de următorii factori:

1. Opțiune de configurare a căii

Această opțiune poate fi configurată pe head-end router. Un tunel poate fi configurat în două moduri: explicit sau dinamic. În mod explicit, trebuie să specificăm ID-ul/ adresa IP a fiecărui router pe care trebuie direcționat tunelul TE, inclusiv tail-end router. În mod dinamic, head-end router este lăsat să ia deciziile despre cum este direcționat cel mai bine tunelul TE în întreagă rețea. În acest caz trebuie configurată doar destinația finală a tunelului. Head-end router analizează baza de date MPLS-TE învățată

fie prin OSPF, fie prin IS-IS și ia în considerare informațiile de topologie și resursele link-urilor. Acest proces de calcul se realizează cu ajutorul algoritmului PCALC (Path Calculation).

1. Prioritățile de configurare și menținere

Tunelurile MPLS-TE pot avea o importanță diferită în rețea. De exemplu, unele tuneluri sunt mai lungi, din punct de vedere al numărului de hop-uri (routere) care trebuie străbătute în rețea și pot fi considerate mai importante decât tunelurile mai scurte. De asemenea, într-o rețea putem avea tuneluri TE cu diferite nevoi în ceea ce privește lățimea de banda, tunelurile care necesită lățime de banda mai mare fiind considerate prioritare. Semnalarea unui tunel cu o importanță mai ridicată poate fi făcută mai târziu, iar acest lucru poate duce la o configurare nu tocmai optima pentru tunelul respectiv, sau la imposibilitatea găsirii link-urilor cu lățimea de banda necesară. Așadar, pentru a evita aceste situații, s- au introdus priorități pentru tunelurile TE care asigură direcționarea optimă a tunelurilor importante. [3] Pentru un tunel TE se pot configura două tipuri de priorități: configurare (setup) și menținere (holding). Ambele priorități indică dacă tunelul TE respectiv anticipează un alt tunel. Cu cât valoarea priorității este mai mică, cu atât importanța este mai mare. Prioritatea de configurare indică cât de important este tunelul pentru a anticipa celelalte tuneluri, iar prioritatea de menținere indică faptul că un tunel cu o importanță ridicată care de asemenea, necesită mai multe resurse, nu va fi niciodată anticipat

de alte tuneluri.

Atât prioritatea de configurare, cât și prioritatea de menținere variază de la 0 la 7, iar cu cât valoarea este mai mică, cu atât prioritatea este mai mare. În mod implicit, ambele priorități au valoarea

7. Este important ca prioritatea de configurare să nu fie setată cu o valoare mai mare decât valoarea priorității de menținere. [6]

1. Indicatori de atribut și biți de afinitate

După cum am menționat anterior, attribute flags indică resursele link-ului, capacitatea sau politicile administrative. Totodată, indică dacă un tunel cu nevoi specifice de resurse poate traversa acel link. Aceste flag-uri sunt pe 32 de biți, fară sintaxă asociată. Fiecare bit poate fi setat sau nesetat și poate avea orice semnificație pe care operatorul și-o dorește. De exemplu, un bit poate indica că link-ul are o întârziere mai mică de 100 ms.

1. Reoptimizare

Un tunel TE poate ajunge pe o cale din rețea care nu mai este considerată ca fiind optimă. Acest lucru se întâmplă dacă un link care a fost oprit anterior, devine din nou disponibil sau dacă un link nu avea suficientă lățime de bandă disponibilă în momentul în care tunelul a fost semnalat, iar acum lățimea de bandă rezervată a link-ului a scăzut suficient astfel încât să permită rutarea tunelului TE. Reoptimizarea este procesul prin care un tunel este redirecționat în rețea pe calea cea mai optimă. Există doi declanșatori care determină reoptimizarea tunelului TE: reoptimizare bazată pe evenimente și reoptimizare manuală. Reoptimizarea bazată pe evenimente are în vedere reoptimizarea tunelului atunci când un link din rețea devine din nou disponibil. În mod implicit această reoptimizare nu este activată, însă utilizând o comandă specifică o putem utiliza. Reoptimizarea manuală se referă la reoptimizarea forțată, imediată a tuturor tunelurilor TE de pe head-end router sau a unui tunel specific. [3]

### RSVP – TE

Protocolul Resource Reservation Protocol for Traffic Engineering este o versiune extinsă a protocolului RSVP pentru MPLS. Acesta utilizează extensii de inginerie de trafic pentru a semnaliza automat LSP-urile, solicitând resurse pentru fluxuri de trafic unidirecționale. RSVP-TE funcționează cu protocoale de rutare unicast și multicast și consultă tabelele locale de rutare pentru a transmite mesajele RSVP-TE.

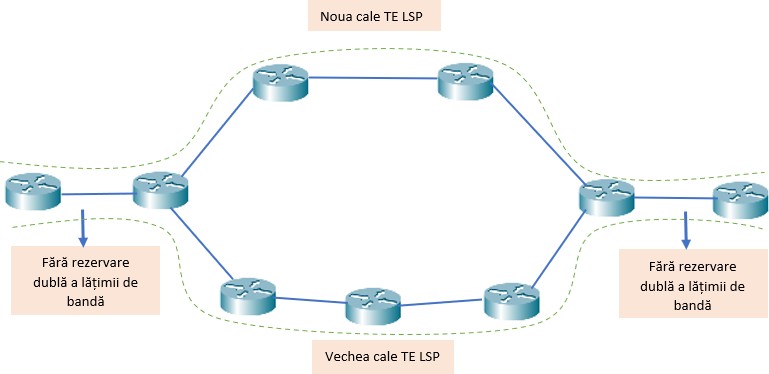
RSVP-TE utilizează mesajele PATH și RESV pentru a semnaliza calea. Fiecare hop, LSR, pe care ar trebui să-l traverseze tunelul TE este introdus într-un ERO (Explicit Route Object), care este o listă ordonată de adrese IP ale LSR-urilor. Mesajul PATH este transmis de la head-end router către router- ul următor. La primirea mesajului, acesta elimină propria adresă IP din ERO și trimite mesajul PATH către următorul router. Procesul se continuă până când tail-end router primește mesajul PATH, urmând ca acesta să trimită înapoi spre head-end router mesajul RESV. Dacă mesajul RESV este recepționat fără erori RSVP, calea este configurată, iar tunelul TE devine up și semnalizat. Așadar, mesajele PATH sunt utilizate pentru a solicita și semnala legăturile etichetelor necesare pentru a stabili LSP-ul de la intrare către ieșire și pentru a facilita routerele de-a lungul căii să facă rezervările necesare pentru lățimea de banda, în timp ce mesajele RESV sunt răspunsul la mesajele PATH recepționate. [3]

Pentru a face RSVP compatibil cu MPLS-TE s-au introdus noi extensii ale protocolului pentru a sprijini îdeplinirea cerințelor necesare pentru ingineria traficului prin MPLS. Pentru a rula corect, RSVP- TE necesită un protocol de rutare IGP cu extensii pentru TE care să ruleze simultan, astfel încât anunțarea constrângerilor impuse pe diferitele link-uri din rețea să fie posibilă.

Următoarele capabilități TE sunt acceptate:

* + - * Hop limit - numărul maxim de noduri (routere) LSR pe care un anumit LSP le poate traversa, utilizat pentru a controla timpul maxim de întârziere
      * Admin groups – grupurile administrative sunt o metodă pentru definirea unor LSR-uri specifice care trebuie evitate sau incluse în LSP; poate exista situația în care se dorește ca un LSR solicitat intens să fie exclus din calea LSP-ului sau un anumit LSR să facă parte din acel LSP
      * Bandwidth – Capacitatea de lățime de bandă este foarte importantă întrucât mesajul PATH conține cerera de rezervare a lățimii de banda (bandwidth reservation request), iar capacitatea disponibilă este confirmată prin mesajul RESV

Metoda Shared Explicit RSVP (SE) este anunțată în mesajul PATH atunci când s-a găsit o cale mai optimă pentru un tunel TE, iar LSP-ul actual trebuie înlocuit. SE determină înlocuirea căii actuale TE LSP doar atunci când noua cale a fost configurată complet, evitând asftel pierderile de trafic. Atunci când cerințele pentru un tunel TE se modifică, este necesară reoptimizarea. În lipsa utilizării RSVP SE se va rezerva lățimea de bandă de două ori: o dată pentru vechiul LSP și o dată pentru noul LSP. [3]



Există și alte tipuri de mesaje RSVP:

*Figură 1. 17. Stilul RSVP SE [3]*

* + - * + Path Tear – este similar cu mesajul PATH, cu excepția faptului că este trimis atunci când routerul principal vrea să semnaleze că TE-LSP ar trebui să fie down după intervenția unui administrator
        + Resv Tear – este similar cu mesajul RESV, cu excepția faptului că este trimis de către tail-end router ca răspuns la primirea unui mesaj path-tear
        + PathErr – este un mesaj trimis către head-end router și este recepționat cel mai adesea atunci când un link utilizat de TE LSP devine down
        + ResvErr – este un mesaj trimis către tail-end router

## Segment-Routing MPLS

Segment Routing (SR) este un protocol conceput pentru a transmite pachete de date folosind modelul de rutare sursă (source routing model). Segment Routing MPLS (SR-MPLS) este implementat pe baza planului de redirecționare (forwarding plane) și are ca scop creșterea flexibilității și scalabilității rețelei. SR împarte căile în segmente și atribuie un ID de segment (SID) atât segmentelor, cât și nodurilor specifice, urmând ca acestea să fie aranjate secvențial într-o lista pentru a forma o cale de redirecționare.

### Beneficiile Segment-Routing

Segment-Routing aduce o serie de beneficii pentru rețelele actuale, dintre care putem enumera:

* + - *Simplitate*

Atunci când SR este implementat peste un plan de date MPLS oferă posibilitatea de creare a tunelurilor MPLS fară un alt protocol suplimentar, doar utilizând IGP (OSPF sau IS-IS). Operarea se face mult mai simplu, fără protocoale de distribuție a etichetelor sau protocoale de semnalizare (LDP, RSVP).

* + - *Suportă Fast Reroute (FRR)*

Furnizează FRR automat pentru orice topologie. Dacă un link sau un nod eșuează în rețea, MPLS utilizează mecanismul FRR pentru convergență. Cu segmnet-routing timpul de convergență este sub 50 msec.

* + - *Scalabil*

Se înlătură necesitatea LDP, așadar nu mai avem nevoie de etichete din baza de date LDP, pentru MPLS-TE se elimină LSP-urile și complexitatea tunelurilor de configurat, astfel SR poate opera la o scară mai mare, fiind adaptabil la nevoile și cerințele de schimbare ale utilizatorilor. [7]

* + - *Implementare IP/MPLS sau IPv6*

În rețelele IP/MPLS poate fi implementat utilizând planul de date fără nicio schimbare, iar în rețelele IPv6 se implementează adăugând un header de extensie.

* + - *Interoperabilitatea cu protocoalele de distribuție a etichetelor*

SR poate interacționa cu ușurință cu protocoalele LDP și RSVP-TE existente; cu LDP se configurează mapare simplă a etichetelor SR – LDP, iar pentru RSVP este configurat un „binding segment identifier”; prin aceste metode interacțiunea între domeniile SR și non-SR este posibilă.

* + - *Compatibil cu SDN*

Segment-routing este o arhitectură concepută pentru a permite implementarea SDN (Software Define Network). Acesta oferă un echilibru între inteligența distribuită bazată pe rețea, iar aici ne referim la protecția automată a link-urilor și a nodurilor, dar și inteligența centralizată bazată pe controller, ce are ca scop optimizarea traficului. [7]

SR integrat cu o arhitectură SDN bazată pe controller oferă garanții stricte de performanță a rețelei și ajută la utilizarea eficientă a resurselor. Rețeaua folosește informații minime de stare (state information) pentru a îndeplini aceste cerințe. Controller-ul are o imagine completă asupra topologiei de rețea și a fluxurilor de trafic. De asemenea, la solicitarea unui router, acesta poate transmite informații cu privire la anumite caracteristici specifice cum ar fi: întârzierea sau lățimea de bandă. În plus, poate calcula calea optimă și returna o listă de segmnete (segment list) corespunzătoare pentru o anumită destinație, similară cu stiva de etichete MPLS, nefiind necesar ca router-ul solicitant să realizeze o semnalizare suplimentară în rețea înainte de a trimite traficul. În plus, întrucât rețeaua menține doar starea segmentului, capacitatea rețelei crește, poate suporta un număr mai mare de cereri fără crearea congestiei de trafic. [8]

* + - *Ingineria traficului*

Controller-ele și platformele de orchestrare pot interacționa cu SR – Traffic Engineering pentru o centralizare optimizată. O congestie în rețea poate declanșa o aplicație pentru a optimiza, recalcula tunelurile SR-TE. Acestea sunt programate dinamic în rețea utilizând mecanismul de orchestrare a rețelei. Acest mecanism reprezintă o abordare bazată pe politici de automatizare care coordonează componentele

hardware și software pe care o aplicație sau un serviciu software le necesită pentru a rula. Agilitatea este crescută întrucât tunelurile SR-TE nu necesită semnalizare. [8]

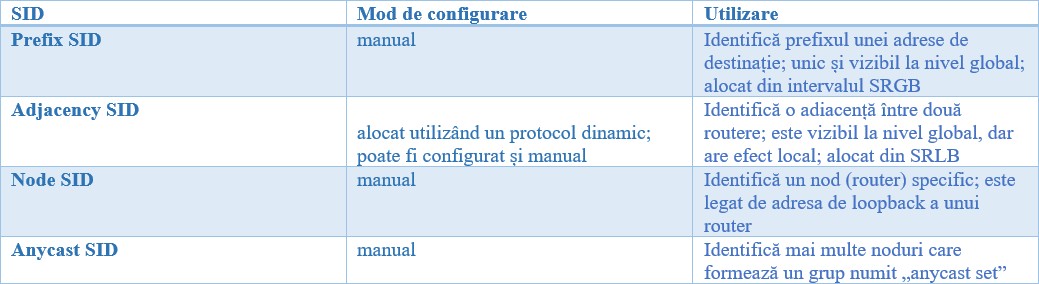
### Noțiuni de bază

Când vorbim despre SR-MPLS avem în vedere următoarele noțiuni de bază:

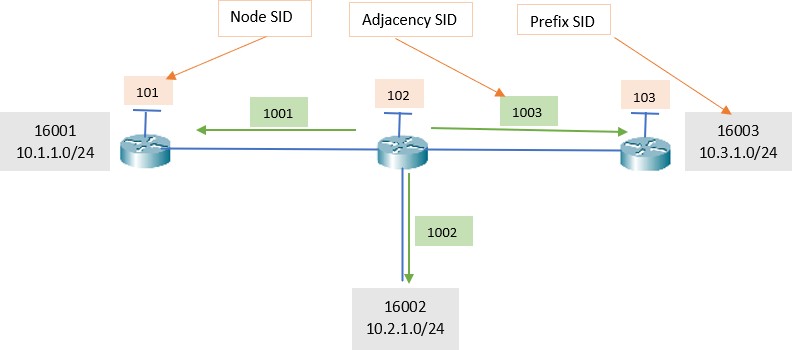
* + - Segment Routing Domain (SRD) – domeniul de rutare
    - Segment ID (SID) – ID de segment care identifică în mod unic un segment sau un nod
    - Segment Routing Global Block (SRGB) – un set de etichete globale specificate de administrator pentru SR-MPLS
    - Segment Routing Local Block (SRLB) - un set de etichete locale specificate de administrator pentru SR-MPLS [9]

### Tipuri de SID

Tabelul 2.1 ilustrează tipurile de identificatori de segment și utilizarea acestora:



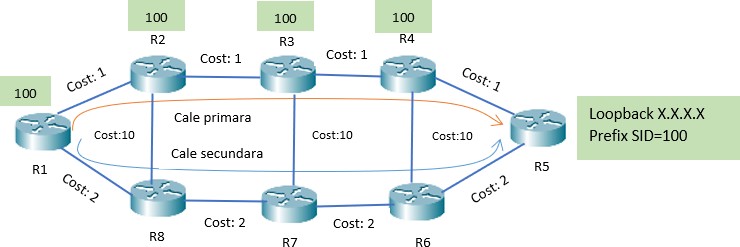
*Tabel 2. 1. Tipuri de SID – mod de configurare și utilizare [9]*



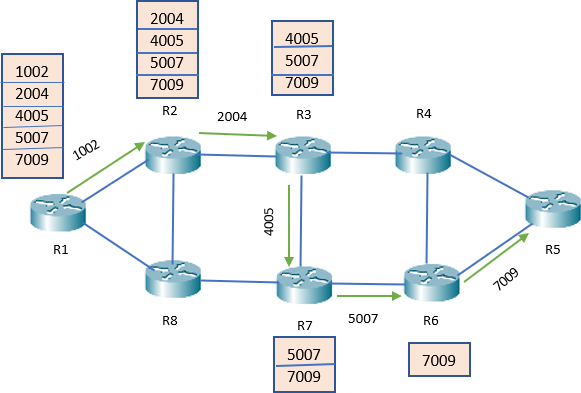
*Figură 2. 1. Tipuri de SID [9]*

Figura 2.1 ilustrează tipurile de SID într-o mică topologie. Cu alte cuvinte, un prefix SID are același înțeles cu o adresă destinație, iar un prefix de adiacență indică interfața de ieșire, link-ul dintre routere pe care se transmit pachetele de date.

Aceste 3 tipuri de identificatori de segment pot fi folosiți separat sau în diverse combinații. Vom prezenta 3 astfel de situații:

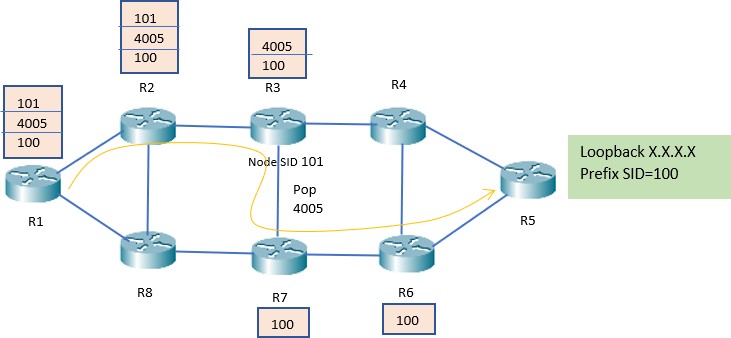


*Figură 2. 2. Prefix SID-bazat pe căile de redirecționare [9]*

În figura 2.2 , destinația este R5 și are ca Prefix-SID valoarea 100, această valoare fiind anunțată în întreaga arie cu ajutorul protocolului IGP configurat pe routere. Fiecare router va rula SPF pentru a stabili calea cea mai scurtă către R5, calea cu cel mai mic cost.

*Figură 2. 3. Adjacency SID- bazat pe căile de redirecționare [9]*

În figura 2.3 , fiecărei adiacențe i se atribuie un SID de adiacență. Lista cu identificatorii de adiacență este definită pe router-ul de intrare și este utilizată pentru a specifica exact căile explicite.



*Figură 2. 4. Adjacency SID + Node SID - bazat pe căile de redirecționare [9]*

În figura 2.4 este ilustrată utilizarea în paralel a SID-urilor de adiacență și de nod pentru a forța o adiacență specifică pe o anumită cale. Manipularea traficului în acest mod permite o echilibrare pe căi, acestea nu sunt fixate și sunt numite și căi explicit libere. 100 și 101 sunt node SID pentru R5 și R3, iar 4005 SID de adiacență pentru link-ul dintre R3 și R7. După cum se poate observa din figură, R1 are ca top label 101, astfel încât traficul să fie redirecționat către R3. R2 schimbă eticheta de sus cu una de

aceeași valoare și trimite mai departe pachetul către R3, urmând ca acesta să elimine eticheta corespunzătoare lui, 101, și să trimită pachetul mai departe având ca top label 4005, SID-ul de adiacență. În final, routerul R6 va primi un pachet cu eticheta 100, va scoate eticheta și va trimite pachetul către R5. [9]

### Mecanism de redirecționare SR

Atunci când SR este utilizat direct în arhitectura MPLS, mecanismul de redirecționare rămâne neschimbat. SID-urile sunt codificate ca etichete MPLS, iar lista de segmente este văzută că o stivă de etichete. Segmentul care trebuie procesat se află în partea superioară a stivei, iar odată ce acesta este analizat, eticheta corepsunzătoare este eliminată.

Având în vedere că Prefix SIDs sunt configurate manual pe diferite dispozitive se iau în considerare situațiile în care apar conflicte de etichete, care pot fi conflicte de prefixe, atunci când un prefix este asociat cu SID-uri diferite, sau conflicte de SID, atunci când același SID e asociat cu prefixe diferite. La apariția ambelor tipuri de conflict în același timp este necesară soluționarea prioritara a conflictului de prefix. [9]

De exemplu, dacă avem următoarele rute: a. 1.1.1.1/32 1

b. 1.1.1.1/32 2

c. 2.2.2.2/32 3

d. 3.3.3.3/32 1

Se vor trata prima dată conflictele de prefix, deci avem în vedere rutele a și b. Criteriul de alegere constă în SID-ul cel mai mic. Astfel rămân următoarele căi:

a. 1.1.1.1/32 1

c. 2.2.2.2/32 3

d. 3.3.3.3/32 1

Având în vedere că sunt 2 căi cu același SID, se va alege ruta care are un prefix mai mic: a. 1.1.1.1/32 1

b. 2.2.2.2/32 3

Similar ca în MPLS, în SR-MPLS avem stiva de etichete care reprezintă un set de etichete cu identificator de adiacență (adjacency SID) unde fiecare etichetă identifică o adiacență a unui router local, iar stiva descrie toate adiacențele de-a lungul unui SR-MPLS TE-LSP. La redirecționarea pachetelor, routerul caută adiacența corespunzătoare etichetei cu informația SID-ului de adiacență, îndepărtează eticheta și trimite mai departe pachetul. Atunci când toate etichetele sunt eliminate din stiva de etichete, pachetul este trimis în afara tunelului SR-MPLS. [9]

După cum am menționat anterior, unul dintre beneficiile majore aduse de Segment-Routing este eliminarea nevoii de semnalizare suplimentară și a protocoalelor de distribuție a etichetelor, LDP și RSVP-TE, utilizând în schimb doar protocoalele de rutare în sine, IGP sau BGP, pentru a distribui etichetele. [7]

Comparând distribuția etichetelor în SR utilizând IGP și distribuția etichetelor utilizând LDP vom identifica câteva asemănări și deosebiri: în ambele cazuri, etichetele sunt anunțate automat printre routere atunci când se formează un peering de adiacență și LSP-urile nu se configurează manual, în cazul distribuirii etichetelor utilizând LDP este necesară sincronizarea LDP-IGP, eliminând necesitatea LDP în implementările SR se elimină automat și problema sincronizării. De asemenea, în Segment-Routing,

SID-urile sunt unice la nivel global, atribuite în întreg domeniul, în timp ce în LDP, etichetele sunt semnificative la nivel local, alocându-se o valoare unică la fiecare hop. Totodată, SR oferă capabilități TE pentru adresele IPv6, în timp ce suportul LDP pentru adresele V6 lipsește. [7]

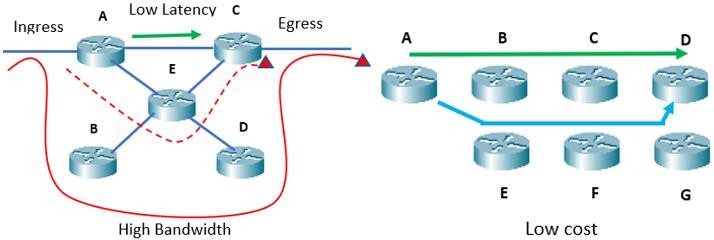
Segment-Routing MPLS Traffic Engineering (SR-MPLS TE) este o nouă tehică de direcționare a traficului implementată pe baza unei extensii IGP, pornind de la MPLS-TE. Tunelurile SR-MPLS TE sunt capabile să îndeplinească cerințe superioare în comparație cu RSVP-TE. În cazul SR-MPLS TE, protocolul IGP extins atribuie și distribuie etichetele, iar fiecărei legături îi corespunde o singură etichetă partajată de toate LSP-urile, reducându-se astfel consumul de resurse și volumul de lucru cu tabelele de redirecționare a etichetelor, pe când în cazul RSVP-TE, MPLS alocă și distribuie etichete, iar fiecărui LSP i se atribuie o etichetă, astfel se consumă mai multe resurse și rezultă sarcini dificile de lucru cu tabelele de redirecționare.

