



ÓBUDAI EGYETEM

**Kandó Kálmán villamosmérnöki kar
Híradástechnika intézet**

OTDK dolgozat

MPLS és IP hálózatok

**OE-KVK
2015**

**készítette: Maklár Tamás Zoltán
konzulens: Dr. Gyányi Sándor**

Tartalomjegyzék

Tartalomjegyzék	1
1. MPLS eljárás ismertetése	2
1.1. MPLS történeti áttekintés	2
1.2. MPLS vezérlés és átvitel	3
1.4. MPLS hálózati elemek	3
1.5. Címke felépítése	4
1.6. Továbbítás felépítése	5
1.7. Vezérlés	7
1.8. LDP ismertetése	9
1.9. Traffic Engineering	10
1.10. Resource Reservation Protocols – Traffic Engineering (RSVP-TE)	10
1.11. MPLS-VPN funkciók	11
1.12. GMPLS (General-MPLS).....	12
1.12.1. Adatátviteli sík	12
1.12.2. Jelzési sík.....	13
1.13. Összehasonlítás más hálózatokkal, összegzés	13
2. Hálózatok tervezése és implementációja GNS3 szimulációs környezetben	15
2.1. GNS3 (Graphical Network Simulator) bemutatása	15
2.2. MPLS példahálózat tervezése és implementációja GNS3-ban	26
2.3. MPLS gerinc magja, LSR-ek létrehozása	27
2.4. Edge-LSR-ek, LER-ek hozzáadása	35
2.5. VPN hozzáadása MPLS gerinchez, P és PE router-ek kialakítása	38
2.6. Customer Edge eszközök csatlakoztatása, PE eszközökben VRF-ek létrehozása	40
2.7. Összegzés	56
Összefoglalás.....	57
Irodalomjegyzék	58

1. MPLS eljárás ismertetése

1.1. MPLS történeti áttekintés

Az internet létrejöttével és fejlődésével, egyre nagyobb lett az igény nagy adatátviteli sebességű alkalmazások használatára. Az OSI modell szempontjából a 2-ik és 3-ik rétegbeli eszközök ár, teljesítmény arányban igen nagy eltérést mutattak. A Layer2-ben működő kapcsolók, vagy switchek igen nagy sebességgel voltak képesek továbbítani a csomagokat, viszonylag alacsony költséggel, a Layer3 eszközök, forgalomirányítók viszont sokkal rugalmasabb csomagtovábbítási algoritmusokkal rendelkeztek.

Ezen a gondolatmeneten végig haladva, természetesnek tűnt, hogy a két réteg előnyeit ötvözni kell egy protokollban. Első ilyen megoldás az IP, mint hálózati, az ATM, mint adattovábbítási réteg szintjén jelent meg. Az IP-t népszerűsége, az ATM-et pedig sebessége miatt tekintették előnyösnek. Megindultak a szabványosítási kísérletek az IP és az ATM integrálására. Az első ilyen szabványok az IP-over-ATM (RFC1483, RFC1577) a LAN Emuláció és a MultiProtocol-over-ATM (MPOA) voltak.

Az első cellatovábbításon alapuló router-ek a Toshiától jelentek meg, majd később az Ypsilon-tól az IP vezérelt ATM switchek. A Cisco által kiadott Tag-switching, már nem csak ATM-mel, hanem szinte, az összes adatkapcsolati eljárás felett működött. A Cisco előtt nem sokkal az IBM is megjelent egy hasonló Aggregate Route-based IP switching (ARIS) rendszerrel.

Mivel a cégek különböző kutatási irányban haladtak, ezért szükség volt a központi szabványosításra, 1997-ben megalakult az IETF MPLS munkacsoportja Cisco elnökség mellett.

Történeti áttekintés után következzen az MPLS architektúra ismertetése.

1.2. MPLS architektúra

A klasszikus IP hálózatokban a csomagtovábbításnál a forgalomirányítók minden egyes csomag IP fejrészét analizálják, majd ez alapján hoznak döntést a továbbításról. Dinamikus illetve statikus route táblák segítségével történik meg a cél IP cím felé való továbbítás. Az MPLS egy címkekapcsolási módszer, ami integrálja Layer2 switching és Layer3 routing előnyeit. Ahogyan a Layer2-es hálózatok (ATM, Frame Relay), az MPLS is címkét rendel a csomaghoz, a csomagalapú illetve a cellalapú hálózatokon való átvitelhez. Az átvitel alapja a címke kicserélésén (label swapping) alapul. A címke rövid, rögzített hosszúságú és jelöli adott forgalomirányító esetén a továbbítás módját.

Fontos különbség még a hagyományos gerinc, WAN technológiák és az MPLS között, hogy címkék stack-elhetőek, ami lehetővé teszi új alkalmazások használatát, mint például a VPN vagy Traffic Engineering. Az MPLS hálózatokat az IP csomag továbbításának szempontjából nevezhetjük MPLS felhőnek is, mert az adatkapcsolat nem biztos, hogy homogén rendszert alkot, lehet egy hálózatban Carrier Ethernet, ATM, Frame Relay technológia is egyszerre, de ez a MPLS továbbítás szempontjából nem számít.

1.3. MPLS vezérlés és átvitel

Az MPLS-ben alapvetően elkülöníthető egymástól az átvitel, és a vezérlés. Az átvitel felelős a csomagok kapcsolásáért és a csomagra ragasztott címke cseréléseért a címke adatbázis alapján. A vezérlés pedig a megfelelő címke adatbázis kialakításáért és karbantartásáért felelős.

Az MPLS eszközök futtatnak egy, esetleg több routing protokollt, hogy a címkeadatbázist ki tudják cserélni egymással. Ehhez elengedhetetlen, hogy a hálózatban található eszközök IP címeit tudják. Tulajdonképpen minden eszköz egyszerre router és switch is az MPLS felhőben. Az IP routing továbbá azt a célt is szolgálja, hogy az IP és címke összerendelés alapján felépülhessen a címkekapcsolási tábla.

A címkekapcsolási tábla hirdetését az IETF által szabványosított LDP (Label Distribution Protocol) végzi, Cisco eszközök esetén még szóba jöhet TDP (Tag Distribution Protocol). Tekintsük át, hogyan is épül fel egy MPLS hálózat!

1.4. MPLS hálózati elemek

Az MPLS hálózatnak, vagy más néven MPLS felhőnek alapvetően két hálózati eleme van. Az egyik LSR a másik pedig az ELSR vagy LER eszközök. Az LSR, Label Switch Router eszköz tulajdonképpen egy kapcsoló, amely képes a címke alapján továbbítani a csomagokat, valamint képes címke kicserélésére és címkekapcsolási tábla hirdetésére. Ezek az eszközök, alkotják a hálózat magját. Az ELSR vagy LER, Edge-LSR vagy Label Edge Router feladata az MPLS hálózatba beérkező csomagokat ellátni címkével, valamint a kilépő csomagok esetén eltávolítani a címkét, ehhez viszont már útvonal-számítási feladatok is társulnak, tehát tekinthetjük a hálózat szempontjából forgalomirányítónak. A LER-ek nevéből is adódik, hogy az MPLS felhő szélén helyezkednek el, ők tartják a kapcsolatot más hálózatokkal is.

1.5. Címke felépítése

Az MPLS hálózatban a csomagtovábbítás a címkekapcsoláson alapul, ahhoz megértsük ezt a mechanizmust meg kell vizsgálnunk, hogy hogyan is épül fel MPLS címke, amit a 1.1-es ábrán láthatunk:

20 bit	3 bit	1 bit	8 bit
Címke értéke	EXP	S	TTL

1.1. ábra MPLS címke felépítése

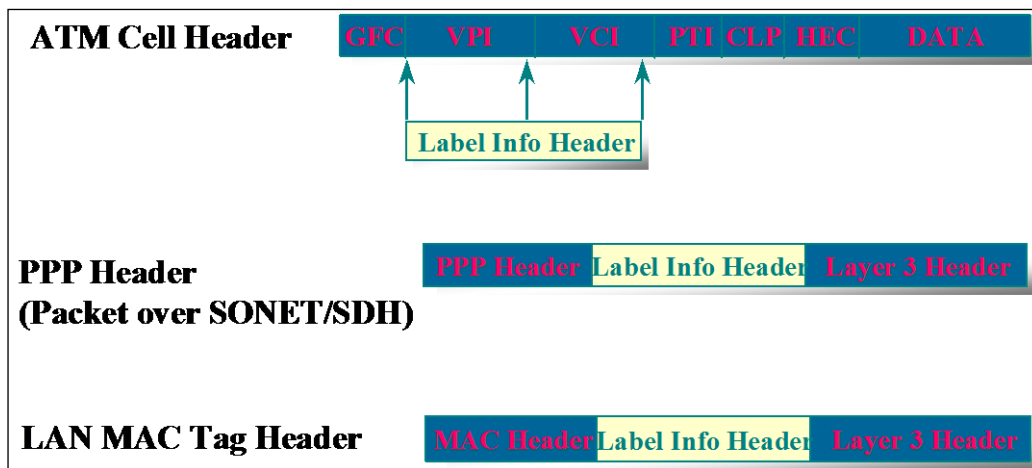
Ebből:

- 20 bit a címke értéke, az első 16 érték jelzésre és egyéb kommunikáció számára foglalt.
- Az EXP 3 bitje vagy más néven Class of Service hasonlóan működik az IP fejrész ToS bitjéhez.
- 1 bit bottom of Stack a címke verem alját jelzi ha több címke van, akkor elértük-e többszintű címkézés alját.
- 8 bit TTL ami a Time-to-live a csomag élettartamát jelöli hasonlóan az IP csomagban található TTL mezőhöz, célja hogy elkerüljük a hurkok kialakulását.

A címkének alapvetően 2 előfordulása lehetséges, az egyik esetben a Layer2 fejrész részévé válik, a másik esetben pedig az adattovábbítási réteg és a hálózati réteg fejrésze közé ékelődik be.