De asemenea, în cazul SR scalabilitatea este crescută, informațiile din tunel sunt transportate în pachete, astfel se elimină necesitatea transmiterii informațiilor despre starea tunelului, în timp ce în cazul RSVP-TE scalabilitatea este considerată ca fiind mai scăzută, fiind necesară menținerea informațiilor despre starea tunelului. Din punct de vedere al funcționalității planului de control, în SR se utilizează doar IGP, astfel numărul de protocoale utilizate este redus, în timp ce prin utilizarea RSVP-TE planul de control devine complex. În cazul SR-MPLS, controlul și ajustarea căii se poate face numai prin operarea etichetei de la intrare, configurația nu trebuie livrată fiecărui nod, iar atunci când un router ce aparține căii eșuează, controller-ul recalculează calea și actualizează stiva de etichete a routerului de intrare pentru a ajusta calea respectivă. În cazul utilizării RSVP-TE, ajustarea normală sau aparția unui scenariu de defecțiune implică transmiterea informațiilor de configurare către fiecare router. [9]

SR-TE a devenit frecvent utilizat nu numai pentru că oferă simplitate și scalabilitate, ci și pentru că oferă o metodă de implementare a căilor de traffic-engineering utilizând comportamentul ECMP al IP. Equal-Cost Multi-Path routing (ECMP) este o strategie de rutare în care redirecționarea pachetelor către o singură desintație poate avea loc pe mai multe căi cu prioritate egală, e o decizie care se face independent pe fiecare router și determină echilibrarea traficului în rețea. Complexitatea este redusă semnificativ în rețea datorită beneficiilor suplimentare ale automatizării prin implementarea SR Policy și direcționarea automată a traficului. [10]

SR Policy se identifică prin următorul triplet:

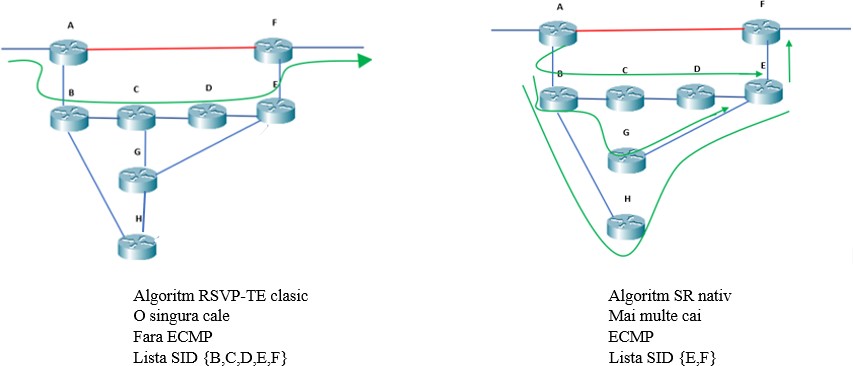
* Head-end – unde este inițiată politica
* End-point – care este destinația politicii
* Culoare – o valoarea numerică arbitrară care arată diferite tipuri de politici, de exemplu, verde pentru calea cu latență scăzută și roșu pentru calea cu lățime de bandă mare



*Figură 2. 5. SR Policy în Segment-Routing [11]*

În figura 2.5 , sunt ilustrate două politici diferite: verde cu latența scăzută și roșu cu lățime de bandă mai mare, iar în figura 2.5 albastru identifică un cost scăzut și verde o întârziere mica. O politică

(policy) poate avea mai multe căi candidate. De exemplu, politica roșie are două căi candidate (candidate paths). Calea preferată se identifică prin numărul de preferință cel mai mare, acesta fiind un atribut specific al candidate paths. Un alt atribut este Binding Segment (BSID), care este un nou tip de segment fundamental atunci când vorbim de traffic-engineering. De exemplu, în figura 2.5, BSID are o valoare X cu următoarea cadidate-path X= {Prefix B, Prefix D}. Dacă router-ul A primește un pachet cu stiva de etichete {X, Prefix C}, va ști că X este BSID. Routerul va executa operațiunea „pop” pentru eticheta de sus și va adăuga calea candidată la stiva de etichete, având că rezultat o stivă nouă care urmează traseul roșu {Prefix B, Prefix D, Prefix C}.

Un alt avantaj SR-TE în comparație cu RSVP-TE este reprezentat de utilizarea algoritmului nativ SR. În loc de utilizarea algoritmilor complecși ai RSVP-TE pentru a calcula cea mai optimă cale bazată pe constrângeri, SR-TE utilizează SR Native Algortihm. Acest lucru are că rezultat reducerea stivei de etichete și o cale care este echilibrată (load balanced) mulțumită capabilităților native ECMP ale SR. [10]

*Figură 2. 6. Algoritmul RSVP-TE versus Algoritmul SR nativ [10]*

În figura 2.6 este ilustrată o comparație între mecanismele clasice de traffic-engineering și mecanismele SR Native TE. Se dorește găsirea celei mai bune căi între routerele A și F care nu includ link-ul roșu, cel direct conectat. Pentru a avea ECMP utilizând RSVP-TE sunt necesare configurații suplimentare care ar spori complexitatea implementării. În schimb, utilizarea SR care suportă nativ ECMP, aduce avantaje precum: echilibrarea traficului și utilizarea a mai puține SIDs.

SR-TE are un nou mod de a instanția SR Policy la cerere în loc să o configureze în prealabil. Utilizând protocolul BGP, se creează un mod dinamic, flexibil și automat de aplicare a politicii SR. Aceasta permite realizarea ingineriei de trafic într-un mod simplu și automat, reprezentând un avantaj în comparație cu protocolul RSVP-TE care necesită preconfigurarea unor tuneluri complexe. [10]

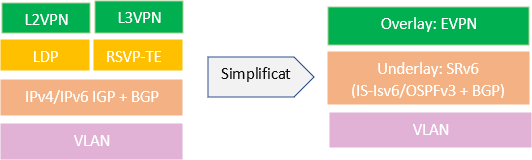
## Segment-Routing IPv6

Segment-Routing IPv6 deschide noi oportunități de implementare care depășesc funcționarea rețelei implementate cu SR-MPLS. SRv6 aduce conceptul de seturi de instrucțiuni (funcții) care permit modele complexe de programare în rețea. [10] Utilizând tehnologia de redirecționare IPv6 existentă, SRv6 implementează programarea în rețea prin headere flexibile de extensie IPv6, reduce numărul de tipuri de protocoale necesare, oferă extensibilitate și îndeplinește cerințele diversificate a mai multor servicii noi. [12]

Unul dintre motivele care au stat la baza dezvoltării SRv6 a fost reprezentat de accelerarea dezvoltării aplicațiilor în internet, datorată progresului informatizării globale. Pe măsură ce rețelele cresc, apare o diversitate a tipurilor de servicii de rețea și a cerințelor pe care acestea le impun pentru rețele. În acest context, rețelele tradiționale IP/MPLS prezintă o serie de provocări. Una dintre ele ar fi reprezentată de spațiul limitat din MPLS IPv4 având în vedere faptul că noile servicii necesită informații suplimentare de redirecționare ce trebuie adăugate în pachete. Spațiul etichetei MPLS este fixat la 20 de biți, neavând extensibilitate, nu mai poate îndeplini eficient noile cerințe de programare în rețea. O alta problemă ar fi utilizarea simultană a serviciilor L3VPN și L2VPN care determină coexistarea numeroaselor protocoale (LDP, RSVP, IGP, BGP) pe un dispozitiv, fapt ce duce la un management complicat și la o implementare dificilă pe scară largă. De asemenea, strânsa legătură dintre planul de date și planul de control fac dificilă gestionarea dezvoltării rapide de noi servicii.

SRv6 este o tehnologie bazată pe planul de redirecționare IPv6. Combinând avantajele de source- routing ale SR, precum și simplitatea și extensibilitatea IPv6, SRv6 oferă o serie de avantaje unice. Printre acestea putem menționa:

* Inteligența: SRv6 oferă capabilități ridicate de programare, este bazat în totalitate pe arhitectura SDN, îndeplinește cerințele diverse a numeroaselor servicii, permite programarea și optimizarea rețelei pe baza informațiilor globale
* Ultra-simplificat: cu SRv6, LDP, RSVP-TE și etichetele MPLS nu mai sunt necesare, simplificându-se astfel foarte mult protocoalele și gestionarea; în plus, combinația dintre EVPN (Ethernet VPN) și SRv6 înlocuiește mai multe protocoale cum se poate observa în figura de mai jos:



*Figură 3. 1. Beneficiu SRv6 [12]*

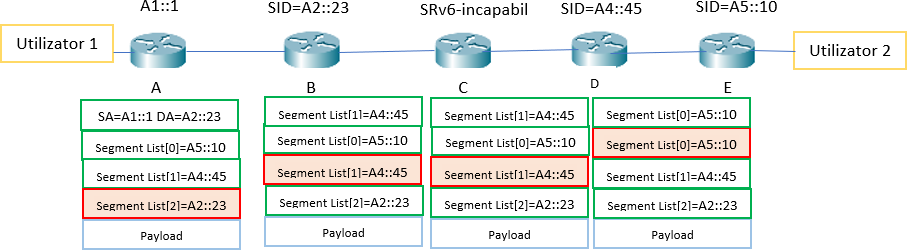
* Ip nativ: SRv6 redirecționeaza pachetele bazate pe IPv6 nativ și este implementat prin extinderea antetelor pachetelor fără a modifica structura originală de încapsulare a pachetelor IPv6, așadar pachetele SRv6 rămân pachete IPv6 și pot fi identificate de dispozitive IPv6 obișnuite; acest fapt aduce un avantaj întrucât dipozitivele IPv6 obișnuite pot fi implementate împreună cu dispozitivile SRv6, stabilindu-se o compatibilitate mai bună a SRv6 cu rețelele existente [12]

SRv6 implementează programarea căilor prin combinarea segmentelor. Pentru SRv6, o rețea este văzută ca un computer. Similar cu programarea computerelor, serviciile de rețea sunt programate într-o serie de structuri de redirecționare, care sunt livrate dispozitivelor de rețea de-a lungul unei căi, astfel încât să se îndeplinească cerințele de personalizare a serviciului.

Pentru implementarea SR pe baza planului de redirecționare IPv6 se adaugă un Segment-Routing Header (SRH) ca header de extensie. SRH specifică o cale explicită IPv6 și stochează informațiile din lista de segmente IPv6 care reprezintă o cale de redirecționare formată prin aranjarea segmentelor și a nodurilor de rețea în secvență.

Segmentele SRv6 definesc instrucțiunile de rețea, dictând unde ar trebui să meargă pachetele și cum ar trebui să ajungă acolo. Fiecare segment SRv6 este identificat printr-un SID SRv6, care este o adresă IPv6 de 128 de biți și constă în 3 părți: localizator, funcție și argumente. Acestea au lungimi personalizabile, oferind capabilități flexibile de programare. Localizatorul identifică locația unui nod de rețea și oferă capacitatea de rutare IPv6, ghidând în același timp căutarea adresei pentru redirecționarea pachetelor. Diferite comportamente de redirecționare sunt exprimate prin diferite funcții. Argumentul reprezintă un câmp opțional suplimentar utilizat pentru a defini parametrii pentru execuția instrucțiunilor și poate conține flux, serviciu sau alte informații. [12]

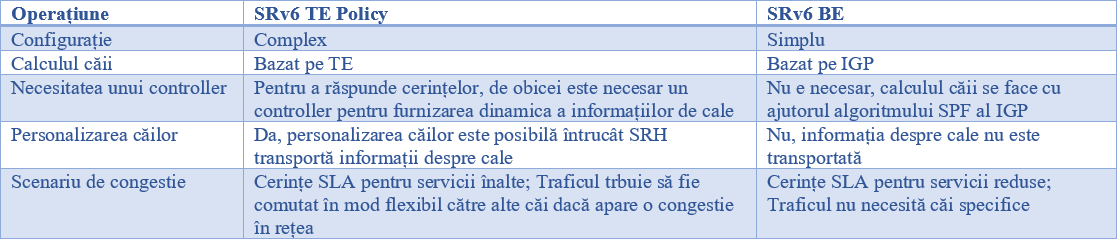
Pentru a ilustra procesul de redirecționare a unui pachet SRv6 putem urmări figura de mai jos în care un pachet trebuie redirecționat de la Utilizatorul 1 către Utilizatorul 2. Routerele A,B,D și E acceptă SRv6, în timp ce routerul C nu, acesta acceptând doar IPv6. Procesul de redirecționare a pachetelor curpinde mai multe etape. Nodul sursa A încapsulează informații despre calea SRv6, SID-urile fiind încapsulate în ordine inversă, iar valoarea câmpului SL indică lista de segmente ce trebuie procesată. Prin urmare, router-ul A copiază valoarea Segment List (A2::23) ce trebuie procesată în câmpul DA și trimite mai departe pachetul către B. Routerul B, decrementează valoarea SL cu 1, actualizează câmpul DA cu noua valoare Segmnet List (A4::45) și apoi trimite pachetul către C. Nodul C nu poate identifica SRH deoarece nu are capabilități SRv6, așadar doar caută adresa destinație în tabela sa de rutare și redirecționeaza pachetul către D. După primirea pachetului, router-ul D realizează un proces similar cu router-ul B, trimițând pachetul către E. Routerul E decapsulează pachetul și încheie procesul prin trimiterea acestuia către utilizatorul 2.



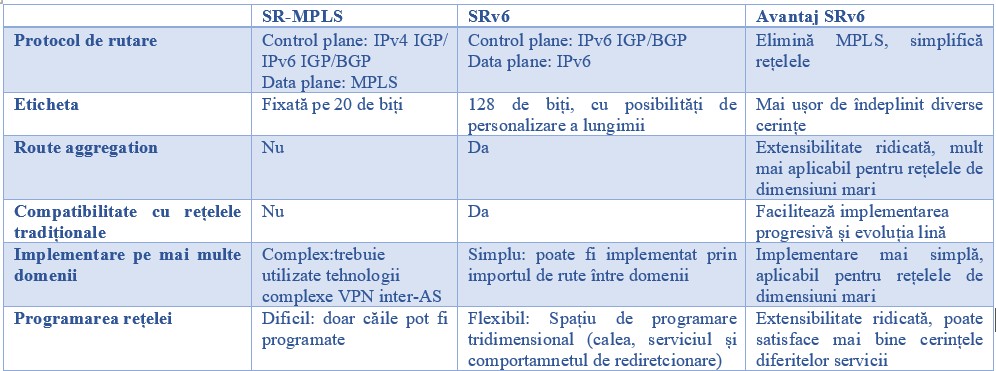
*Figură 3. 2. Redirecționarea pachetului SRv6 [12]*

SRv6 poate funcționa fie prin SRv6 Traffic Engineering Policy, care folosește mecanisumul source-routing pentru a instrui redirecționarea pachetelor într-o rețea pe baza unei liste ordonate de segmente (path information) încapsulate de nodul sursă, în cazul anterior, router-ul A, fie prin modul SRv6 Best Effort (BE), care calculează calea optimă SRv6 pe baza algoritmului SPF al IGP și utilizează

un Service SID pentru a ghida redirecționarea pachetelor pe link-uri. SRv6-BE nu are capabilități de traffic-engineering și este utilizat în general pentru furnizarea rapidă a serviciilor. [12]



*Tabel 3. 1 Diferențe între modurile de funcționare SRv6 [12]*



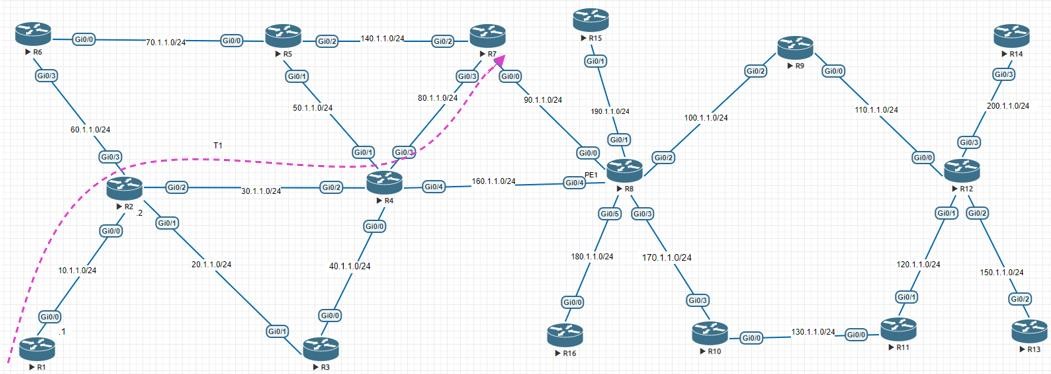
*Tabel 3. 2. Diferențe între implementările SR-MPLS și SRv6 [12]*

## Implementare

Acest capitol prezintă implementarea propriu-zisă a unor topologii de rețea simulate în mediul EVE-NG având ca scop punerea în evidență a caracteristicilor implementării tradiționale MPLS și MPLS Segment-Routing. Se vor prezenta tehnologii de inginerie de trafic utilizate pentru rețelele MPLS-VPN și metode de protecție a traficului. De asemenea, se vor pune în evidență și tehnologiile de traffic- engineering pe care Segment-Routing le aduce într-o rețea de Service Provider atunci când este implementat peste o arhitectură MPLS.

### MPLS – Traffic Engineering: Link Protection, Node Protection

Pentru a pune în evidență caracteristicile protocolului MPLS, atât elementele sale componente, cât și funcționalitățile acestuia în ceea ce privește ingineria traficului, am creat o rețea utilizând simulatorul EVE-NG. Topologia rețelei este reprezentată în figura de mai jos:

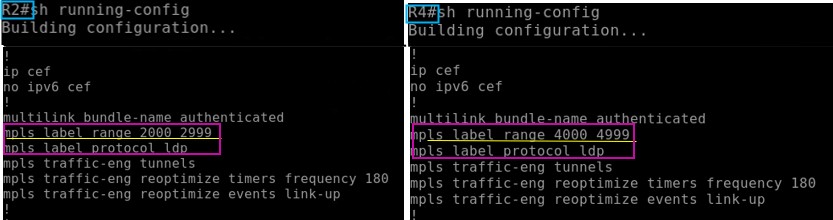


*Figură 4. 1. Topologia de rețea utilizată pentru exemplificarea MPLS – Traffic Engineering*

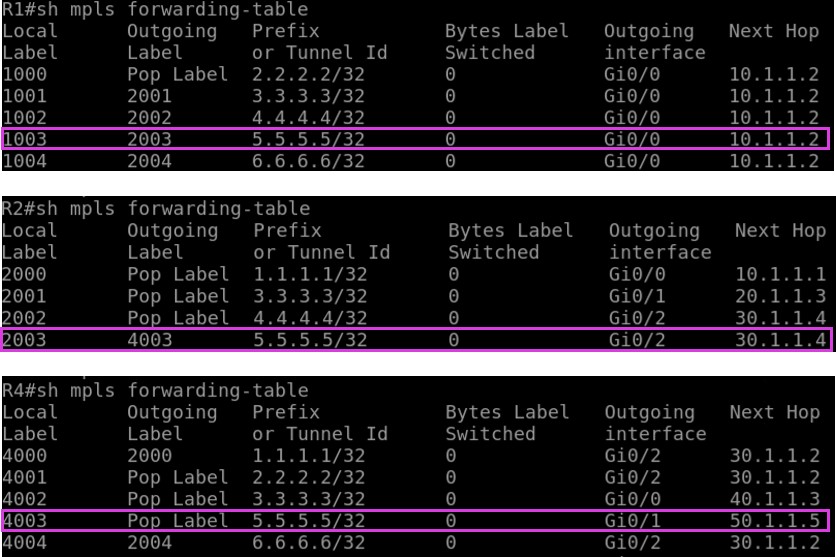
Pentru a putea configura MPLS în această rețea, primul pas a constat în configurarea IP-urilor pe interfețele fiecărui router. De exemplu, pentru R1 am accesat interfața Gi0/0 cu ajutorul comenzii

„interface Gi0/0” căreia i-am configurat o adresă IP utilizând comanda: „ip address 10.1.1.1 255.255.255.0” urmată de „no shutdown”. Am procedat similar pentru celelalte echipamente. Pentru fiecare dintre acestea, interfața Loopback 0 a fost aleasă cu mască IP de 32 de biți (ex: 1.1.1.1/32 pentru R1) deoarece aceasta este o condiție necesară pentru funcționalitatea MPLS. Interfața de loopback trebuie să fie accesibilă prin tabelul global de rutare, aceasta fiind utilizată de protocolul IGP pentru configurarea MPLS și Traffic Engineering.

De asemenea, am configurat ca protocol de rutare unul de tipul link-state, și anume OSPF. Înainte de a activa mpls în modul de configurare globală a routerelor, am stabilit câte un interval de valori pentru etichetele ce vor fi alocate de către fiecare router. Etichetele sunt alocate în mod implicit cu valori cuprinse în intervalul „16 -1048575”, însă pentru a identifica mai ușor etichetele alocate de fiecare dispozitiv, am stabilit un range de valori intuitiv pentru fiecare router: pentru R1 de la 1000 la 1999, pentru R2 de la 2000 la 2999, etc. In figura 4.2 putem observa o secvență din configurația dispozitivelor R2 si R4, intervalul de etichete alocat, precum și protocolul de distribuție utilizat, LDP.



*Figură 4. 2. Secvență din configurația echipamentelor R2 și R4*

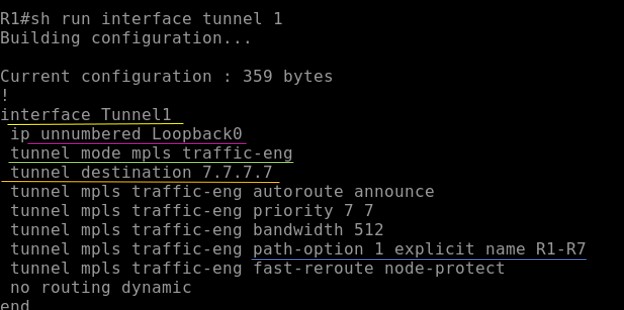
Am activat MPLS atât în modul global de configurare, cât și pe interfețe utilizând comanda „mpls ip”. Pentru a analiza modul în care etichetele sunt alocate, dar și operațiunile de „swap”, „push” sau „pop” pe care fiecare router le face asupra acestora vom lua în cosiderare următoarea situație: să presupunem că R1 trebuie să trimită trafic către R5 (5.5.5.5/32), pe calea R1-R2-R4-R5. În tabela de redirecționare MPLS, figura 4.2, putem observa cum sunt gestionate pachetele MPLS primite.

*Figură 4. 3. Tabela de redirecționare MPLS*

R1 are ca și etichetă corespunzătoare pentru IP-ul interfeței de loopback a lui R5, 5.5.5.5/32, eticheta locala „1003”. Acesta trimite pachetul etichetat cu „2003” pe interfață Gi0/0 către Next Hop

10.1.1.2. R2 efectuează operațiunea de „swap”, înlocuiește eticheta „2003” cu „4003” și trimite pachetul pe interfața Gi0/2 către Next Hop 30.1.1.4. În cele din urmă, după cum putem observa și din figură, în câmpul „Outgoing Label” operațiunea efectuată este „pop” atunci când pachetul ajunge la R4, fiind penultimul router din calea de transmitere a pachetului.

Primul tunel configurat este Tunnel 1 care face legătură între R1(headend) și R7(tailend). Pentru buna funcționalitate a acestuia am avut în vedere parcurgerea mai multor etape. Inițial am configurat pe R1, în modul de configurare globală, o interfață tunelului MPLS-TE care să fie nenumerotată deoarece tunelul reprezintă o legătură unidirecțională, specificând totodată și IP-ul destinație, interfața de loopback a router-ului 7. Am setat modul de încapsulare a tunelului mpls, specificând utilizarea acestuia pentru mpls traffic-engineering și am setat o cale LSP IP explicită de la R1-R2-R4-R7. Pentru ca tunelul configurat pe head-end router, R1, să devina UP, este necesar ca toate routerele de pe calea stabilită să aibă activat în modul global de configurare, pe interfețe și sub procesul OSPF „mpls traffic-engineering tunnels”. Toate aceste etape de configurare pot fi vizualizate cu ajutorul comenzii „show run interfate tunnel 1” după cum se poate observa si în figura următoare:



*Figură 4. 4. Configurația Tunelului 1*

Odată ce tunelul este configurat statusul său devine UP, calea este validă și este semnalizat:



*Figură 4. 5. Starea Tunelului T1*

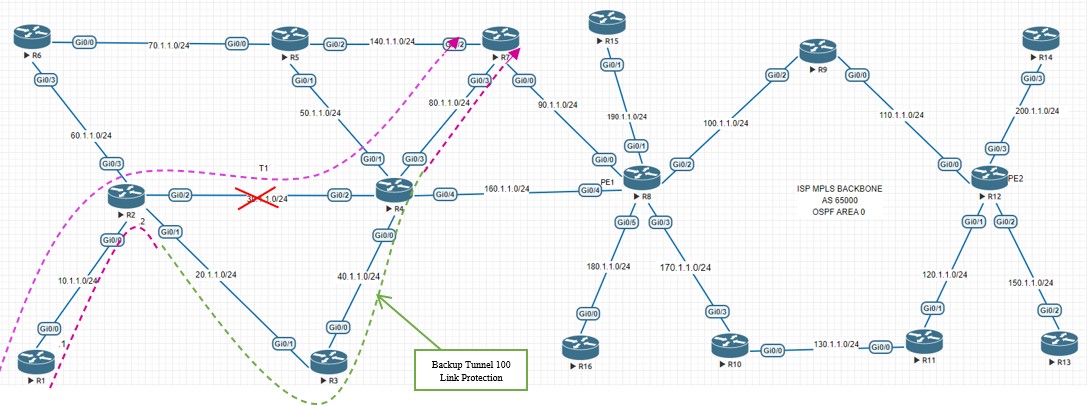
TE este de obicei activat în rețeaua de bază, unde capacitatea link-urilor este mare. Daca un link sau un router eșuează, traficul este redirecționat în jurul punctului unde s-a produs defecțiunea. Această redirecționare are loc relativ rapid pentru traficul IP și MPLS. Cu toate acestea, chiar dacă redirecționarea durează doar câteva secunde, ar putea însemna că tot traficul poate fi redus până în punctul de eșec din cauza capacității mari a legăturilor. Pentru un anumit trafic, cum ar fi Voice over IP (VoIP), acest lucru ar putea fi devastator pentru serviciu. De aceea este necesară asigurarea protecției link-urilor și a routerelor cu TE. Un tunel de rezervă poate fi creat în prealabil. Acest lucru înseamnă că nu apare o întârziere la semnalizarea tunelului de rezervă atunci când dispozitivul sau link-ul protejat eșuează. [3]

Link Protection și Node Protection sunt două metode utilizate pentru protecția cu TE. Acestea au în comun faptul că redirecționarea traficului se face cât mai aproape de punctul de defecțiune, astfel timpul de redirecționare a LSP-urilor de protecție pe tunelul de rezervă e de ordinul milisecundelor. Acest mecanism de protecție a link-urilor și a routerelor în MPLS Traffic-Engineering este numit Fast Reroute (FRR).

Cu Link Protection, un anumit link utilizat pentru traffic-engineering este protejat. Acest lucru înseamnă că toate tunelurile TE care traversează acest link sunt protejate prin tunelul de rezervă. Aceasta tehnică se mai numește și Facility Backup. Tunelul de rezervă se mai numește și Next-Hop Bypass Tunnel (NHOP) și începe întotdeauna în punctul de reparație locala, Point of Local Repair (PLR). Tunelul de rezervă întotdeauna se conectează cu router-ul next-hop, numit și Merge Point (MP), este considerat un punct de îmbinare deoarece acesta face legătura între tunelul protejat și tunelul de rezervă. Tunelul de

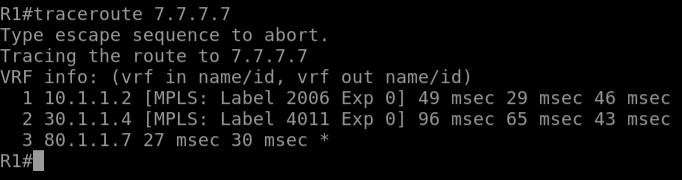
rezervă este configurat explicit și semnalizat de RSVP. Este important să menționam că tunelul de rezervă trebuie să fie utilizat temporar. Protecția este temporară deoarece eșecul link-ului declanșează PLR să trimită un mesaj „PathErr” către routerul principal al tunelului TE. Când headend router al tunelului TE protejat primește mesajul PathErr, recalculează o nouă cale pentru tunel și o semnalează. [3]

Am considerat un scenariu în care link-ul dintre R2 și R4 ar eșua, așadar este necesară o cale alternativă, un tunel de rezervă asftel încât traficul să nu fie afectat. Alternativa optimă constă în redirecționarea traficului către cel mai apropiat router, adică R3.



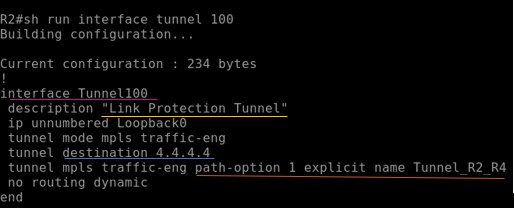
*Figură 4. 6. Backup Tunnel 100 - Link Protection*

Figura următoare ilustrează traseul pe care îl urmează traficul de-a lungul tunelului T1 într-un scenariu de bună funcționalitate.



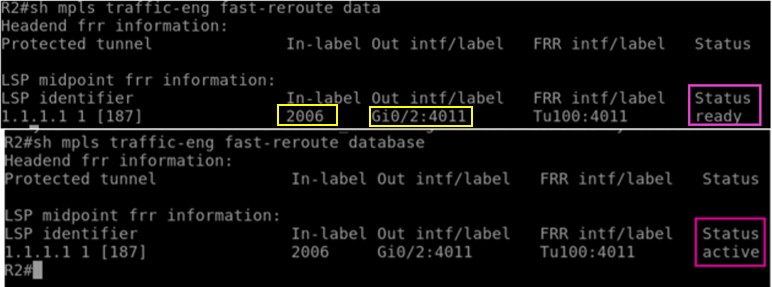
*Figură 4. 7. Calea tunelului T1*

Pentru a configura Link Protection FRR trebuie să configurăm inițial un tunel de rezervă care va fi utilizat atunci când calea principală eșuează. Link-ul de rezervă (backup link) trebuie să fie configurat în locul de unde stabilim tunelul de rezervă, în acest caz pe Routerul 2 deoarce aici avem o cale redundantă optimă. Având în vedere că link-ul dintre R2 și R4 se defectează, automat avem nevoie de o altă cale care să înceapă de la R2 și să ajungă la R4. Există două astfel de posibilități: R2-R4-R5-R4 și R2-R3-R4. Vom alege calea cea mai scurtă, adică utilizând router-ul R3. Este absolut necesar să specificăm pe interfața protejată, Gi0/2 a routerului R2, comanda „mpls traffic-eng backup-path Tunnel 100”. După configurarea pe R2 a tunelului de rezervă, Tunnel 100, asociat cu eșecul link-ului Gi0/2, este necesar un ultim pas pentru ca FRR Link Protection să funcționeze. Trebuie să activăm redirecționarea rapidă pe tunelul original TE, și anume T1, configurat pe R1.



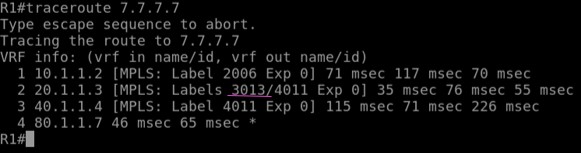
*Figură 4. 8. Configurația Tunelului de backup 100*

În figura 4.9 putem observa în câmpul „Status” că tunelul este „ready”, adică este activat, ceea ce înseamnă că monitorizează legătură pe care trebuie să o protejeze, este semnalizat, dar nu este utilizat pentru FRR în acest moment. De asemnea, output-ul ne oferă informații despre eticheta cu care vă ajunge pachetul la router, 2006, și eticheta cu care ar trebui să fie redirecționat, 4011, dar în caz de defecțiune nu se poate întâmpla acest lucru deoarece calea către routerul R4 este întreruptă. Așadar trebuie să protejăm și eticheta 4011. De aceea, pachetul va avea două etichete pe calea de rezervă și anume 4011 și eticheta impusă de Tunelul 100, 3013.



*Figură 4. 9. Trecerea de la starea "ready" la "active" odată cu eșuarea link-ului protejat*

Odată ce am inactivat interfața Gi0/2 de pe R2, tunelul de rezervă își schimbă statusul din „ready” în „active”. Figura de mai jos ilustrează un output ce indică calea parcusă de trafic după eșuarea link- ului. Traseul este cel de culoare verde indicat în figura 4.6 de mai sus. De asemenea, putem observa eticheta suplimentară atașată de backup tunnel: 2 20.1.1.3 [MPLS: Labels 3013/4011 Exp 0] 35 msec 76 msec 55 msec. Atunci când interfață Gi0/2 devine up, Tunnel 100 intră din nou în statusul „ready”, iar traficul de pe Tunnel 1 reia calea explicită configurată inițial.

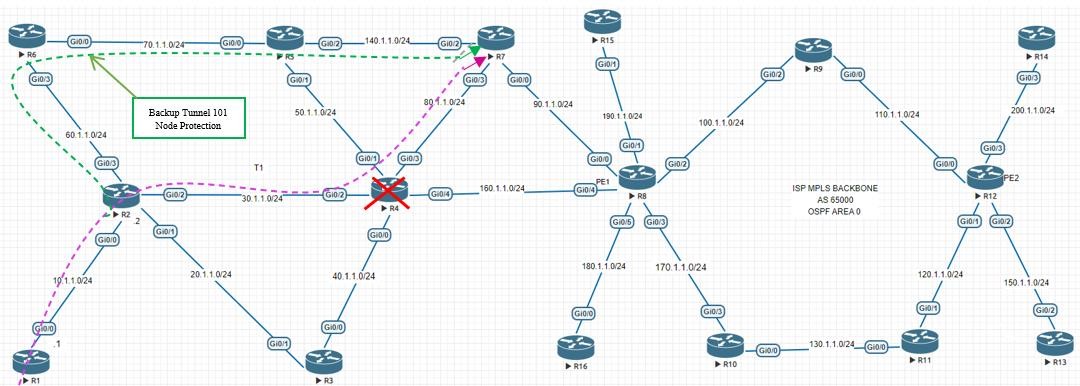


*Figură 4. 10. Calea de backup - T100 – Link Protection*

În cazul Node Protection FRR avem în vedere protejarea unui router, nu doar a unui link. Node Protection funcționează prin crearea unui tunel de rezervă next-next-hop (NNHOP). Un tunel de rezervă NNHOP este un tunel care se află la un hop în spatele routerului protejat. [3]

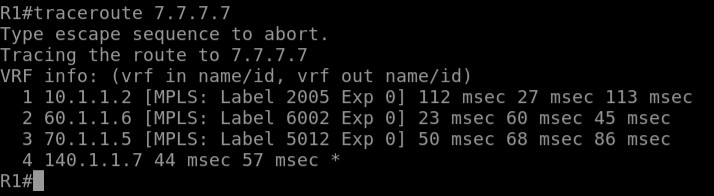
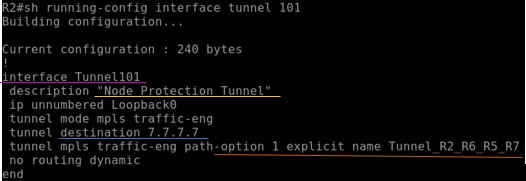
Pentru a evidenția Node Protection FRR, am considerat un scenariu în care routerul R4 ar suferi o defecțiune, ținând seama că tunelul principal este T1. Tunelul 101 este cel de rezervă, secundar, ales asftel încât să se evite complet R4. Așadar, tunelul T101 parcurge calea R2-R6-R5-R7, cea de culoare

verde din 4.11. Ca și în cazul Link Protection, odată ce tunelul este configurat statusul său devine „ready”, urmând că atunci când se semnalizează defecțiunea routerului R4, acesta să devină „active”.



*Figură 4. 11. Backup Tunnel 101 - Node Protection*

Înainte de a simula defecțiunea produsă pe R4 trebuie menționat că traficul urmează traseul menționat în figura 4.1. Outputul din figura 4.13 indică calea parcusă de trafic după eșuarea lui R4.



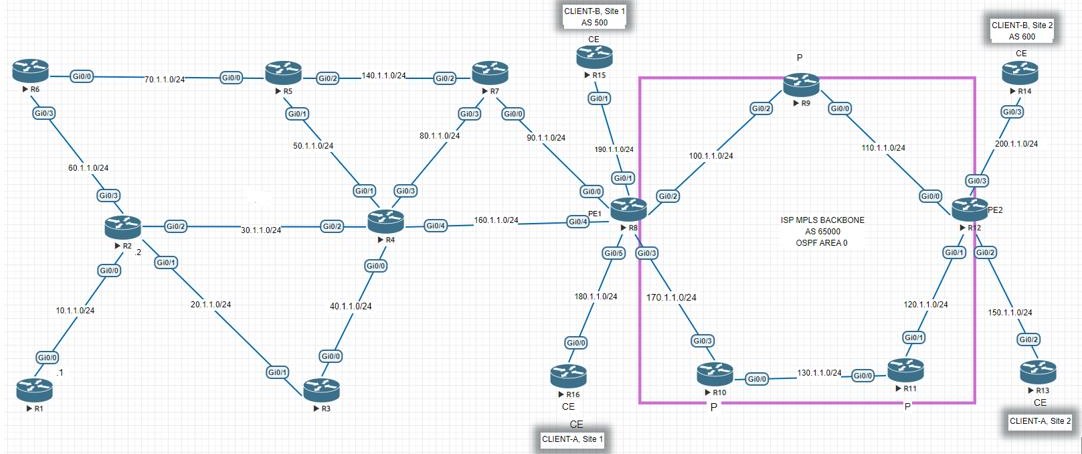
*Figură 4. 12. Configurația Tunelului de backup 101 Figură 4. 13. Calea de backup - T101 - Node Protection*

# MPLS – Traffic Engineering: MPLS-VPN

MPLS-VPN (MPLS Vitual Private Networks) este cea mai populară și răspândită implementare a tehnologiei MPLS, popularitatea sa crescând exponențial de când a fost inventată. MPLS-VPN poate oferi scalabilitate și poate împarți rețeaua în rețele mici separate, ceea ce este adesea necesar în rețelele mai mari ale întreprinderilor, unde infrastructura IT comună trebuie să ofere rețele separate departamentelor individuale. De obicei, VPN-ul aparține unei singure companii și are mai multe site-uri interconectate. [3]

O rețea privată virtuală (VPN) este o rețea privată printr-o infrastructura comună. VPN-ul poate furniza comunicații la nivel OSI 2 sau 3. VPN-ul aparține de obicei unei singure companii și are mai multe site-uri interconectate prin infrastructura comună a furnizorului de servicii (ISP- Internet Service Provider). Rețeaua privată necesită ca toate site-urile clienților să se poată interconecta și să fie complet separate de alte VPN-uri, aceasta fiind cerința minimă de conectivitate. [3]

Pentru a ilustra importanța ingineriei traficului într-o rețea privată MPLS am considerat un scenariu în care doi clienți A și B doresc să interconecteze fiecare două site-uri aflate la distanțe geografice considerabile, iar comunicația dintre dispozitivele clientului A și cele ale clientului B să nu fie posibilă.



*Figură 4. 14. MPLS VPN*

Routerele R8 și R12 sunt routere de tipul Provider Edge (PE), caracteristica lor principală fiind că au conexiune directă cu routerele Customer Edge (CE), în scenariul considerat R16 și R13. Dispozitivele R9, R10 și R11 sunt routere Provider (P), acestea neavând conexiune directă la routerele clientului (CE). Am considerat implementarea modelului VPN peer-to-peer și backbone-ul MPLS ca fiind alcătuit din R8, R9, R10, R11 și R12. Așadar, routerele P și PE necesită rularea protocolului MPLS întrucât acestea trebuie să poată distribui etichete și să trimită pachete etichetate. Routere CE au o conexiune directă L3 cu routerul PE și nu rulează MPLS.

Deoarece routerele CE și PE interacționează la nivelul 3 am decis configurarea rutelor statice pentru site-urile clientul A și a protocolului BGP pentru clientul B, astfel încât routerele CE să trimită traficul către routerele PE, acesta fiind sigurul peer în afara propriului site. Este important să menționam că în cazul modelului VPN peer-to-peer, routerul CE nu se conectează la niciunul dintre routerele CE de la celelalte site-uri din rețeaua furnizorului de servicii. R15 aparține sistemului autonom cu numărul 500, iar R14 sistemului autonom 600. ASN (Autonomous System Number) este utilizat în principal pentru protocolul BGP și identifică în mod unic o rețea sub o singură administrație tehnică. Acest număr este necesar atunci când dorim să rulăm BGP și să stabilim o legătură cu furnizorul de servicii.

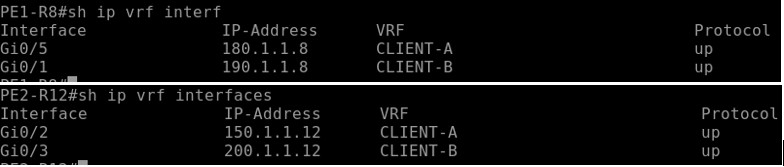
Având în vedere că implementarea VPN cu ajutorul MPLS aduce o soluție scalabilă în ceea ce privește distribuirea rutelor în rețeaua furnizorului de servicii, nu a fost necesară implementarea BGP pe routerele P, astfel încât acestea să nu fie încărcate cu informații de rutare pentru rutele VPN. În plus, acestea nu vor avea tabelul de rutare al clienților folosind două etichete MPLS. Ca atare, informațiile VPN sunt primite numai pe routerele Provider Edge ale rețelei MPLS VPN, doar acestea două rulând BGP.

Pentru implementarea cu succes a rețelei MPLS VPN din figura de mai sus, am avut în vedere câteva elemente cheie pe care le-am configurat pe routerele PE: Virtual Routing Forwarding (VRF), Route Distinguisher (RD), Route Target (RT), propagarea rutelor prin MP-BGP și redirecționarea pachetelor etichetate.

Routerele care fac parte din MPLS Backbone, adică din rețeaua ISP, necesită configurarea unui protocol de rutare de tipul link-state, și anume OSPF. Am stabilit un range de etichete intuitiv pentru fiecare router și am activat MPLS și LDP atât pe interfețe, cât și în modul de configurare globală. Am configurat BGP atât pe dispozitivele clientului B, cât și pe routerele PE. Într-o rețea reală ISP, se ia în

considerare un număr foarte mare de rute și de aceea este necesară utilizarea unui protocol de rutare stabil pentru transportul acestora, BGP fiind protocolul ideal care să îndeplinească această cerință.

Pentru ca rutele clienților să fie conținute în tabele de rutare separate pe routerele PE, am configurat câte un VRF pentru fiecare client pe acestea. VRF este o instanță de rutare și redirecționare specifică VPN. Un router PE are o instanță VRF pentru fiecare VPN atașat deoarece rutarea ar trebui să fie separată și privată pentru fiecare client. Ca atare, toate pachetele IP primite pe intrefața VRF sunt identificate fără ambiguitate ca aparținând acelui VRF. [3]

De asemnea, este absolut necesar să precizăm pe routerele PE care interfață are rolul de redirecționarea traficului pentru un vrf specific:

*Figură 4. 15. Interfețele cu rol de redirecționare a traficului din VRF-uri*

Routerele PE primesc rute IPv4 de la routerele CE. Aceste rute IPv4 de pe site-ul VPN sunt introduse în VRF routing table. Totuși, atunci când rutele IPv4 trebuie exportate în rețeaua furnizorului de servicii, sub procesul BGP nu există încă nimic care să diferențieze rutele în cazul în care există mai multe vrf-uri pe routerele PE. Prin urmare, apare necesitatea configurării unui RD, route-distinguisher.

Prefixele VPN sunt propagate în rețeaua MPLS VPN prin Multiprotocol BGP (MP-BGP). Atunci când BGP transportă prefixele IPv4 în rețeaua furnizorului de servicii, acestea trebuie să fie unice. Dacă clienții ar avea adrese IP suprapuse, rutarea s-ar efectua în mod eronat. De aceea Route Distinguisher (RD) este soluția de care avem nevoie. Ideea ce stă la baza RD-urilor este că fiecare prefix de la fiecare client primește un identificator unic pentru a distinge același prefix de diferiți clienți. [3]

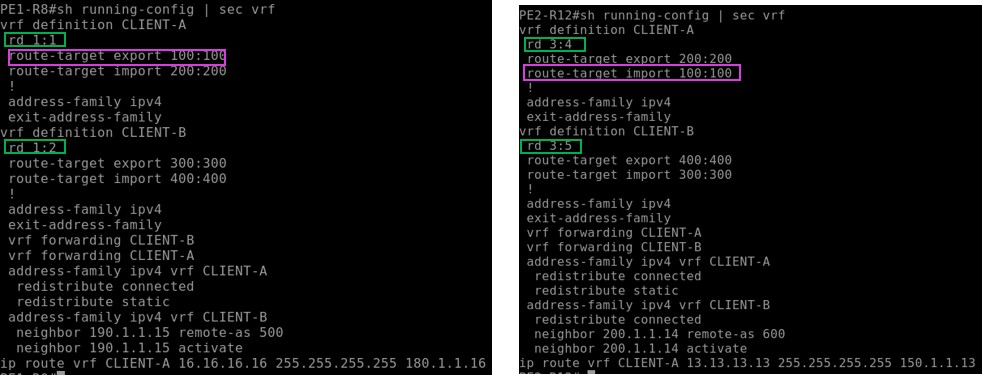
Aceste rute IPv4 sunt atașate RD-ului care este atribuit acelui VRF. Astfel, ele devin rute vpnv4, care sunt apoi introduse în MP-BGP. BGP se ocupă de distribuirea acestor rute vpnv4 către toate routerele PE din rețeaua VPN MPLS. Pe routerele PE, din rutele vpnv4 este îndepărtat RD-ul și prefixul este introdus în tabela de rutare VRF ca rută IPv4. Este important să menționam că un RD nu este un identificator pentru VPN și nu indică cărui VRF îi aparține prefixul. Fiecare instanță VRF de pe routerul PE trebuie să aibă un RD alocat.

Deoarece furnizorul de servicii care rulează rețeaua MPLS VPN rulează BGP într-un sistem autonom, iBGP rulează între routerele PE. Propagarea de la eBGP, care rulează între routerele PE și CE, la MP-iBGP în rețeaua VPN MPLS și invers este automată și nu necesită configurare suplimentară. [3]

Am avut în vedere utilizarea interfețelor de loopback, alese și acestea în mod intuitiv, și accesibilitatea lor prin OSPF în backbone-ul mpls. PE1 și PE2 ridică o adiacență iBGP, devin iBGP neighbors, iar PE1 trimite rutele vpnv4 către PE2. Atunci când rutele vpnv4 ajung la PE2, acesta va șterge valoarea RD alocată de PE1 deoarece semnificația acesteia este doar locală. Pentru că PE2 are două VRF- uri configurate, apare problema legată de lipsa unui mecanism prin care acesta să știe cărui vrf îi aparține acel prefix ipv4. Soluția constă în configurarea de Route Target (RT).

Conceptul de a avea mai multe site-uri care solicită comunicarea peste o rețea MPLS VPN a necesitat dezvoltarea Route Target (RT). Un RT este o comunitate extinsă BGP (BGP extended community) care indică rutele care ar trebui importate din MP-BGP în VRF. Exportarea unui RT (route- target export) înseamnă că ruta vpnv4 exportată primește o valoare pentru RT atunci când este

redistribuită din tabelul de rutare VRF în MP-BGP de către routerul PE. Importarea unui RT presupune ca ruta vpnv4 primită de la MP-BGP să fie verifcată asftel încât să existe o corespondență între valoarea de import a RT și informația din tabela VRF. [3]

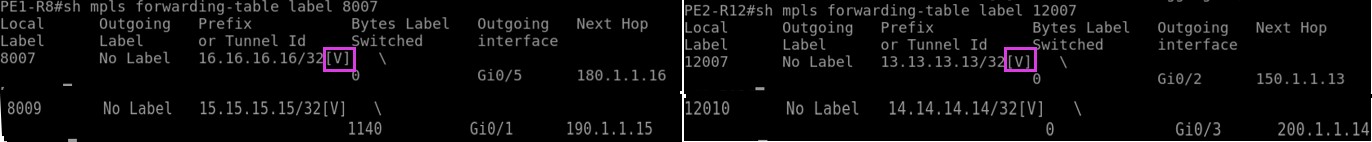
Asftel, pentru a distinge cărui VRF îi aparțin rutele respective am configurat RT de import și export. Când PE1 trimite rutele vpnv4, RD și prefixul, va trimite și o valoare RT de export care trebuie să coincidă cu valoarea RT de import a lui PE2, după cum se poate observa și în figura 4.16. Procesul este similar în cazul trimiterii rutelor vpnv4 de la R12 către R8, PE2 către PE1. După cum am menționat anterior, valoarea RD are semnificație locală, deci nu trebuie să fie neapărat identică pe ambele dispozitive: am ales pentru PE1, rd 1:1 și pentru PE2, rd 3:4. Configurația din secțiunea vrf a routerelor PE1 și PE2 poate fi vizualizată atât în figura 4.16, cât și în [Anexa 1](#_bookmark117) de la finalul lucrării.

*Figură 4. 16. Configurația routerelor PE1 și PE2 – secțiunea vrf*

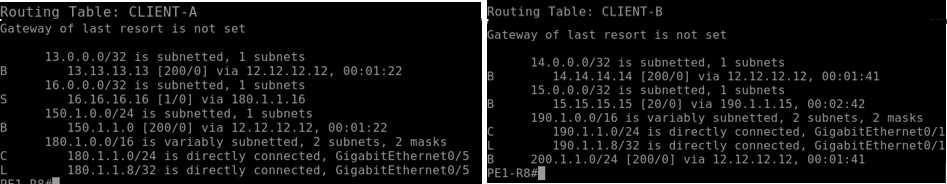
Pentru a verifica conectivitatea dintre cele două site-uri ale clientului A, vom utiliza comenzi de tipul show, ping și traceroute:

* În tabela de redirecționare mpls ruta pentru VPN este identificată prin [V], iar eticheta locală generată de routerele PE prin MP-BGP Process sunt etichetele VPN (8007 în cazul PE1 pentru clientul A și 8009 pentru clientul B, 12007 în cazul PE2 pentru clientul A și 12010 pentru clientul B). Câmpul

„Outgoing Label” are indicat „No Label” deoarece routerul PE care primește pachetul trebuie să elimine ultima etichetă pe care o conține stiva, și anume VPN label, pentru ca pachetul să fie redirecționat către client ca pachet IPv4, fără etichetă mpls deoarece pe dispozitivele clienților nu am configuarat mpls.

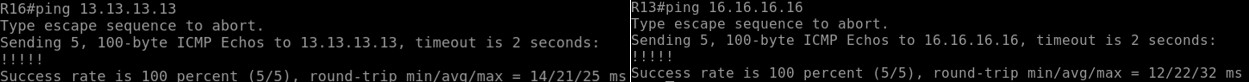


*Figură 4. 17. Rutele pentru VPN afișate în tabela de redirecționare MPLS*

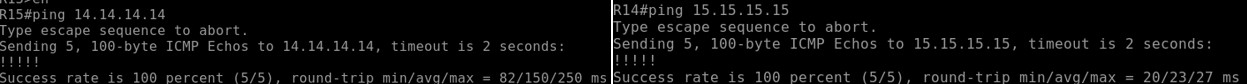


*Figură 4. 18. Tabelele de rutare ale clienților A și B*

* + Avem conectivitate între dispozivele clienților din cele două site-uri

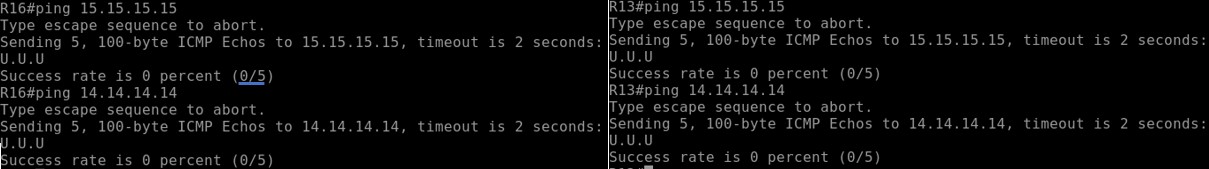


*Figură 4.19. Conectivitate între site-urile clientului A*

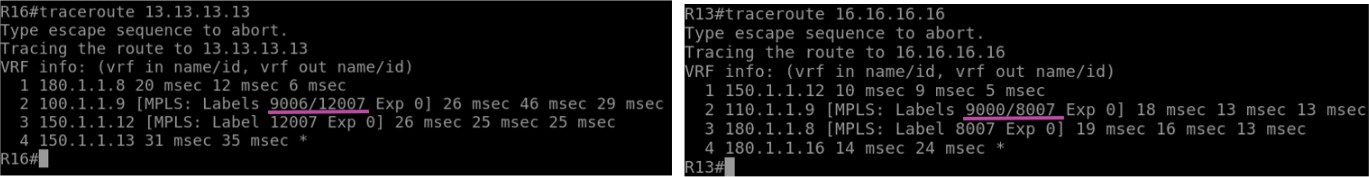


*Figură 4. 20. Conectivitate între site-urile clientului B*

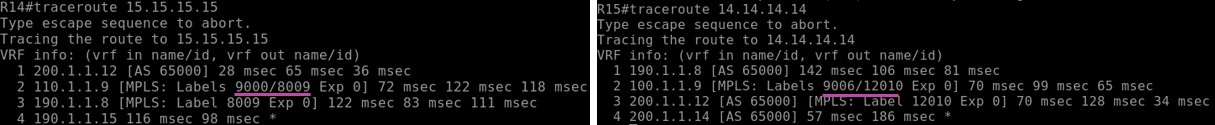
* + Nu avem conectivitate între clientul A și clientul B deoarece dispozitivele fiecăruia se află într- un vrf diferit:



*Figură 4. 21. Lipsa conectivității dispozitivelor ce aparțin clienților diferiți*

* + Cu ajutorul comenzii traceroute putem observa redirecționarea pachetului de-a lungul mpls backbone. În figurile de mai jos sunt ilustrate situațiile comunicării dintre cele două site-uri ale clientului A, respectiv B:

*Figură 4. 22. Calea de comunicație între site-urile clientului A*



*Figură 4. 23. Calea de comunicație între site-urile clientului B*

Când un pachet IP trimis de CE, corespunzător clientului A de exemplu, intra în routerul PE1, acesta caută adresa IP de destinație în tabela de rutare vrf pentru clientul A. Acesta găsește vrf-ul corect luând în considerare interfața pe care a venit pachetul și verifică cu ce tabelă vrf este asociată acea interfață. O intrare specifică în tabela vrf indică de obicei că trebuie adăugate două etichete. Dacă routerele PE1 și PE2 ar fi fost conectate direct, pachetele ar fi necesitat o singură etichetă, doar eticheta VPN. Routerul PE1 adaugă pachetului eticheta VPN 12007, urmând ca mai apoi să adauge peste aceasta eticheta 9006, fiind considerată top label. Pachetul IPv4 părăsește routerul PE1 cu două etichete deasupra acestuia și ajunge la R9. Aici, având în vedere că R9 este ultimul router P din LSP, eticheta de sus va fi extrasă și pachetul intră în routerul PE2 doar cu eticheta VPN în stiva de etichete. După cum am menționat și anterior, routerul PE2 de ieșire va scoate eticheta VPN și va trimite pachetul IPv4 către client.

### MPLS – Traffic Engineering : MPLS-TE si MPLS VPN

Există două variante de implementare a tunelurilor TE peste o rețea MPLS VPN: 1.Tunelurile TE între routerele PE

2.Tuneluri TE între un router PE și un router P

În primul caz se ia în vedere configurarea a două tuneluri TE, câte unul pentru fiecare direcție, între perechea de routere PE, traficul VRF circulând prin acestea. Pachetele au două etichete: eticheta de sus (top label) este eticheta TE, iar eticheta de jos este eticheta VPN. Dacă TE este activat pe toate routerele P ale furnizorului de servicii, dar LDP nu e activat, tot traficul VRF trebuie să circule peste tunelurile TE. În caz contrar, traficul are doar eticheta VPN și nu ajunge niciodată la routerul PE de ieșire. LDP nu trebuie să ruleze pe interfața tunelului TE în acest caz, unde routerele PE sunt routere head-end și tail-end. [3]

În al doilea caz, atunci când un router P este routerul tail-end al tunelului în loc de un router PE, trebuie să avem în vedere două aspecte esențiale: LDP să fie activat pe toate link-urile și să existe o sesiune LDP între routerele head-end și tail-end ale TE LSP.

Prima cerință este necesară deoarece pachetele trebuie să fie schimbate cu etichetă către routerul PE. În caz contrar, pachetele devin neetichetate, iar etichetele IGP și VPN se pierd. Fără eticheta VPN, pachetele nu pot fi niciodată comutate pe interfața VRF corectă, nu pot ajunge la PE de ieșire, deoarece o căutare a adresei IP de destinație a pachetului duce fie la eliminarea pachetului, fie la o rutare incorectă. Acesta este rezultatul faptului că rutele IP VRF sunt necunoscute pe routerele P. De aceea, trebuie să ne asiguram că LDP rulează între routerele P.

A doua cerință este puțin mai subtilă întrucât routerul tail-end al tunelului TE LSP trimite o etichetă NULL implicită către LSR-ul din amonte. Dacă routerul tail-end este PE, este în regula, deoarece pachetului i se extrage top label la penultimul router și sosește cu eticheta VPN astfel încât PE va ști să comute către interfața corectă. În schimb, atunci când tail-end este P, apar probleme întrucât acesta trimite o eticheta NULL implicita către routerul său din amonte, această acțiune având ca rezultat că pachetele ajung la el cu eticheta VPN ca top label. În cazul acesta, routerul P fie renunță la pachet deoarece eticheta VPN îi este necunoscută, fie redirecționează pachetul în mod eronat.

Așadar, sesiunea LDP ce trebuie stabilită între routerele PE și P se poate face prin configurarea a două tuneluri între routerul PE și P, câte unul pentru fiecare direcție cu LDP activat pe interfețele tunelului TE. Acest lucru are ca rezultat faptul că pachetele VPN vor avea acum 3 etichete: eticheta de jos (bottom

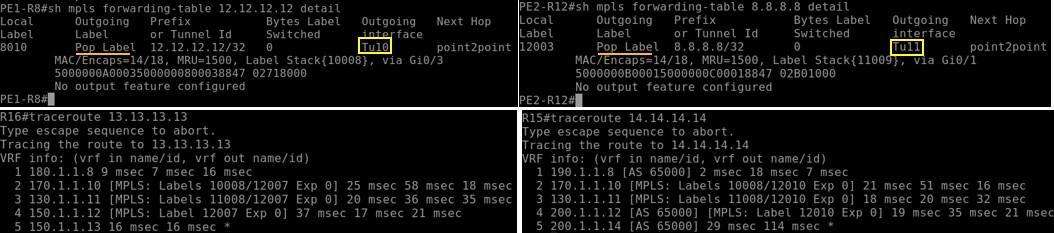
label) va fi eticheta VPN, eticheta din mijloc (middle label) este cea care este anunțată în sesiunea LDP vizată, iar eticheta de sus (top label) va fi eticheta tunelului TE. [3]

Am decis ca tunelurile principale să fie implementate considerând prima variantă, între PE1 și PE2 vom avea tunelul T10, cel ilustrat cu portocaliu în figura de mai jos, iar de la PE2 la PE1 tunelul T11, cel de culoare roșie. Am ales calea tunelului T10 ca fiind R8-R10-R11-R12, iar a lui T11 R12-R11- R10-R8. Înainte de configurarea acestora, traficul se trimitea pe calea R8-R9-R12, această cale fiind calculată cu ajutorul algoritmului SPF (Shortest Path First) al protocolului OSPF implementat în MPLS Backbone.



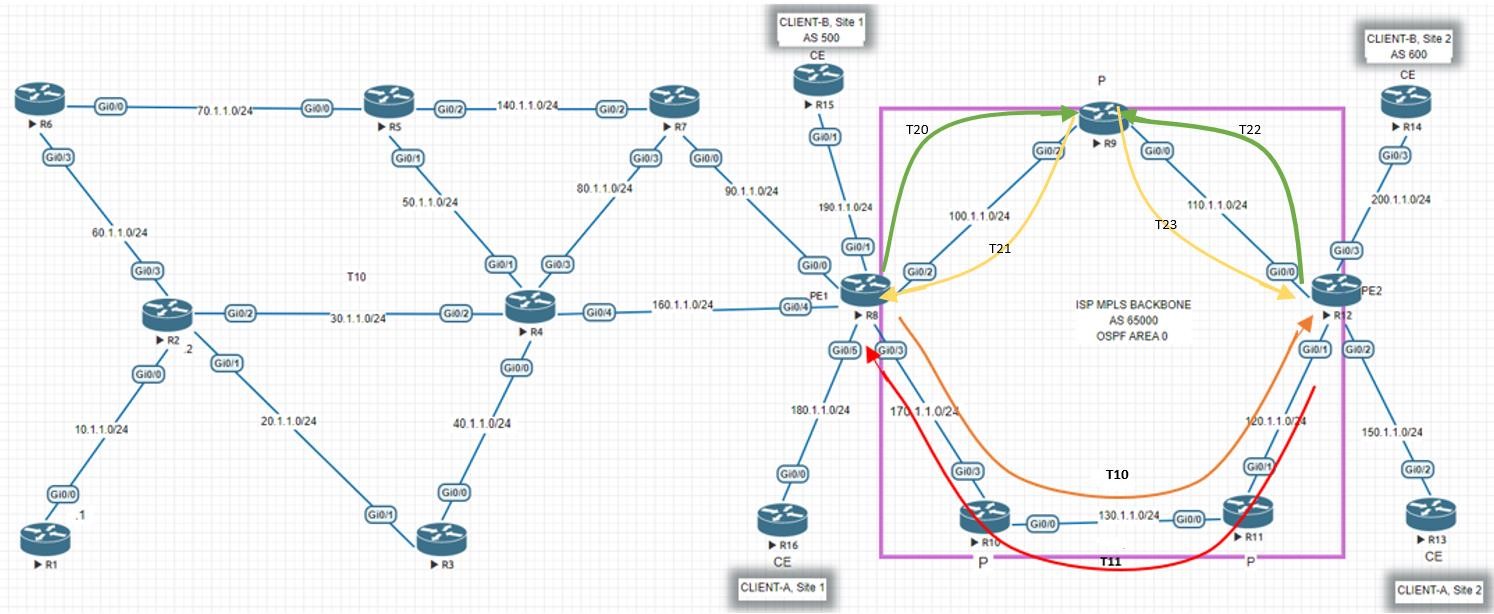
*Figură 4. 24. Ilustrarea tunelurilor T10 si T11*

În cele din urma, putem observa că traficul nu mai urmează calea stabilită de SPF, ci acesta este trimis direct prin tunel:

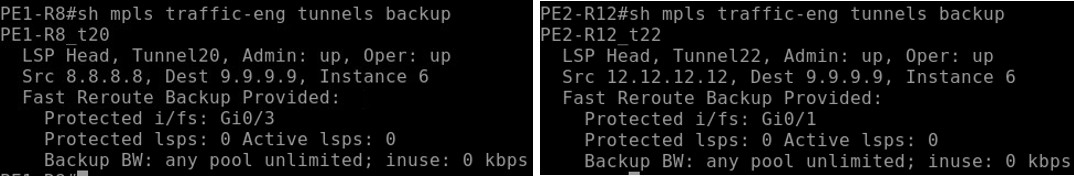


*Figură 4. 25. Calea de transmitere a traficului prin T10, respectiv T11*

Este necesar să luam în calcul protecția link-urilor astfel încât traficul transmis între site-urile celor doi clienți să nu fie afectat în cazul apariției unei defecțiuni. R9 va fi dispozitivul către care se va redirecționa traficul în cazul în care apare o defecțiune pe interfața Gi0/3 a lui PE1 sau pe interfața Gi0/1 a lui PE2. Astfel vom implementa a doua metodă pentru Traffic-Engineering într-o rețea MPLS VPN: de la PE1 la P (R9) vom avea Tunelul 20, iar de la PE2 la P (R9) Tunelul 22. Este necesar să implementăm și pe R9 două tuneluri în direcțiile routerelor PE, 21 și 23. Acestea sunt ilustrate în figura de mai jos:



*Figură 4. 26. Ilustrarea tunelurilor de backup – T20, T21, T22, T23*

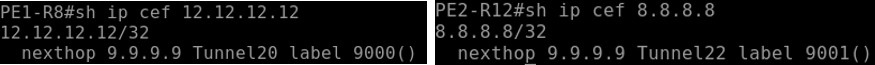


*Figură 4. 27. Informații privind tunelurile de backup T20 și T22 ale routerelor PE1 și PE2*

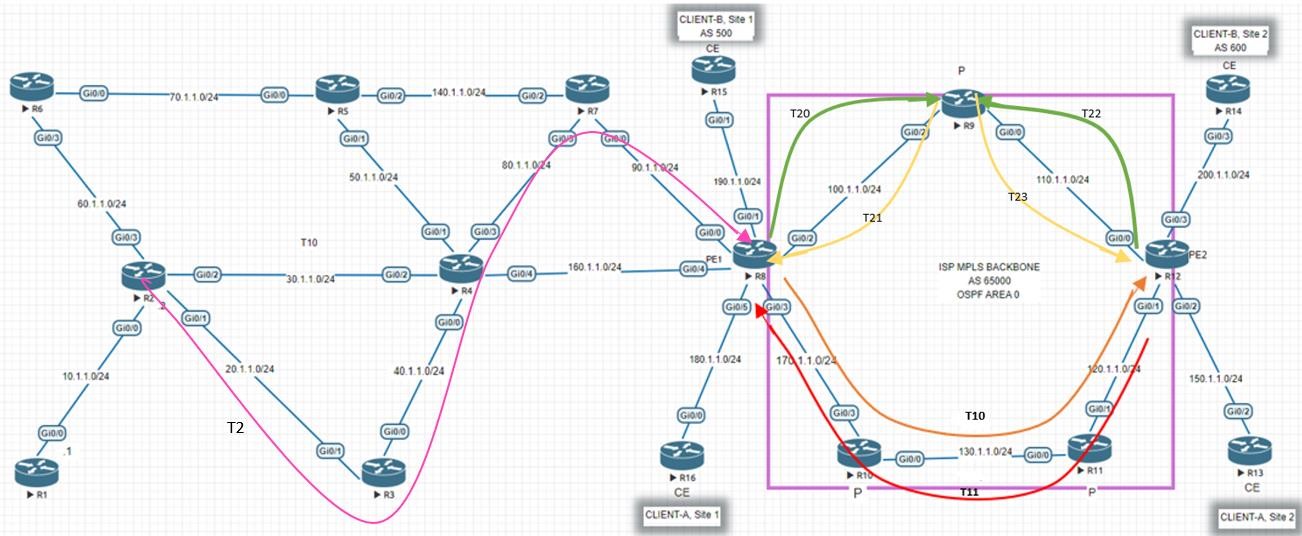
Odată ce am creat tunelurile de backup, atunci când unul dintre link-urile protejate se defectează, traficul se redirecționează pe tunelurile de rezervă:

* Link-urile protejate funcționează în parametrii normali

*Figură 4. 28. Calea de transmitere a traficului prin tunelurile T10 și T11 într-un scenariu de bună funcționalitate*

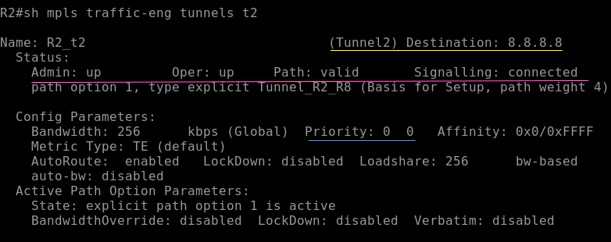
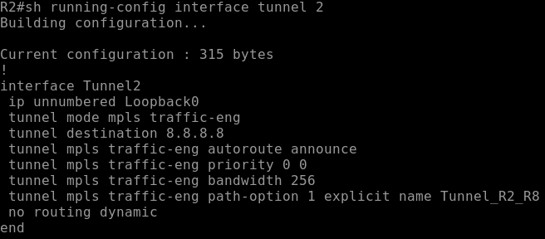
* Unul dintre link-uri se defectează

*Figură 4. 29. Calea de transmitere a traficului prin tunelurile de backup T20 și T22 în cazul defecțiunii unui link*



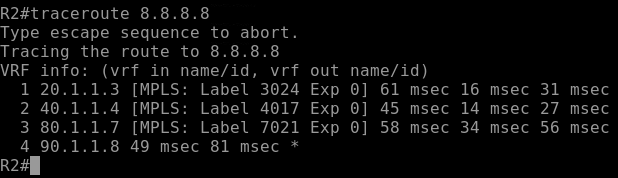
*Figură 4. 30. Ilustrarea Tunelului T2*

De asemenea, am configurat pe routerul R2 un tunel T2 cu prioritate 0, adică o importanță ridicată, având ca destinație routerul PE1, R8. Timpul de reoptimizare periodică a fost setat la 180 de secunde cu ajutorul comenzii „mpls traffic-eng reoptimize timers frequency 180”. Am activat și reoptimizarea bazată pe evenimente, „ mpls traffic-eng reoptimize events link-up”, astfel încât în cazul în care un link eșuează și apoi devine disponibil, tunelul T2 să fie reoptimizat imediat.



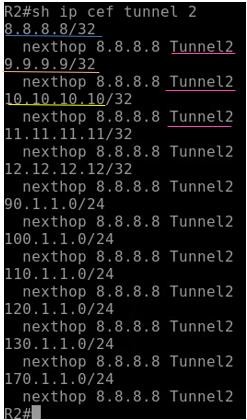
*Figură 4. 31. Configurația Tunelului T2 Figură 4. 32. Starea Tunelului T2*

În figura 4.31 sunt ilustrate comenzile utilizate pentru crearea tunelului T2, iar în figura 4.32 output-ul prezentat ne oferă informații cu privire la starea tunelului: acesta este UP, este semnalizat și utilizează o cale configurată explicit care este activă.



*Figură 4. 33. Calea de transmisie a traficului de-a lungulul tunelului T2*

În figura de mai jos putem observa tabela FIB (Forwarding Information Base) care ne indică la ce destinații putem ajunge utilizând Tunnel 2. R2 va utiliza tunelul Tunnel 2 pentru a ajunge la oricare dintre ip-urile de destinație ce aparțin ISP MPLS-Backbone cu excepția site-urilor clienților A și B din VRF-uri.



*Figură 4. 34. Tabela FIB – Tunnel 2*

### Segment-routing MPLS-Traffic Engineering: Topology-Independent Loop-Free Alternate (TI-LFA)

Unul dintre avantajele SR este tehnologia eficientă Topology-Independent Loop-Free Alternate (TI-LFA) Fast Reroute (FRR). Înainte de introducerea TI-LFA FRR, existau două tehnologii importante: Loop-Free Alternante (LFA) și Remote Loop-Free Alternante (RLFA). Totuși, complexitatea topologiilor de rețea a determinat ca aceste două tehnologii să nu mai răspundă cerințelor impuse. Astfel TI-LFA FRR bazat pe SR poate oferi protecția nodurilor și a link-urilor în orice topologie, compensând dezavantajele tehnologiilor tradiționale de protecție a tunelurilor. Dacă un link sau un router eșuează, TI- LFA FRR comută rapid traficul pe o cale de rezervă, minimizând pierderile de pachete care rezultă în timp ce routerele converg după o modificare de topologie. [13]

Soluționarea este rapidă (<50 msec) și se realizează prin utilizarea căilor de rezervă calculate anterior care sunt sigure de utilizat până când procesul de distribuție a rutelor convergente este complet. Algoritmul TI-LFA permite crearea unei căi complete explicite de-a lungul rețelei, construiește un tunel SR ce are rolul de cale de rezervă (repair path sau backup path). Tunelul TI-LFA este construit prin specificarea explicită a routerului sau a unui set de routere de backup pe care tunelul trebuie să le traverseze. Traficul este transmis pe tunel atunci când calea primară nu mai este disponibilă. [14]

Există câteva concepte de bază când vorbim de tehnologia TI-LFA:

* + - Spațiul P = set de noduri accesibile prin link-uri (cu excepția celui protejat) de la nodul rădăcină (root node sau source node) fără a traversa link-ul protejat
    - Spațiul P extins = set de noduri accesibile prin link-uri (cu excepția celui protejat) care provin de la alte noduri vecine cu nodul sursă al link-ului protejat
    - Spațiul Q = set de noduri accesibile prin link-uri (cu excepția celui protejat) de la nodul destinație al link-ului protejat
    - Nod PQ = se află atât în spațiul P extins cât și în spațiul Q, funcționând ca nod intermediar pentru traficul protejat
    - LFA = algoritm ce calculează un standby link, rulat pe un nod ce utilizează SPF pentru a calcula cea mai scurtă cale către destinație pe baza costului cel mai mic al link-urilor
    - RLFA = Remote LFA, stabilește un nod PQ de-a lungul unei căi protejate și creează un tunel între sursă și nodurile PQ pentru a oferi protecție; în cazul eșuării unui link, traficul trece automat pe calea de rezervă, ceea ce îmbunătățește fiabilitatea rețelei și reduce pierderea de pachete

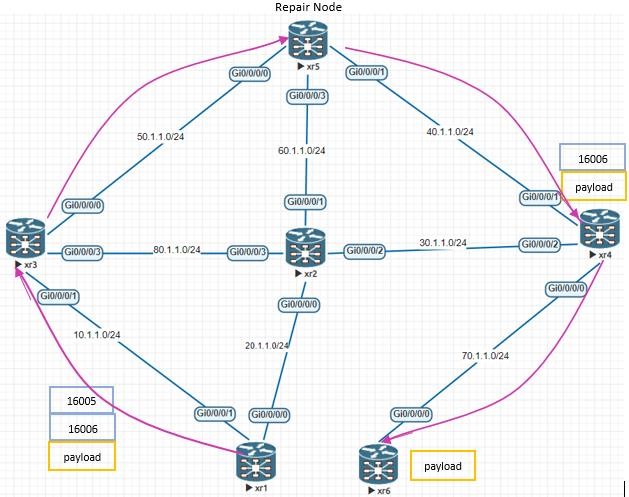
În unele scenarii, spațiul P și spațiul Q nu au vecini conectați direct, prin urmare nu poate fi calculată o cale de rezervă care să îndeplinească cerințele. În această situație, algoritmul TI-LFA calculează spațiul P și spațiul Q pe baza unei căi protejate și stabilește un tunel segmet-routing între nodul sursă și nodul PQ pentru a oferi protecție de rezervă. [13]

TI-LFA bazat pe segment-routing are următoarele avantaje:

* + - Îndeplinește cerințele de bază ale convergenței rapide IP FRR
    - Teoretic poate accepta orice scenariu de protecție
    - Utilizează un algoritm de complexitate moderată
    - Selectează o cale de rezervă pe o rută convergentă și nu are o stare intermediară în comparație cu celelalte tehnici FRR

Protecția traficului TI-LFA implică protejarea unui link sau a unui nod, aceasta din urmă fiind prioritară. [13]

Pentru exemplificarea ingineriei traficului într-o rețea Segment-Routing MPLS am considerat topologia de rețea din figura de mai jos. Înainte de a configura TI-LFA FRR au fost necesară îndeplinirea unor etape esențiale: alocarea ip-urilor pe interfețe conform topologiei și alegerea interfeței de Loopback 0 cu mască IP de 32 de biți (ex: 1.1.1.1/32 pentru xr1), configurarea unui protocol de rutare de tipul link- state, am ales protocolul IS-IS (Intermediate Systems to Intermediate Systems), activarea Segment- Routing sub procesul IS-IS și activarea MPLS Traffic-Engineering în modul de configurare globală.



*Figură 4. 35. Topologia de rețea utilizată pentru exemplificare Segment-Routing MPLS Traffic-Engineering*

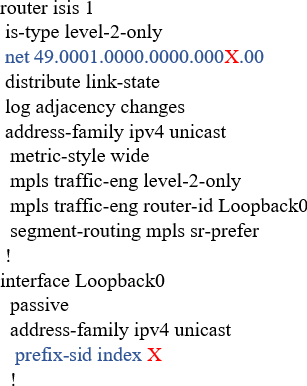
Protocolul IS-IS este un protocol de rutare IGP (Interior Gateway Protocol) care funcționează după principiul Flooding Information, astfel toată informația din tabela de rutare este trimisă în rețea pentru ca toate routerele din același domeniu de rutare să își populeze tabela de rutare. Acest protocol folosește același algoritm de rutare ca OSPF, și anume Shortest Path First.

IS-IS și OSPF sunt protocoale similare despre care se pot menționa și câteva comparații precum: IS-IS este executat într-un domeniu de rutare, pe când OSPF se execută într-un AS. Un domeniu de rutare poate fi împărțit în mai multe arii cu o arie centrală comună. Routerele din aria centrală se numesc routere de Level 2 sau sisteme intermediare de Level 2, iar celelalte routere din afara zonei centrale se numesc routere de Level 1 sau sisteme intermediare de Level 1.

Am decis ca toate routerele din topologie să aparțină nivelului 2 IS-IS și pentru fiecare dintre acestea am configurat sub interfața de loopback un index (SID-Segment Identifier index, 1 pentru xr1, 2 pentru xr2, etc.) pentru identificatorul de segment asociat cu prefixul IP. Intervalul SRGB (Segment Routing Global Block) este cel implicit: 16000-23999. Acesta este un interval de valori ale etichetelor rezervate pentru segmnet-routing, având o semnificație globală în întregul domeniu și fiind disponibil

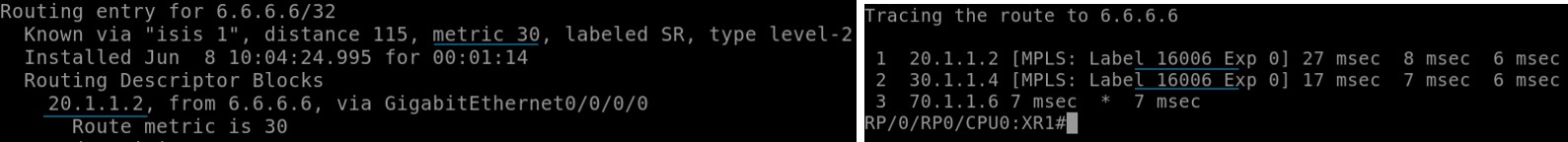
atunci când SR este activat pe un dispozitiv. Prefix-SID pentru fiecare nod a fost creat luând în considerare limita inferioară a SRGB + indexul stabilit. Prin urmare, pentru xr1 prefix-SID este 16001, pentru xr2 16002, etc. Metrica pentru fiecare link are o valoare implicită 10, costul total până la destinație fiind reprezentat de suma valorilor de pe toate interfețele de ieșire de-a lungul căii.

În ceea ce privește configurarea protocolului IS-IS, comenzile date pe echipamente sunt similare, diferă doar două comenzi, cele evidențiate în figura 4.36. Aceste comenzi pot fi vizualizate și în [Anexa 2](#_bookmark118) de la finalul lucrării. Pentru fiecare dispozitiv, X-ul de culoare roșie a fost înlocuit cu cifra corespunzătoare echipamentului: 1 pentru xr1, 2 pentru xr2, etc.



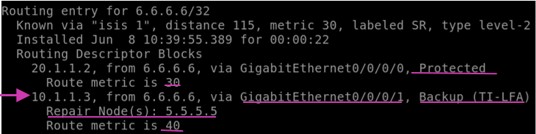
*Figură 4. 36. Configurație Procotol IS-IS*

După cum am menționat anterior, TI-LFA FRR este un mecanism de protecție a routerelor și a link-urilor într-o rețea, având rolul de a comuta rapid traficul pe o cale de rezervă, calculată și aleasă în mod dinamic cu ajutorul protocolului IGP. Prin utilizarea căii post-convergență planificată anterior se previne congestia tranzitorie și rutarea suboptima. Să presupunem că xr1 trebuie să trimită trafic către xr6. Calea stabilită de IGP fiind cea cu cel mai mic cost : xr1-xr2-xr4-xr6. În figura de mai jos putem observa valoarea 30 pentru metrică, fiecare link având costul implicit 10, și faptul că avem momentan doar o singură cale către 6.6.6.6/32, prin interfața Gi0/0/0/0. De asemenea, fiecare index pentru Prefix- SID configurat inițial a fost anunțat în întregul domeniu, astfel fiecare router a calculat aceeași valoare pentru eticheta corespunzătoare routerului xr6, 16006.

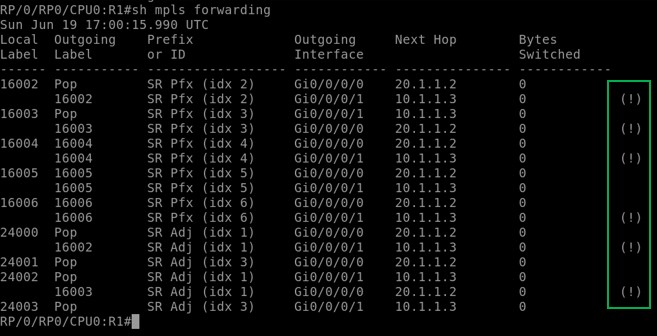


*Figură 4. 37. Calea de transmitere a traficului de la xr1 către xr6*

După activarea TI-LFA fast-reroute sub procesul IS-IS, pe interfața Gi0/0/0/0 a routerului xr1, vom avea o cale de rezervă stabilită de algorimtul TI-LFA (figura 4.39) care va asigura comutarea rapidă a pachetelor către xr6 în cazul unei defecțiuni pe link-ul Gi0/0/0/0. Metrica pentru calea de rezervă va avea valoarea 40, routerul xr5 va fi Repair Node sau P Node (Point of repair) , iar Gi0/0/0/0 va fi acum un link protejat. În figura 4.39 putem observa că în tabela de redirecționare mpls, calea de backup va fi evidențiată cu simbolul din cadranul verde.

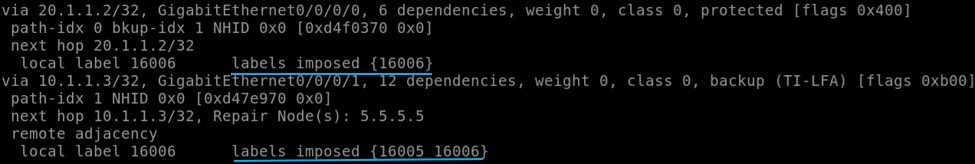


*Figură 4. 38. Apariția unei căi de backup după activarea TI-LFA*



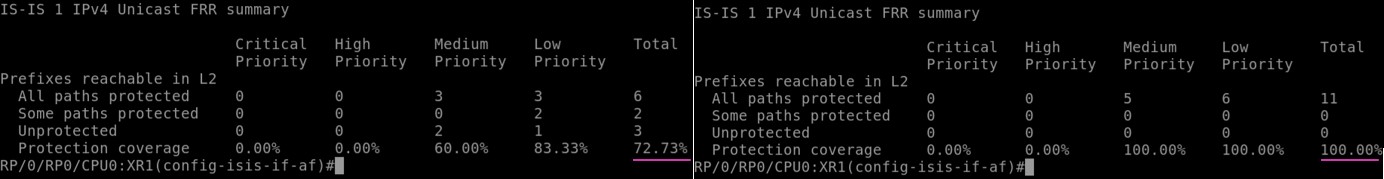
*Figură 4. 39. Apariția căilor de backup în tabela de redirecționare mpls după activarea TI-LFA*

Calea de rezervă conține o etichetă suplimentară 16005, o informație suplimentară adăugată astfel încât calea post-convergență să fie funcțională.



*Figură 4. 40. Vizualizarea etichetelor impuse*

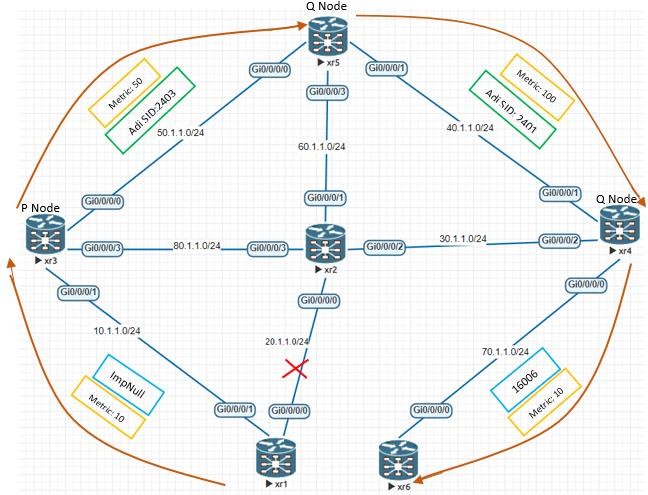
Configurând TI-LFA doar pe interfață Gi0/0/0/0 a routerului xr1 procentul de acoperire este de 72,73%. Prin urmare, pentru a obține o protecție de 100%, am configurat TI-LFA FRR și pe a doua interfață Gi0/0/0/1. Cu ajutorul comenzii „show isis fast-reroute summary” am vizualizat acoperirea de protecție pe care TI-LFA o oferă asupra topologiei:



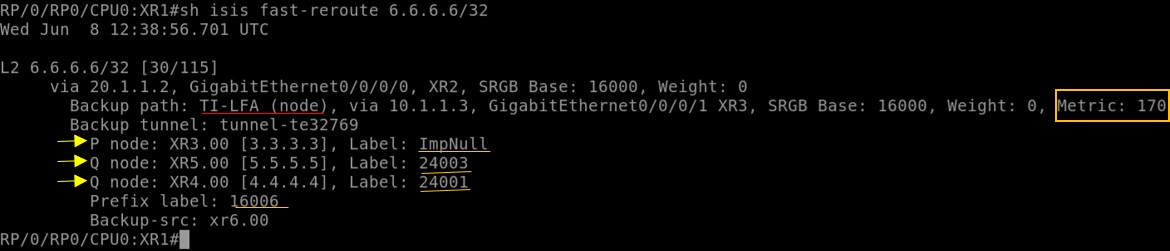
*Figură 4. 41. Protection coverage*

De asemenea, vom lua în considerare un scenariu în care link-urile nu au aceeași valoarea a costului. După cum am menționat anterior, TI-LFA oferă atât link-protection, cât și node-protection. Vom

pune în evidență mecanismul de redirecționare în cazul unei defecțiuni pe routerul xr2, cu node- protection, nodul vecin fiind exclus din calcularea căii de rezervă post-convergență. Am stabilit pentru link-ul Gi0/0/0/0 al routerului xr3 valoarea costului 50, iar pentru Gi0/0/0/1 al nodului xr5 valoarea 100. Crescând metrica IGP vom determina apariția a două segmente suplimentare utilizate pentru trimiterea traficului de la xr1 către xr6 pe calea de rezervă.



*Figură 4. 42. Ilustrarea unei căi post-convergență în cazul unor costuri diferite pe link-uri*

Routerul din spațiul P nu va mai fi xr5 ca în cazul anterior când metrica link-urilor era aceeași. În această situație, xr3 va fi nodul P (repair node) deoarece în mod implicit routerul nu va încerca să utilizeze link-ul cu metrica 50. Acesta va trimite traficul înapoi și va crea o buclă. De aceea, TI-LFA folosește un prefix de adiacență cu valoarea 2403 astfel încât xr3 să trimită traficul pe interfață Gi0/0/0/0 și unul cu valoarea 2401 astfel încât xr5 să trimită pachetul pe link-ul cu metrica 100, Gi0/0/0/1. Calea post- convergență este indicată în figura de mai sus. Pentru a trimite traficul de la un nod P la un nod Q este necesar Adjacency SID, iar pentru a trimite traficul către un nod P este nevoie de Prefix SID. Xr1 va trimite eticheta ImpNull deoarece este direct conectat cu xr3. Costul total al căii de rezervă calculate este acum 170.

*Figură 4. 43. Informații privind calea de backup TI-LFA în cazul unor costuri diferite pe link-uri*

Pentru a preveni microbuclele ce pot fi create în rețea, având ca rezultat pierderi de pachete, fluctuații sau sosirea pachetelor în ordine diferită față de cea în care au fost trimise, am configurat segmnet-routing microloops avoidance pe xr1. Microbuclele (microloops) sunt bucle scurte de pachete care apar în rețea în urma unor modificări de topologie care pot surveni în urma unor evenimente de modificare a conexiunii, a link-urilor sau a metricii. Microbuclele sunt cauzate de convergența non- simultană a diferitelor noduri din rețea. Dacă routerele recalculează calea și trimit trafic către un router vecin care nu a recalculat calea încă, traficul poate fi în buclă între aceste două dispozitive. Caracteristica Segmnet-Routing Microloop Avoidance detectează dacă microbuclele sunt posibile în urma unei modificări de topologie. Dacă un nod anticipează că ar putea apărea o microbucla în noua topologie, acesta creează o cale numită loop-free SR-TE către destinație folosind o listă de segmente. [15]

### Segment-routing MPLS-Traffic Engineering: Policy

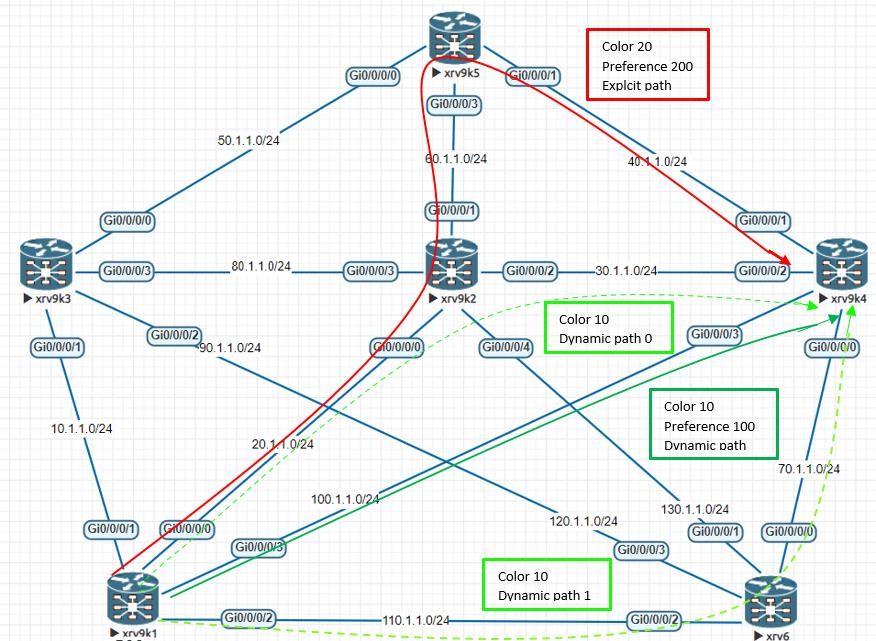
Segment Routing Traffic Engineering Policy este o tehnologie de tunel dezvoltată pe baza SR. O politică SR-MPLS TE este un set de căi candidate (candidate paths) reprezentate prin una sau mai multe liste de segmente (SID lists). Fiecare listă de segmente identifică o cale de la sursă la destinație, indicând dispozitivului să redirecționeze traficul prin calea stabilită.

SR-MPLS TE Policy este alcătuită din 3 elemente esențiale: head-end, color și end-point. Head- end este routerul pe care se configurează politica, iar informațiile privind culoarea și destinația sunt adăugate politicii respective prin configurare. O politică SR poate conține mai multe căi candidate cu atribute specifice: Preference și Binding SID. Calea candidată validă este cea cu cea mai mare valoare a atributului „preference”, iar calea cu cea de-a doua cea mai mare valoare a atributului funcționează ca o cale de backup. Tripletul (headend, color, endpoint) trebuie să fie unic. [16]

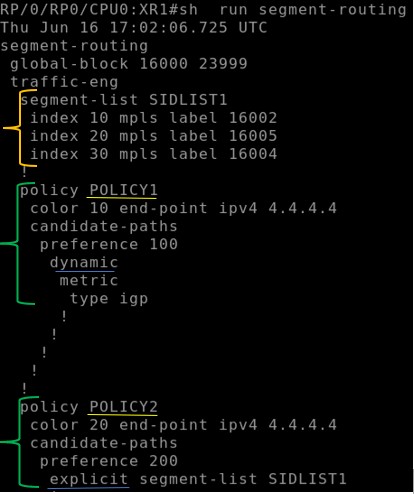
O cale candidată poate fi configurată dinamic sau explicit. În cazul unei căi dinamice, routerul head-end calculează o cale sub forma unei liste de segmente și de fiecare data când apare o schimbare de topologie, calea este recalculată. Configurarea unei căi în mod dinamic necesită exprimarea unui obiectiv de optimizare (optimzation objective) și a unui set de constrângeri (set of constraints). Obiectivul de optimizare se referă la anumite condiții de selectare a link-urilor care formează calea, cele cu întârziere scăzută de exemplu, sau a unui număr limitat de hopuri (hopcount). De asemenea, constrângerile impuse pot fi stabilite în ceea ce privește lățimea de bandă disponibilă sau existența unei căi de rezervă. Configurarea unei căi explicite presupune specificarea unei liste de segmente corespunzătoare căii. [17]

O condiție esențială pentru buna funcționalitate a SR-MPLS TE este ca toată informația IGP să fie distribuită în SR-TE Database și de aceea este necesar ca toate dispozitivele din rețea să aibă configurat sub procesul IS-IS distribuirea informației link-state.

În figura de mai jos sunt ilustrate două SR-TE Policy, color 10 și color 20 cu priorități diferite configurate în mod dinamic și explicit. Pentru configurarea unei SR-TE Policy în mod dinamic având ca head-end xr1 și end-point xr4 am specificat ca și constrângere metrica IGP, astfel principala cale candidată este cea de-a lungul link-ului ce conectează direct cele două dispozitive întrucât valoarea metricii pe toate link-urile este cea implicită IS-IS, și anume 10. Odată cu schimbarea valorii pentru metrica interfeței Gi0/0/0/3 a lui xr1, calea este recalculată (dynamic path 0 și 1 din figura) pentru a obține un cost minim, 20 în această situație.

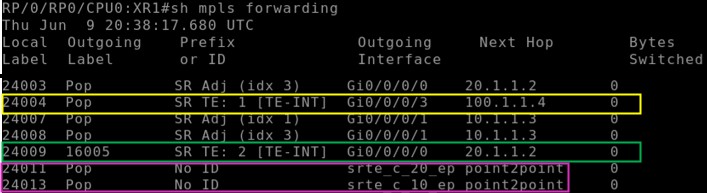


*Figură 4. 44. Ilustrarea unor SR-Policy configurate explicit și dinamic*



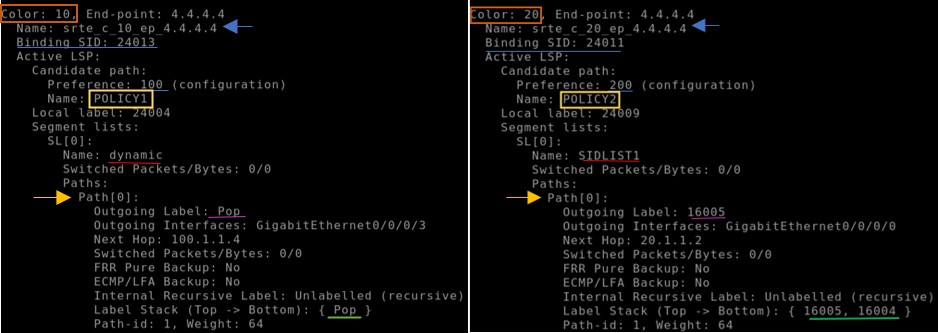
*Figură 4. 45. Configurația SR-Policy*

În figura 4.45 este ilustrată configurația de pe routerul xr1 a politicilor Policy1 și Policy2 descrise anterior. Aceasta poate fi vizualizată și în [Anexa 3](#_bookmark119) de la finalul lucrării. Odată ce SR-Policy devin active și utilizabile, vom observa apariția a două interfețe suplimentare: „srte\_c\_10\_ep\_4.4.4.4” corespunzătoare pentru Policy1 și respectiv „srte\_c\_20\_ep\_4.4.4.4” pentru Policy 2 (figura 4.46). De asemenea, urmărind figura 4.47 putem observa că în tabela de redirecționare mpls vor apărea două etichete locale suplimentare, 24011 și 24013, care sunt corespunzătoare celor două interfețe menționate anterior.



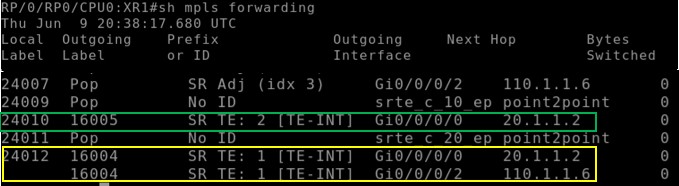
*Figură 4. 46. Interfețele lui xr1 Figură 4. 47. MPLS Forwarding Table*

Pentru a vizualiza caracteristicile celor două politici SR configurate cu aceeași destinație, și anume adresa de loopback a lui xr4 (4.4.4.4/32), am utilizat comanda „show segment-routing traffic- engineering forwarding policy detail”. Figura 4.48 ilustrează output-ul rezultat: putem identifica fiecare politică SR pe de-o parte cu ajutorul atributului „color” care este unic, iar pe de altă parte prin Binding SID: 24013 corespunzător Policy1, respectiv 24011 pentru Policy2. De asemenea, putem observa că Policy 1 este configurată dinamic, iar calea aleasă este cea mai scurtă, utilizând link-ul ce conectează direct xr1 cu xr4, astfel în câmpul „Outgoing Label” nu avem nicio etichetă „pop”, în timp ce pentru Policy 2 configurată explicit, utilizând lista de segmente „SIDLIST1”, este necesară etichetarea pachetului cu Prefix SID 16005, corespunzător lui xr5.

s

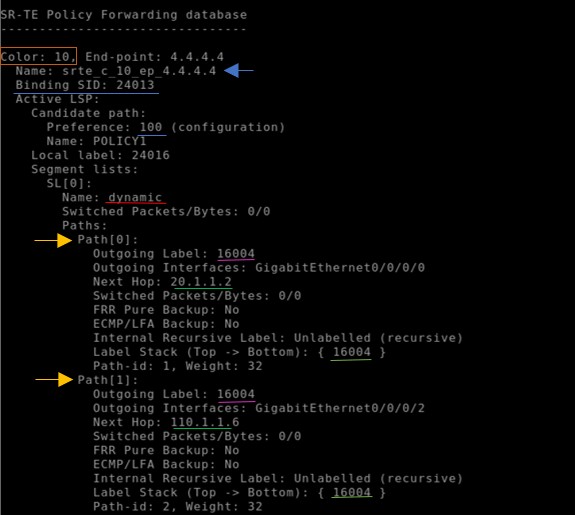
*Figură 4. 48. Caracteristicile Politicilor SR configurate*

Pentru a ilustra facilitățile unei politici SR configurate dinamic, vom crește costul link-ului Gi0/0/0/3 a lui xr1. O politică SR configurată dinamic se actualizează atunci când apare o modificare în rețea, asftel xr1 va recalcula cea mai optimă cale pentru a trimite trafic către xr4. Astfel, în tabela de redirecționare mpls, figura 4.49 , vom avea corespunzător pentru 16004 două căi cu costuri egale, utilizând interfețele Gi0/0/0/0 și Gi0/0/0/2.



*Figură 4. 49. MPLS Forwarding Table după modificarea costului interfeței Gi0/0/0/3*

De asemenea, utilizând comanda „show segment-routing traffic-engineering forwarding policy detail” putem observa caracteristicile pentru Policy 1 care includ acum două posibile căi, Path[0] și Path[1] pentru destinația 4.4.4.4/32:



*Figură 4. 50. Caracteristicile SR Policy 1 după modificarea costului interfeței Gi0/0/0/3*

### Segment-routing MPLS-Traffic Engineering: Path Computation Element

Path Computation Element Protocol (PCEP) descrie un set de proceduri prin care un client PCC (Path Computation Client) poate raporta și delega controlul căilor LSP (Label Switched Paths) provenite de la PCC la PCE corespunzător. PCE poate solicita PCC să facă update și să modifice parametrii LSP- urilor pe care le controlează. SR-PCE este acceptat pe planul de date MPLS și planul de control IPV4. [18]

Există câteva elemente cheie atunci când ne referim la SR-PCE:

* + - PCC = Path Computation Client, orice client care solicită calculul căii care urmează să fie efectuat de PCE
    - PCE= Path Computation Element, o entitate (componentă, aplicație sau nod de rețea) care este capabil să calculeze o cale sau o rută de rețea pe baza unor informații și a aplicării constrângerilor de calcul
    - PCEP Peer = element implicat într-o sesiune PCEP (poate fi PCC sau PCE)
    - TED = Traffic Engineering Database, conține topologia și informații despre resursele domeniului, poate fi populată cu ajutorul extensiilor IGP
    - TE LSP = Traffic Engineering Label Switched Path

Atunci când PCC și PCE nu sunt asociate, este necesar un protocol de comunicare între acestea. PCEP (Path Computation Element Protocol) este un protocol conceput special pentru comunicațiile între un PCC și un PCE sau între două PCE-uri. Un PCC poate folosi PCEP pentru a trimite o solicitare de calcul a căii pentru unul sau mai multe TE LSP către un PCE, iar PCE poate răspunde cu una sau mai

multe căi calculate în funcție de constrângerile impuse. PCEP utilizează ca și protocol de transport TCP, îndeplinind astfel cerințele pentru mesagerie fiabila și controlul fluxului.

Sunt definite mai multe mesaje PCEP:

* + - Mesajele Open și Keepalive – folosite pentru a iniția și menține o sesiune PCEP
    - PCReq – mesaj PCEP trimis de un PCC către un PCE pentru a solicita calculul căii
    - PCRep – mesaj PCEP trimis de un PCE către un PCC ca răspuns la o solicitare de calcul al căii; poate conține fie un set de căi luând în considerare respectarea constrângerilor, fie un răspuns negativ în caz contrar și de asemenea motivul pentru care nu s-a găsit nicio cale
    - PCNtf – mesaj de notificare PCEP trimis de un PCC către un PCE sau invers pentru a notifica un anumit eveniment
    - PCErr – mesaj PCEP cauzat de apariția unei erori în ceea ce privește o condiție a protocolului
    - Close message – mesaj de încheiere a sesiunii PCEP

Setul de PCE-uri disponibile poate fi configurat static pe un PCC sau descoperit dinamic. Un PCC poate avea sesiuni PCEP cu mai multe PCE și invers. [19]

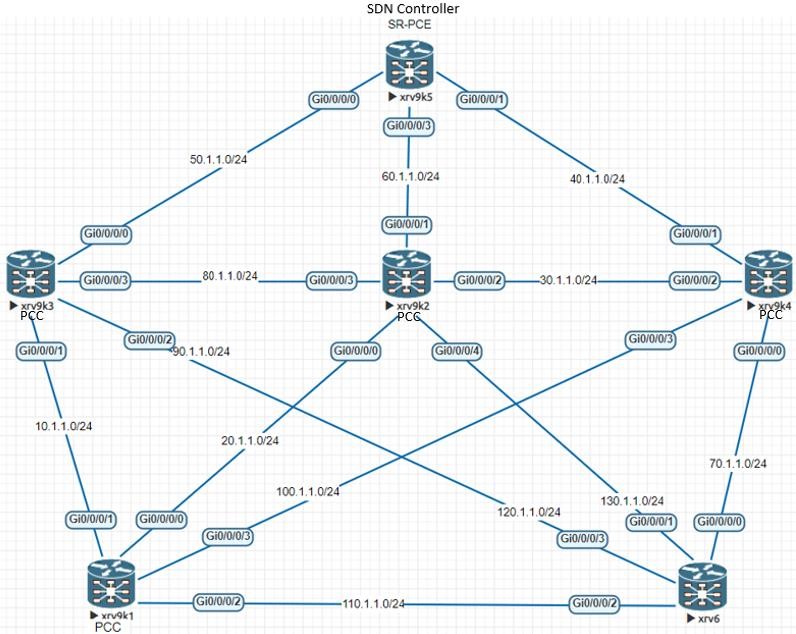
SR-PCE învață informații despre topologie prin intermediul IGP (OSPF sau IS-IS) sau prin BGP Link-State (BGP-LS). SR-PCE este capabil să calculeze căi folosind următoarele metode:

* + - Metrica TE - SR-PCE utilizează valoarea TE în calculele căilor sale pentru a optimiza valoarea TE cumulativă
    - Metrica IGP - SR-PCE utilizează valoarea IGP în calculele sale pentru a optimiza accesibilitatea
    - LSP Disjointness - SR-PCE utilizează algoritmii de calcul ai căii pentru a calcula o pereche de LSP-uri disjuncte. Căile disjuncte pot proveni din același head-end router sau din unul diferit [20]

XR Traffic Controller este un dispozitiv IOS XR care se comportă ca un PCE. XTC colectează informații despre topologia rețelei prin mai multe protocoale de rutare: OSPF, IS-IS sau BGP Link-State (BGP-LS). Baza de date topologică este proiectată pe mai multe domenii, astfel se depășesc limitările soluțiilor de inginerie a traficului tradiționale. Vizualizarea datelor topologice end-to-end oferă posibilitatea de calcul al căilor într-un mod mult mai simplu. XTC folosește PCEP pentru a trimite actualizări în cadrul rețelei. [18]

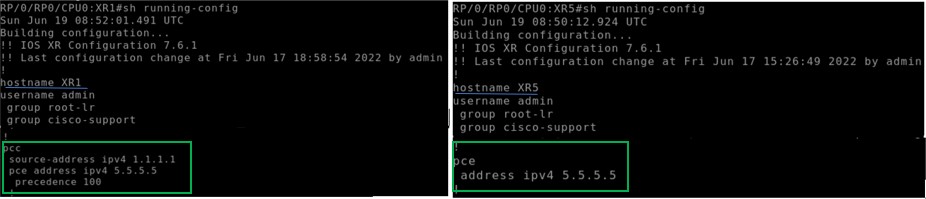
Pentru a crește rezistența rețelei, XTC poate calcula două căi disjuncte, două căi care nu traversează același link, astfel se pot îndeplini cerințele avansate ale clienților. XTC depășește limitările ingineriei traficului în rețelele mari alcătuite din mai multe domenii în care calcularea căilor este complexă din cauza vizibilității limitate între domeniile învecinate, cum ar fi un alt sistem autonom sau zonele IGP învecinate. În cadrul SDN (Software-Defined Networking) și al automatizării rețelelor, Path Computation Element (PCE) este o piesă critică pentru redirecționarea traficului IP/MPLS. [18]

Pentru a pune în evidență avantajele utilizării PCE într-o rețea am decis configurarea dispozitivului xr5 ca Path Computation Element și a routerelor xr1, xr3, xr3, xr4 ca Path Computation Client.



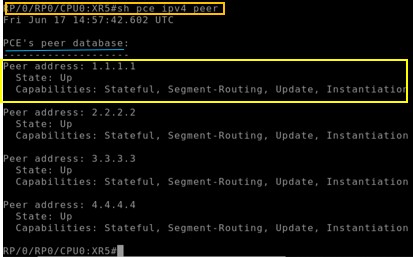
*Figură 4. 51. Topologia de rețea controlată de SR-PCE*

Configurația utilizată pentru dispozitive este ilustrată în figura 4.52:



*Figură 4. 52. Configurația pentru dispozitivele PCE și PCC*

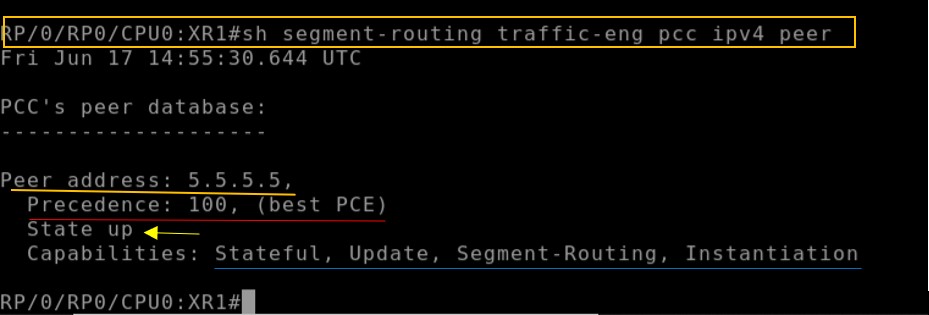
Pentru a verifica că sesiunea PCEP este stabilită între PCE și PCC vom utiliza următoarele comenzi: „show pce ipv4 peer” pentru „PCE’s peer database” și „show segment-routing traffic-eng pcc ipv4 peer detail” pentru „PCC’s peer database”. Din figura de mai jos putem observa că starea este „Up” pentru legăturile cu toți clienții PCC, iar capabilitățile sunt „Stateful, Segment-Routing, Update”.



*Figură 4. 53. PCE’s peer database*

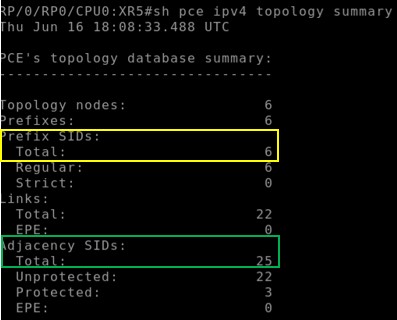
Pentru a vizualiza „PCC’s peer database” , vom da comanda „show segment-routing traffic-eng pcc ipv4 peer brief” pe unul dintre routerele configurate ca PCC. În figura 4.54 putem observa că adresa din câmpul „Peer address” este exact adresa de loopback a dispozitivului configurat ca PCE,

5.5.5.5 a lui xr5. De asemenea, prioritatea are valoarea 100, urmată de informația „best PCE” . În cazul în care într-o topologie se configurează două dispozitive PCE, acestea vor avea priorități diferite, iar cel cu prioritatea cea mai mare va fi considerat „best PCE”. Starea legăturii este „up” și capabilitățile sunt

„Stateful, Segment-Routing, Update”.

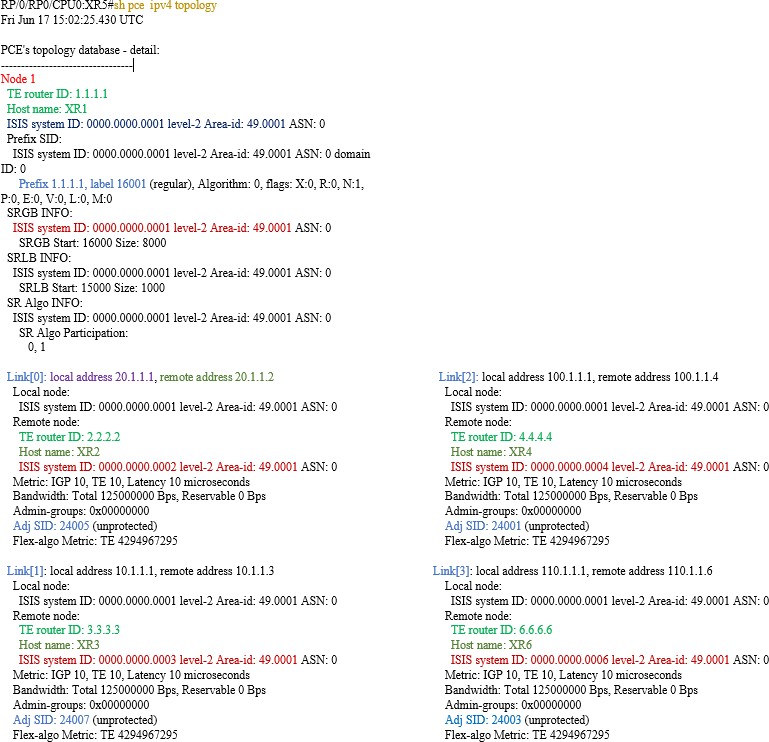
*Figură 4. 54. PCC’s peer database*

O caracteristică importantă a unui dispozitiv configurat ca PCE într-o rețea este că acesta menține o bază de date topologică a informațiilor provenite prin IGP. Pentru a vizualiza informațiile pe care PCE le are despre topologie am utilizat comanda „ show pce ipv4 topology summary”. Putem observa în figura

4.55 faptul că xr5 cunoaște numărul total de SID-uri de adiacență și prefix utilizate în întreagă rețea pentru redirecționarea traficului.

*Figură 4. 55. Informații topologice deținute de SR-PCE*

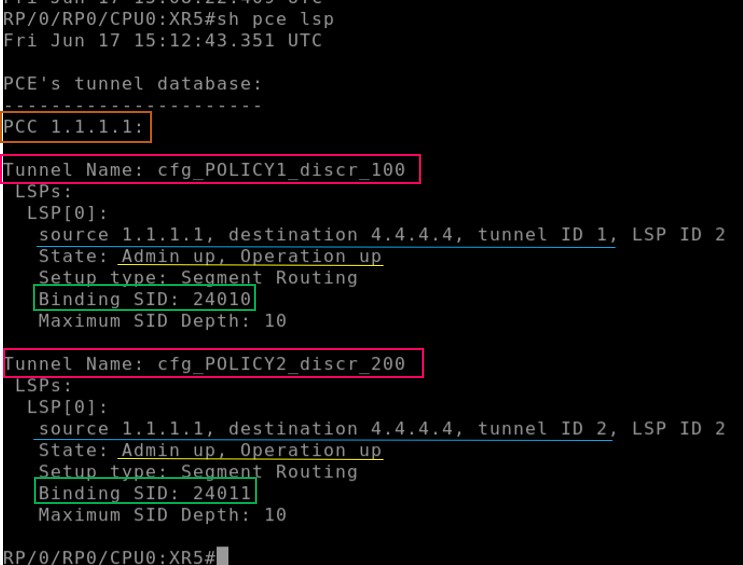
De asemenea, pentru a vizualiza informații mai amănunțite în legătură cu topologia putem utiliza comanda „show pce ipv4 topology”. Xr5 deține informații despre toate dispozitivele din rețeaua simulată și utilizează această bază de date topologică pentru a calcula căile. O parte din aceste informații, specifice pentru xr1, se pot vizualiza în figura următoare. Xr5 deține informații despre nodurile rețelei, dar și despre link-urile conectate. Putem vizualiza din figura 4.51 că xr1 are conectate 4 link-uri către celelalte dispozitive, iar în figura 4.57 sunt ilustrate informațiile despre aceste link-uri pe care xr5 le deține în baza de date topologică.



*Figură 4. 56. Informații din baza de date topologică a routerului SR-PCE*

Fiecare link este identificat printr-o adresă locală, adresa configurată pe interfața specifică, și de asemenea și o adresă remote, care este adresa configurată pe interfața routerului vecin. De asemenea, pentru fiecare link există un câmp „TE router ID” care conține adresa de loopback a routerului vecin lui xr1. În plus, fiecare link se identifică și printr-un prefix de adiacentă „Adj SID: 24005” pentru „Link[0]” de exemplu. Xr5 deține informații și despre domeniul IS-IS pentru nodul local, xr1 în cazul acesta „ISIS system ID: 0000.0000.0001 level-2 Area-id: 49.0001”, cât și despre nodurile remote „ISIS system ID: 0000.0000.0002 level-2 Area-id: 49.0001”, pentru link-ul către xr2. În plus, „PCE’s topology database” oferă informații și despre costurile și întârzierea fiecărui link

„Metric: IGP 10, TE 10, Latency 10 microseconds”.

În figura 4.57 putem observa că xr5 deține informații și despre politicile configurate anterior, care au ca head-end dispozitivul xr1 și ca end-point xr4. De asemenea, output-ul ne oferă informații despre „tunnel ID” și despre „Binding SID” care sunt unice pentru fiecare LSP configurat.

*Figură 4. 57. Informații din baza de date topologică a routerului SR-PCE referitoare la politicile SR configurate*

## Concluzii

Tehnologia a cunoscut o creștere exponențială într-o perioadă scurtă de timp, ceea ce a determinat apariția multor dispozitive și implicit o creștere a numărului de utilizatori. Complexitatea rețelelor de astăzi au ca și consecință nevoi și cerințe din ce în ce mai mari. Comunicarea online a devenit semnificativ mai importantă în ziua de azi și pentru a îndeplini cerințele superioare impuse, tehnologiile de transport a informației au fost dezvoltate pentru a atinge noi niveluri de performanță. Utilizând MPLS se pot extinde posibilitățile a ceea ce putem transporta prin rețea , acesta oferind o metodă simplă de comutare a pachetelor, nefiind nevoie ca routerele care transportă traficul să cunoască sarcina utilă, fiind necesară doar informația etichetei pentru a redirecționa pachetele. După cum am observat și în subcapitolul 4.2, utilzând MPLS într-o rețea de service provider doar routerele de margine (edge routers) trebuie să ruleze BGP, iar cele din nucleul rețelei transmit doar pachetele etichetate, sunt scutite de sarcina rulării BGP și astfel se reduce memoria utilizată. Totodată, MPLS oferă un avantaj și rețelelor private virtuale, unde site-urile ce aparțin diverselor companii sunt interconectate și complet separate de alte VPN-uri. MPLS asigură confidențialitatea utilizând conceptul de VRF, iar informațiile de rutare de la diferiți clienți sunt păstrate separat, pachetele fiind redirecționate doar pe baza etichetei. În plus, MPLS oferă soluții pentru inginerie de trafic, utilizând optim infrastructura de rețea, asigurând o fluidizare a traficului și evitând congestiile.

În subcapitolul 4.4. am analizat beneficiile pe care le oferă Segment-Routing implementat peste un plan de date MPLS. Segment-Routing MPLS oferă o implementare simplă, elimină semnalizarea suplimentară oferită de protocolul RSVP-TE și utilizarea protocoalelor de distribuție, LDP, care erau absolut necesare pentru implementarea MPLS tradițională. Segment-Routing creează tuneluri MPLS doar utilizând protocoalele IGP. Furnizează Fast-Reroute automat pentru orice topologie cu ajutorul mecanismului TI-LFA (Topology-Independent Loop-Free Alternate). TI-LFA FRR bazat pe SR oferă protecția nodurilor și a link-urilor în orice topologie, compensând astfel cu dezavantajele tehnologiilor tradiționale de protecție a tunelurilor. În cazul în care un link sau un nod eșuează, TI-LFA FRR comută rapid traficul pe o cale de rezervă, minimizând astfel pierderile de pachete care rezultă în timp ce routerele recalculează căile după o modificare de topologie. Soluționarea este rapidă și realizată prin utilizarea căilor de rezervă calculate anterior. TI-LFA creează un tunel de rezervă SR numit „repair path” sau „backup path” care urmează să fie utilizat atunci când calea primară nu mai este disponibilă. Pierderile de pachete sunt reduse cu ajutorul tehnologiei TI- LFA Microloop Avoidance, astfel modificările spontane ale topologiei nu vor mai afecta transmisia traficului de la sursă la destinație.

De asemenea, Prefix-SID au o semnificație globală în rețea, fiecare index asociat cu Prefix- SID configurat pe echipament este anunțat în întregul domeniu prin protocolul IGP configurat și astfel fiecare router calculează aceeași valoare pentru eticheta corespunzătoare fiecărui vecin, operarea se face mult mai simplu în comparație cu MPLS unde eticheta era schimbată hop-by-hop în rețea.

Am evidențiat în cadrul capitolului 4.5 utilizarea tehnologiei de tunel Segment-Routing Traffic Engineering Policy dezvoltată pe baza SR, prin care se elimină complexitatea tunelurilor de configurat, astfel Segment-Routing poate opera la o scara mai mare, fiind adaptabil la cerințele și nevoile de schimbare ale utilizatorilor. SR-Policy oferă un set de căi candidate (candidate paths) reprezentate prin una sau mai multe liste de segmente prin care se identifică o cale de la sursă la destinație. Configurate în mod dinamic, SR-Policy oferă posibilitatea specificării unor constrângeri și a unui obiectiv de optimizare. Atunci când o cale candidată devine indisponibilă, traficul se redirecționează automat pe calea ai cărei parametrii respectă condițiile impuse.

În plus, segment-routing este o arhitectură concepută pentru a permite implementarea SDN (Software Define Network). După cum am observat în capitolul 4.6, implementând Segment-Routing Path Computation Element putem obține pe un dispozitiv o imagine completă asupra topologiei de

rețea și a fluxurilor de trafic. Acest dispozitiv este numit și XR Traffic Controller și deține o bază de date topologică creată prin colectarea informațiilor cu ajutorul protocolului link-state, IS-IS. XTC utilizează această bază de date pentru a calcula diferite căi optime de la sursă la destinație și stabilește o sesiune PCEP cu clienții PCC.

În concluzie, Segment-Routing implementat într-o arhitectură MPLS aduce beneficii din multe puncte de vedere, este o soluție optimă pentru a facilita ingineria traficului în rețelele cu cerințe și nevoi superioare ale furnizorilor de servicii. Segment-Routing include tehnologii superioare care reduc complexitatea semnificativ în rețea mulțumită beneficiilor suplimentare aduse din punct de vedere al direcționării automate a traficului, oferă o implementare simplă și eficientă, este un mecanism de orchestrare a rețelelor ce garantează agilitate crescută și scalabilitate îmbunătățită.

## Bibliografie

|  |  |
| --- | --- |
| [1] | „The Importance of Online Communication in Today’s World,”. Available: https://eztalks.com/unified-  communications/the-main-benefits-of-online-communication.html. [Accesat 12 02 2022]. |
| [2] | H. R. Doug Marschke, „JUNOS Enterprise Routing”, O'Reilly Media, ISBN: 9780596514426, March 2008. |
| [3] | L. D. Ghein, „MPLS Fundamentals”, 800 East 96th Street, Indianapolis, IN 46240 USA: Cisco Press, ISBN: 1-58705-197-4, 2007. |
| [4] | HUAWEI, „MPLS Control Plane and Forwarding Plane Interaction”. Available: https://support.huawei.com/enterprise/en/doc/EDOC1000178173/e555f541/mpls-forwarding. [Accesat 25 01 2022]. |
| [5] | D. Galațchi, „Curs 3 - ICMP+ARP+RARP+DHCP,” în *Arhitecturi de rețea și internet*, UPB. |
| [6] | HUAWEI, „Configuring the Tunnel Priority”. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100116685/7a1274ff/configuring-the-tunnel- priority. [Accesat 24 06 2022]. |
| [7] | ARISTA , „MPLS Segment-Routing”. Available: https://[www.arista.com/en/solutions/mpls-segment-](http://www.arista.com/en/solutions/mpls-segment-) routing. [Accesat 12 03 2022]. |
| [8] | CISCO, „ Segment Routing Configuration Guide”, Cisco IOS XE Release 3S, 170 West Tasman Drive, USA, Americas Headquarters, Cisco Systems. |
| [9] | HUAWEI, „Understanding Segment Routing MPLS” Available: https://support.huawei.com/enterprise/es/doc/EDOC1000173015/3ee10304/understanding- segment-routing-mpls. [Accesat 25 03 2022]. |
| [10] | R. Mota, „Segment Routing” ACG Research Paper, 2018. |
| [11] | K. M. Clarence Filsfils, „SR Traffic-Engineering” CISCO. Available: https://www.segment- routing.net/tutorials/2017-03-06-segment-routing-traffic-engineering-srte/. [Accesat 28 03 2022]. |
| [12] | HUAWEI, „SRv6”. Available: https://info.support.huawei.com/info-finder/encyclopedia/en/SRv6.html. [Accesat 12 04 2022]. |
| [13] | HUAWEI, „TI-LFA FRR” Available:  https://support.huawei.com/enterprise/en/doc/EDOC1100059465/2149ad9d/ti-lfa-frr. [Accesat 23 05  2022]. |
| [14] | CISCO, „Segment Routing Configuration Guide, Cisco IOS XE Gibraltar 16.12.x” . Available: https://[www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg\_routing/configuration/xe-16-12/segrt-xe-](http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg_routing/configuration/xe-16-12/segrt-xe-) 16-12-book.pdf. [Accesat 29 05 2022]. |
| [15] | CISCO, „Configure Segment Routing Microloop Avoidance”. Available: https://[www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/63x/b-segment-routing-cg-](http://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/63x/b-segment-routing-cg-) ncs5500-63x/b-segment-routing-cg-ncs5500-63x\_chapter\_01010.pdf. [Accesat 29 05 2022]. |

|  |  |
| --- | --- |
| [16] | HUAWEI „SR-MPLS TE Policy,”. Available:  https://support.huawei.com/enterprise/en/doc/EDOC1100125823/c38497ba/sr-mpls-te-policy. [Accesat 02 06 2022]. |
| [17] | CISCO, „Segment Routing Traffic Engineering,” [Interactiv]. Available: https://www.segment- routing.net/tutorials/2017-03-06-segment-routing-traffic-engineering-srte/. [Accesat 2022 06 14]. |
| [18] | „Configure Segment Routing Path Computation,” [Interactiv]. Available: https://[www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-6/segment-](http://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-6/segment-) routing/configuration/guide/b-segment-routing-cg-asr9000-66x/b-segment-routing-cg-asr9000- 66x\_chapter\_01001.pdf. [Accesat 07 06 2022]. |
| [19] | RFC5440, „Path Computation Element (PCE) Communication Protocol (PCEP)” March 2009. |
| [20] | CISCO, „Segment Routing Traffic Engineering (SR-TE)” .Available: https://www.segment- routing.net/tutorials/2017-03-06-segment-routing-traffic-engineering-srte/. [Accesat 14 06 2022]. |

*Anexa 1*

R1#sh running-config Building configuration...

Current configuration : 4208 bytes

!

! Last configuration change at 09:23:38 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R1

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 1000 1999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 1.1.1.1 255.255.255.255

!

interface Tunnel1

ip unnumbered Loopback0 tunnel mode mpls traffic-eng tunnel destination 7.7.7.7

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng priority 7 7

tunnel mpls traffic-eng bandwidth 512

tunnel mpls traffic-eng path-option 1 explicit name R1-R7

tunnel mpls traffic-eng fast-reroute node-protect no routing dynamic

!

interface GigabitEthernet0/0

ip address 10.1.1.1 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 5000

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 1.1.1.1 0.0.0.0 area 0

network 10.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

ip explicit-path name R1-R7 enable next-address 10.1.1.2

next-address 30.1.1.4

next-address 80.1.1.7

!

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R2#show running-config Building configuration...

Current configuration : 4843 bytes

!

! Last configuration change at 10:43:58 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R2

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 2000 2999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize timers frequency 180 mpls traffic-eng reoptimize events link-up

!

!

!

!

!

Redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 2.2.2.2 255.255.255.255

!

interface Tunnel2

ip unnumbered Loopback0 tunnel mode mpls traffic-eng tunnel destination 8.8.8.8

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng priority 0 0

tunnel mpls traffic-eng bandwidth 256

tunnel mpls traffic-eng path-option 1 explicit name Tunnel\_R2\_R8

no routing dynamic

!

interface Tunnel100

description "Link Protection Tunnel" ip unnumbered Loopback0

tunnel mode mpls traffic-eng tunnel destination 4.4.4.4

tunnel mpls traffic-eng path-option 1 explicit name Tunnel\_R2\_R4

no routing dynamic

!

interface Tunnel101

description "Node Protection Tunnel" ip unnumbered Loopback0

tunnel mode mpls traffic-eng tunnel destination 7.7.7.7

tunnel mpls traffic-eng path-option 1 explicit name Tunnel\_R2\_R6\_R5\_R7

no routing dynamic

!

interface GigabitEthernet0/0

ip address 10.1.1.2 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/1

ip address 20.1.1.2 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels

mpls traffic-eng backup-path Tunnel101 mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/2

ip address 30.1.1.2 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels

mpls traffic-eng backup-path Tunnel100 mpls traffic-eng backup-path Tunnel101 mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/3

ip address 60.1.1.2 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 2.2.2.2 0.0.0.0 area 0

network 10.1.1.0 0.0.0.255 area 0

network 20.1.1.0 0.0.0.255 area 0

network 30.1.1.0 0.0.0.255 area 0

network 60.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

ip explicit-path name Tunnel\_R2\_R4 enable next-address 20.1.1.3

next-address 40.1.1.4

!

ip explicit-path name Tunnel\_R2\_R6\_R5\_R7 enable next-address 60.1.1.6

next-address 70.1.1.5

!

ip explicit-path name Tunnel\_R2\_R8 enable next-address 20.1.1.3

next-address 40.1.1.4

next-address 80.1.1.7

next-address 90.1.1.8

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R3#sh running-config Building configuration...

Current configuration : 3310 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R3

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 3000 3999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize timers frequency 180 mpls traffic-eng reoptimize events link-up

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 3.3.3.3 255.255.255.255

!

interface GigabitEthernet0/0

ip address 40.1.1.3 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/1

ip address 20.1.1.3 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 3.3.3.3 0.0.0.0 area 0

network 20.1.1.0 0.0.0.255 area 0

network 40.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R4#sh run

Building configuration...

Current configuration : 4198 bytes

!

! Last configuration change at 09:18:18 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R4

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 4000 4999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize timers frequency 180 mpls traffic-eng reoptimize events link-up

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 4.4.4.4 255.255.255.255

!

interface Tunnel100

description "Link Protection Tunnel" ip unnumbered Loopback0

tunnel mode mpls traffic-eng tunnel destination 2.2.2.2

tunnel mpls traffic-eng path-option 1 explicit name Tunnel\_R4\_R2

no routing dynamic

!

interface GigabitEthernet0/0

ip address 40.1.1.4 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/1

ip address 50.1.1.4 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/2

ip address 30.1.1.4 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/3

ip address 80.1.1.4 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 1024

!

interface GigabitEthernet0/4

ip address 160.1.1.4 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/5 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 4.4.4.4 0.0.0.0 area 0

network 30.1.1.0 0.0.0.255 area 0

network 40.1.1.0 0.0.0.255 area 0

network 50.1.1.0 0.0.0.255 area 0

network 80.1.1.0 0.0.0.255 area 0

network 160.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

ip explicit-path name Tunnel\_R4\_R2 enable next-address 40.1.1.3

next-address 20.1.1.2

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R5#sh run

Building configuration...

Current configuration : 3392 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R5

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 5000 5999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize events link-up

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 5.5.5.5 255.255.255.255

!

interface GigabitEthernet0/0

ip address 70.1.1.5 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/1

ip address 50.1.1.5 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/2

ip address 140.1.1.5 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0

mpls traffic-eng area 0 network 5.5.5.5 0.0.0.0 area 0

network 50.1.1.0 0.0.0.255 area 0

network 70.1.1.0 0.0.0.255 area 0

network 140.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

end

R6#sh run

Building configuration...

Current configuration : 3328 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R6

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 6000 6999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 6.6.6.6 255.255.255.255

!

interface GigabitEthernet0/0

ip address 70.1.1.6 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45 mpls ip

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45 mpls ip

!

interface GigabitEthernet0/3

ip address 60.1.1.6 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 6.6.6.6 0.0.0.0 area 0

network 60.1.1.0 0.0.0.255 area 0

network 70.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R7#sh run

Building configuration...

Current configuration : 3748 bytes

!

! Last configuration change at 09:17:17 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R7

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 7000 7999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize timers frequency 180 mpls traffic-eng reoptimize events link-up

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 7.7.7.7 255.255.255.255

!

interface GigabitEthernet0/0

ip address 90.1.1.7 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 1024

!

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2

ip address 140.1.1.7 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 512

!

interface GigabitEthernet0/3

ip address 80.1.1.7 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

ip rsvp bandwidth 1024

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 7.7.7.7 0.0.0.0 area 0

network 80.1.1.0 0.0.0.255 area 0

network 90.1.1.0 0.0.0.255 area 0

network 140.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

end

PE1-R8#sh run Building configuration...

Current configuration : 5380 bytes

!

! Last configuration change at 09:16:30 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname PE1-R8

!

boot-start-marker boot-end-marker

!

!

vrf definition CLIENT-A rd 1:1

route-target export 100:100

route-target import 200:200

!

address-family ipv4 exit-address-family

!

vrf definition CLIENT-B rd 1:2

route-target export 300:300

route-target import 400:400

!

address-family ipv4 exit-address-family

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 8000 8999

mpls label protocol ldp mpls traffic-eng tunnels

mpls traffic-eng reoptimize timers frequency 180 mpls traffic-eng reoptimize events link-up

!

!

!

!

!

Redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 8.8.8.8 255.255.255.255

!

interface Tunnel10

ip unnumbered Loopback0 tunnel mode mpls traffic-eng tunnel destination 12.12.12.12

tunnel mpls traffic-eng autoroute announce

tunnel mpls traffic-eng path-option 1 explicit name TUNNEL10-PE1-PE2

tunnel mpls traffic-eng fast-reroute no routing dynamic

!

interface Tunnel20

ip unnumbered Loopback0 mpls ip

tunnel mode mpls traffic-eng

tunnel destination 9.9.9.9

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng path-option 1 dynamic no routing dynamic

!

interface GigabitEthernet0/0

ip address 90.1.1.8 255.255.255.0 duplex auto

speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/1 vrf forwarding CLIENT-B

ip address 190.1.1.8 255.255.255.0

duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2

ip address 100.1.1.8 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/3

ip address 170.1.1.8 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels

mpls traffic-eng backup-path Tunnel20 mpls ip

!

interface GigabitEthernet0/4

ip address 160.1.1.8 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/5 vrf forwarding CLIENT-A

ip address 180.1.1.8 255.255.255.0

duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 8.8.8.8 0.0.0.0 area 0

network 90.1.1.0 0.0.0.255 area 0

network 100.1.1.0 0.0.0.255 area 0

network 160.1.1.0 0.0.0.255 area 0

network 170.1.1.0 0.0.0.255 area 0

!

router ospf 2

!

router bgp 65000

bgp router-id 8.8.8.8

bgp log-neighbor-changes

neighbor 12.12.12.12 remote-as 65000 neighbor 12.12.12.12 update-source Loopback0 neighbor 12.12.12.12 send-community both

!

address-family vpnv4 neighbor 12.12.12.12 activate

neighbor 12.12.12.12 send-community both exit-address-family

!

address-family ipv4 vrf CLIENT-A redistribute connected

redistribute static exit-address-family

!

address-family ipv4 vrf CLIENT-B redistribute connected

neighbor 190.1.1.15 remote-as 500

neighbor 190.1.1.15 activate exit-address-family

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

ip route vrf CLIENT-A 16.16.16.16 255.255.255.255

180.1.1.16

!

ip explicit-path name TUNNEL10-PE1-PE2 enable next-address 170.1.1.10

next-address 130.1.1.11

next-address 120.1.1.12

!

!

!

mpls ldp router-id Loopback0

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R9#show running-config Building configuration...

Current configuration : 3749 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R9

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 9000 9999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 9.9.9.9 255.255.255.255

!

interface Tunnel21

ip unnumbered Loopback0 mpls ip

tunnel mode mpls traffic-eng tunnel destination 8.8.8.8

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng path-option 1 dynamic no routing dynamic

!

interface Tunnel23

ip unnumbered Loopback0 mpls ip

tunnel mode mpls traffic-eng tunnel destination 12.12.12.12

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng path-option 1 dynamic no routing dynamic

!

interface GigabitEthernet0/0

ip address 110.1.1.9 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2

ip address 100.1.1.9 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 9.9.9.9 0.0.0.0 area 0

network 100.1.1.0 0.0.0.255 area 0

network 110.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

mpls ldp router-id Loopback0

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

end

R10# show running-config Building configuration...

Current configuration : 3372 bytes

!

! Last configuration change at 08:54:41 UTC Fri Jun 17 2022

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R10

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 10000 10999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 10.10.10.10 255.255.255.255

!

interface GigabitEthernet0/0

ip address 130.1.1.10 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto

speed auto media-type rj45

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3

ip address 170.1.1.10 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 10.10.10.10 0.0.0.0 area 0

network 130.1.1.0 0.0.0.255 area 0

network 170.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

mpls ldp router-id Loopback0

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R11#show running-config Building configuration...

Current configuration : 3310 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec

no service password-encryption

!

hostname R11

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 11000 11999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 11.11.11.11 255.255.255.255

!

interface GigabitEthernet0/0

ip address 130.1.1.11 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/1

ip address 120.1.1.11 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 11.11.11.11 0.0.0.0 area 0

network 120.1.1.0 0.0.0.255 area 0

network 130.1.1.0 0.0.0.255 area 0

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

mpls ldp router-id Loopback0

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

PE2-R12#show running-config Building configuration...

Current configuration : 4938 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname PE2-R12

!

boot-start-marker boot-end-marker

!

!

vrf definition CLIENT-A rd 3:4

route-target export 200:200

route-target import 100:100

!

address-family ipv4 exit-address-family

!

vrf definition CLIENT-B rd 3:5

route-target export 400:400

route-target import 300:300

!

address-family ipv4 exit-address-family

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated mpls label range 12000 12999

mpls label protocol ldp mpls traffic-eng tunnels

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 12.12.12.12 255.255.255.255

!

interface Tunnel11

ip unnumbered Loopback0 tunnel mode mpls traffic-eng tunnel destination 8.8.8.8

tunnel mpls traffic-eng autoroute announce

tunnel mpls traffic-eng path-option 1 explicit name TUNNEL11-PE2-PE1

tunnel mpls traffic-eng fast-reroute no routing dynamic

!

interface Tunnel22

ip unnumbered Loopback0 mpls ip

tunnel mode mpls traffic-eng tunnel destination 9.9.9.9

tunnel mpls traffic-eng autoroute announce tunnel mpls traffic-eng path-option 1 dynamic no routing dynamic

!

interface GigabitEthernet0/0

ip address 110.1.1.12 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels mpls ip

!

interface GigabitEthernet0/1

ip address 120.1.1.12 255.255.255.0

duplex auto speed auto media-type rj45

mpls traffic-eng tunnels

mpls traffic-eng backup-path Tunnel22 mpls ip

!

interface GigabitEthernet0/2 vrf forwarding CLIENT-A

ip address 150.1.1.12 255.255.255.0

duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 vrf forwarding CLIENT-B

ip address 200.1.1.12 255.255.255.0

duplex auto speed auto media-type rj45

!

router ospf 1

mpls traffic-eng router-id Loopback0 mpls traffic-eng area 0

network 12.12.12.12 0.0.0.0 area 0

network 110.1.1.0 0.0.0.255 area 0

network 120.1.1.0 0.0.0.255 area 0

!

router bgp 65000

bgp router-id 12.12.12.12 bgp log-neighbor-changes

neighbor 8.8.8.8 remote-as 65000 neighbor 8.8.8.8 update-source Loopback0 neighbor 8.8.8.8 send-community both

!

address-family vpnv4 neighbor 8.8.8.8 activate

neighbor 8.8.8.8 send-community both exit-address-family

!

address-family ipv4 vrf CLIENT-A redistribute connected

redistribute static exit-address-family

!

address-family ipv4 vrf CLIENT-B redistribute connected

neighbor 200.1.1.14 remote-as 600

neighbor 200.1.1.14 activate exit-address-family

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

ip route vrf CLIENT-A 13.13.13.13 255.255.255.255

150.1.1.13

!

ip explicit-path name TUNNEL11-PE2-PE1 enable next-address 120.1.1.11

next-address 130.1.1.10

next-address 170.1.1.8

!

!

!

mpls ldp router-id Loopback0

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R13#sh run

Building configuration...

Current configuration : 3005 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R13

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 13.13.13.13 255.255.255.255

!

interface GigabitEthernet0/0 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2

ip address 150.1.1.13 255.255.255.0

duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown

duplex auto speed auto media-type rj45

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

ip route 0.0.0.0 0.0.0.0 180.1.1.8

ip route 0.0.0.0 0.0.0.0 150.1.1.12

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

end

R14# show running-config Building configuration...

Current configuration : 3056 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R14

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 14.14.14.14 255.255.255.255

!

interface GigabitEthernet0/0 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3

ip address 200.1.1.14 255.255.255.0

duplex auto speed auto media-type rj45

!

router bgp 600

bgp log-neighbor-changes

network 14.14.14.14 mask 255.255.255.255

neighbor 200.1.1.12 remote-as 65000

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R15# show running-config Building configuration...

Current configuration : 3055 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R15

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure

no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 15.15.15.15 255.255.255.255

!

interface GigabitEthernet0/0 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/1

ip address 190.1.1.15 255.255.255.0

duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto

speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

router bgp 500

bgp log-neighbor-changes

network 15.15.15.15 mask 255.255.255.255

neighbor 190.1.1.8 remote-as 65000

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

End

R16# show running-config Building configuration...

Current configuration : 2969 bytes

!

version 15.6

service timestamps debug datetime msec service timestamps log datetime msec no service password-encryption

!

hostname R16

!

boot-start-marker boot-end-marker

!

!

!

no aaa new-model ethernet lmi ce

!

!

!

mmi polling-interval 60 no mmi auto-configure no mmi pvc

mmi snmp-timeout 180

!

!

!

!

!

!

!

!

!

!

!

ip cef

no ipv6 cef

!

multilink bundle-name authenticated

!

!

!

!

!

redundancy

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback0

ip address 16.16.16.16 255.255.255.255

!

interface GigabitEthernet0/0

ip address 180.1.1.16 255.255.255.0

duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/1 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/2 no ip address

shutdown duplex auto speed auto media-type rj45

!

interface GigabitEthernet0/3 no ip address

shutdown duplex auto speed auto media-type rj45

!

ip forward-protocol nd

!

!

no ip http server

no ip http secure-server

ip route 0.0.0.0 0.0.0.0 180.1.1.8

!

!

!

!

control-plane

!

banner exec ^C

!

line con 0

line aux 0

line vty 0 4 login

transport input none

!

no scheduler allocate

!

end

## Anexa 2

RP/0/RP0/CPU0:R1#sh running-config Sun Jun 19 17:07:22.646 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Sun Jun 19 17:07:13 2022 by admin

!

hostname R1 username admin group root-lr

group cisco-support secret 10

$6$lIiAW1Z9Xsm6CW1.$Zc0mtuf8SnYiYiIyK9nDVr tWmQhk09VvX05neqnm6SzGmuyfh3QDohYaexXi98 CuwE9EmVmhtwlV6JVW3bP3n/

!

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 1.1.1.1 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 20.1.1.1 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 10.1.1.1 255.255.255.0

!

interface GigabitEthernet0/0/0/2 shutdown

!

interface GigabitEthernet0/0/0/3 shutdown

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0001.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

fast-reroute per-prefix tiebreaker node-protecting index 100

microloop avoidance segment-routing mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only passive

address-family ipv4 unicast prefix-sid index 1

!

!

interface GigabitEthernet0/0/0/0 circuit-type level-2-only

point-to-point

address-family ipv4 unicast fast-reroute per-prefix

fast-reroute per-prefix ti-lfa

!

!

interface GigabitEthernet0/0/0/1 circuit-type level-2-only

point-to-point

address-family ipv4 unicast fast-reroute per-prefix

fast-reroute per-prefix ti-lfa

!

!

!

end

RP/0/RP0/CPU0:XR2#sh running-config Sun Jun 19 17:09:02.811 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Sun Jun 19 16:22:13 2022 by admin

!

hostname XR2 username admin group root-lr

group cisco-support secret 10

$6$oMjalqfNAO41l...$wULKhwRr1bOOTvPU7rptih/ UYr0zSk7eg7YlSgJUqw2k.51s724dYenld9gT/k o0rgAl2STsANjtxcb92FPiE.

!

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 2.2.2.2 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0

shutdown

!

interface [GigabitEthernet0/0/0/0@linux.it](mailto:GigabitEthernet0/0/0/0@linux.it)> ipv4 address 20.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 60.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/2 ipv4 address 30.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/3 ipv4 address 80.1.1.2 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0002.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only passive

address-family ipv4 unicast prefix-sid index 2

!

!

interface GigabitEthernet0/0/0/0 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

!

End

RP/0/RP0/CPU0:XR3#sh running-config Sun Jun 19 17:09:29.607 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Sun Jun 19 17:05:00 2022 by admin

!

hostname XR3 username admin group root-lr

group cisco-support secret 10

$6$pXbt91sNymajA91.$1RSq9EGZ.yoezuP4EXDVW G6VPzvUudaSWMKiwYgjeI.IP0FCPXXeeCgjFxPotl BvsbpBNOs.f077.4or41Xp7.

!

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 3.3.3.3 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 50.1.1.3 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 10.1.1.3 255.255.255.0

!

interface GigabitEthernet0/0/0/2 shutdown

!

interface GigabitEthernet0/0/0/3 ipv4 address 80.1.1.3 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0003.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only

passive

address-family ipv4 unicast prefix-sid index 3

!

!

interface GigabitEthernet0/0/0/0 circuit-type level-2-only

point-to-point

address-family ipv4 unicast metric 50

!

!

interface GigabitEthernet0/0/0/1 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

!

end

sRP/0/RP0/CPU0:XR4#sh runing-config Sun Jun 19 17:37:37.223 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Sun Jun 19 17:35:59 2022 by admin

!

hostname XR4 username admin group root-lr

group cisco-support secret 10

$6$8RGJZ1bIJtJJCZ1.$T1AaIb1dkwNBHrH5A2qR5V

0Ifm.Dnm4anuLCJ1MEDcioxm/uiB0OKvEKLavpPH UlMJrUz5j64ELHK/mA/1m/8.

!

ipv4 unnumbered mpls traffic-eng Loopback0 call-home

service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 4.4.4.4 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 70.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 40.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/2 ipv4 address 30.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/3 shutdown

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0004.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only passive

address-family ipv4 unicast prefix-sid index 4

!

!

interface GigabitEthernet0/0/0/0 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

!

mpls traffic-eng

!

end

RP/0/RP0/CPU0:XR4#

[ 1.848266] audit: typ

RP/0/0/CPU0:R6#sh running-config Sun Jun 19 17:10:04.977 UTC

Building configuration...

!! IOS XR Configuration 6.0.1

!! Last configuration change at Sun Jun 19 16:24:06 2022 by admin

!

hostname R6 interface Loopback0

ipv4 address 6.6.6.6 255.255.255.255

!

interface MgmtEth0/0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 70.1.1.6 255.255.255.0

!

interface GigabitEthernet0/0/0/1 shutdown

!

interface GigabitEthernet0/0/0/2 shutdown

!

router isis 1

is-type level-2-only

net 49.0000.0000.0006.00

log adjacency changes address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only passive

address-family ipv4 unicast prefix-sid index 6

!

!

interface GigabitEthernet0/0/0/0 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

!

end

RP/0/RP0/CPU0:XR5#sh running-config Sun Jun 19 17:10:28.416 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Sun Jun 19 17:05:56 2022 by admin

!

hostname XR5 username admin group root-lr

group cisco-support secret 10

$6$lcFf1OouhC1. $sX5rfrNZT2PPqPdUABD9xp.vI

ViYl8mEIEFhzq6.6mFjkAf92pi.m/L6eEKdMg ILPJOISMOKYYsm8gRO.ejvh/

!

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 5.5.5.5 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 50.1.1.5 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 40.1.1.5 255.255.255.0

!

interface GigabitEthernet0/0/0/2 shutdown

!

interface GigabitEthernet0/0/0/3 ipv4 address 60.1.1.5 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0005.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 circuit-type level-2-only passive

address-family ipv4 unicast prefix-sid index 5

!

!

interface GigabitEthernet0/0/0/0

circuit-type level-2-only point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 circuit-type level-2-only

point-to-point

address-family ipv4 unicast metric 100

!

!

interface GigabitEthernet0/0/0/3 circuit-type level-2-only

point-to-point

address-family ipv4 unicast

!

!

!

end

## Anexa 3

RP/0/RP0/CPU0:XR1#show running-config Fri Jun 17 14:18:12.318 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Thu Jun 16 18:48:37 2022 by admin

!

hostname XR1 username admin group root-lr

group cisco-support secret 10

$6$ligfx19vOziqDx1.$Hj8H2GW/o5L.jWIsMdbAfMu qTIp5G5KLz/VtDN8pBRostkIkxzqUrg5Vmb2kKlMN v.WMokDaGa32epkI5kWyu0

!

ipv4 unnumbered mpls traffic-eng Loopback0 call-home

service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 1.1.1.1 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 20.1.1.1 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 10.1.1.1 255.255.255.0

!

interface GigabitEthernet0/0/0/2

ipv4 address 110.1.1.1 255.255.255.0

!

interface GigabitEthernet0/0/0/3

ipv4 address 100.1.1.1 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0001.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0

passive

address-family ipv4 unicast prefix-sid index 1

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 point-to-point

address-family ipv4 unicast metric 10

!

!

!

mpls traffic-eng

!

segment-routing

global-block 16000 23999 traffic-eng

segment-list SIDLIST1 index 10 mpls label 16002

index 20 mpls label 16005

index 30 mpls label 16004

!

policy POLICY1

color 10 end-point ipv4 4.4.4.4 candidate-paths

preference 100 dynamic metric

type igp

!

!

!

!

!

policy POLICY2

color 20 end-point ipv4 4.4.4.4 candidate-paths

preference 200

explicit segment-list SIDLIST1

!

!

!

!

pcc

source-address ipv4 1.1.1.1 pce address ipv4 5.5.5.5 precedence 100

!

!

!

!

end

RP/0/RP0/CPU0:XR2#show running-config Fri Jun 17 14:20:01.468 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Thu Jun 9 17:37:28 2022 by admin

!

hostname XR2 username admin group root-lr

group cisco-support secret 10

$6$E2HDeloX8xc0e...$/ORoJwSiAq2YRXO4aBz4Zu ODbpUFJJVgDcJfw8LXP6Id7aq2cXrAmL96YJICdk US0T7spm4VKjWJ0FAFtDmlL/

!

ipv4 unnumbered mpls traffic-eng Loopback0 call-home

service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 2.2.2.2 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 20.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 60.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/2 ipv4 address 30.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/3 ipv4 address 80.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/4

ipv4 address 130.1.1.2 255.255.255.0

!

interface GigabitEthernet0/0/0/5

shutdown

!

interface GigabitEthernet0/0/0/6 shutdown

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0002.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 passive

address-family ipv4 unicast prefix-sid index 2

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/4 point-to-point

address-family ipv4 unicast

!

!

!

mpls traffic-eng

!

segment-routing

global-block 16000 23999 traffic-eng

pcc

source-address ipv4 2.2.2.2 pce address ipv4 5.5.5.5 precedence 100

!

!

!

!

End

RP/0/RP0/CPU0:XR3# show running-config Fri Jun 17 14:20:48.995 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Thu Jun 9 17:36:37 2022 by admin

!

hostname XR3 username admin group root-lr

group cisco-support secret 10

$6$v.KDT0Upnbxw7T0.$mQ8HJfLK3wH8nPgWbsdZ w2Jwmc0tbJuTut8j1aDL6YS5/x9s/hIHinSUgwsGamL pyI2uMX8YQUJwmIyWVK/bF/

!

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 3.3.3.3 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 50.1.1.3 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 10.1.1.3 255.255.255.0

!

interface GigabitEthernet0/0/0/2 ipv4 address 90.1.1.3 255.255.255.0

!

interface GigabitEthernet0/0/0/3 ipv4 address 80.1.1.3 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0003.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 passive

address-family ipv4 unicast prefix-sid index 3

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 point-to-point

address-family ipv4 unicast

!

!

!

segment-routing

global-block 16000 23999 traffic-eng

pcc

source-address ipv4 3.3.3.3 pce address ipv4 5.5.5.5 precedence 100

!

!

!

!

End

RP/0/RP0/CPU0:XR4# show running-config Fri Jun 17 14:22:00.428 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Thu Jun 9 17:36:00 2022 by admin

!

hostname XR4 username admin group root-lr

group cisco-support secret 10

$6$RCT3X0nLlcJA8X0.$M3/poxcBtHvaE99kxhyEOI MBtSr.lY.kdZTr8XrCyhI7PHLWp0eriL/wM1oK4N/o z8jqyrJ5qOMjLYxe3SCC21

!

ipv4 unnumbered mpls traffic-eng Loopback0

call-home service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 4.4.4.4 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 70.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 40.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/2 ipv4 address 30.1.1.4 255.255.255.0

!

interface GigabitEthernet0/0/0/3

ipv4 address 100.1.1.4 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0004.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 passive

address-family ipv4 unicast prefix-sid index 4

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 point-to-point

address-family ipv4 unicast

!

!

!

mpls traffic-eng

!

segment-routing

global-block 16000 23999 traffic-eng

pcc

source-address ipv4 4.4.4.4 pce address ipv4 5.5.5.5 precedence 100

!

!

!

!

end

RP/0/RP0/CPU0:XR5#show running-config Fri Jun 17 14:21:28.910 UTC

Building configuration...

!! IOS XR Configuration 7.6.1

!! Last configuration change at Thu Jun 9 17:34:14 2022 by admin

!

hostname XR5 username admin group root-lr

group cisco-support secret 10

$6$iTBaV1LXNsM4CV1.$TDMCdMWh7sH28N7GS

3K.qJheFT5KgIz.U0/EGV57nliKBfbgRX6YjujQ2fbkz 35dvxZYgMT4MqtzIjUQPsqu21

!

pce

address ipv4 5.5.5.5

!

ipv4 unnumbered mpls traffic-eng Loopback0 call-home

service active

contact smart-licensing profile CiscoTAC-1 active

destination transport-method email disable destination transport-method http

!

!

interface Loopback0

ipv4 address 5.5.5.5 255.255.255.255

!

interface MgmtEth0/RP0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 50.1.1.5 255.255.255.0

!

interface GigabitEthernet0/0/0/1 ipv4 address 40.1.1.5 255.255.255.0

!

interface GigabitEthernet0/0/0/2 shutdown

!

interface GigabitEthernet0/0/0/3 ipv4 address 60.1.1.5 255.255.255.0

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0005.00

distribute link-state log adjacency changes

address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 passive

address-family ipv4 unicast prefix-sid index 5

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3 point-to-point

address-family ipv4 unicast

!

!

!

mpls traffic-eng

!

segment-routing

global-block 16000 23999 traffic-eng

!

!

End

RP/0/0/CPU0:XR6#show running-config Fri Jun 17 14:22:55.926 UTC

Building configuration...

!! IOS XR Configuration 6.0.1

!! Last configuration change at Thu Jun 9 17:35:07 2022 by admin

!

hostname XR6

ipv4 unnumbered mpls traffic-eng Loopback0 interface Loopback0

ipv4 address 6.6.6.6 255.255.255.255

!

interface MgmtEth0/0/CPU0/0 shutdown

!

interface GigabitEthernet0/0/0/0 ipv4 address 70.1.1.6 255.255.255.0

!

interface GigabitEthernet0/0/0/1

ipv4 address 130.1.1.6 255.255.255.0

!

interface GigabitEthernet0/0/0/2

ipv4 address 110.1.1.6 255.255.255.0

!

interface GigabitEthernet0/0/0/3

ipv4 address 120.1.1.6 255.255.255.0

!

interface GigabitEthernet0/0/0/4 shutdown

!

router isis 1

is-type level-2-only

net 49.0001.0000.0000.0006.00

log adjacency changes address-family ipv4 unicast metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0 segment-routing mpls sr-prefer

!

interface Loopback0 passive

address-family ipv4 unicast prefix-sid index 6

!

!

interface GigabitEthernet0/0/0/0 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/1 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/2 point-to-point

address-family ipv4 unicast

!

!

interface GigabitEthernet0/0/0/3

point-to-point

address-family ipv4 unicast

!

!

!

mpls traffic-eng

!

segment-routing

global-block 16000 23999

!

End