Az első esetben ATM alapú MPLS-ről van szó, ilyenkor a címke az ATM cella VPI/VCI, mezőjébe kerül.

A második esetben Frame alapú MPLS-ről van szó, amikor a címke a MAC és hálózati réteg közé ékelődik be. Ez valamivel több funkcióval rendelkezik, mint az ATM esetben, ugyanis az ATM cellának nem lesz EXP, S és TTL mezője sem.

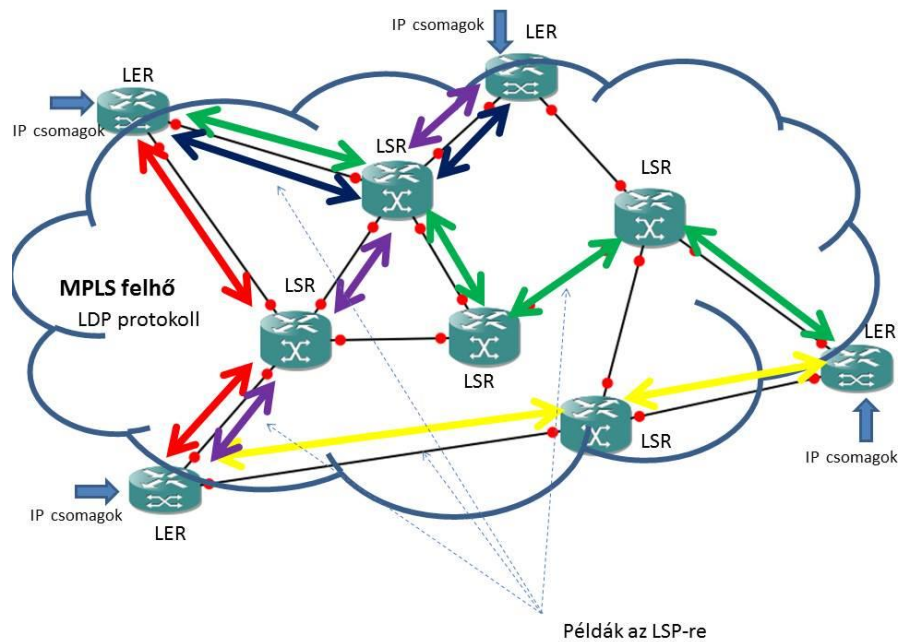


1.2. ábra Címke beépülése adatkapcsolati réteg és a hálózati réteg közé

1.6. Továbbítás felépítése

A továbbítás roppant egyszerű algoritmus, feladata az csupán, hogy a bejövő csomagról eltávolítja a címkét, majd ezt a címkét összehasonlítja a címke összekapcsolási táblában található értékkel, és az alapján továbbítja a csomagot a megfelelő interfészen a címke kicserélése után. A továbbítás a hálózatban alábbi módon épül fel:

1. A hálózat felderítése routing alapján. Az LDP összerendeli a címkéket az egyes útvonalakkal.
2. Az LDP hirdeti a hálózaton a többi LSR illetve LER felé a címke összerendeléseket.
3. Az LER kap egy IP csomagot, amit besorol egy ekvivalens átviteli osztályba (Forwarding Equivalence Class - FEC) ide azok a csomagok kerülnek, amiket az MPLS hálózaton belül azonosan kezelnek. Az IP fejrész analizálás után routing döntést hoz, a megfelelő útvonalhoz tartozó címkét illeszti az IP fejrész elé, majd továbbítja az MPLS felhőbe.
4. Az LSR-k csomagokat címke alapján továbbítják, közben címke cserélést hajtanak végre. Megkeresik a címke alapján, hogy melyik porton kell továbbítaniuk a csomagot.
5. Az MPLS hálózatot elhagyó csomagnál az LER eltávolítja a címkét, és az IP fejrész analizálása után továbbítja a megfelelő interface-n.



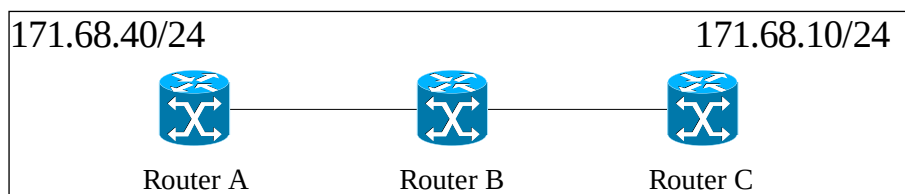
1.3. ábra MPLS felhő hálózati elemek és kapcsolatok

A továbbításból jól látszik, hogy összesen két esetben történik meg az IP fejrészének analízálása: amikor az IP csomag belép a hálózatba, és akkor, amikor elhagyja azt. A 1.3-as ábrán még fel lett tüntetve a lehetséges LSP vagy más néven címkekapcsolt útvonal létrejötte, amit zöld, piros, kék, lila és sárga színnel jelöltünk. Az LSP tulajdonképpen ez LSR-ek sorrendiségét adja meg az MPLS felhő egy részén át. A címkézett csomagok ezeken az LSP-ken keresztül haladnak át, amíg el nem érik a következő LER-t. A LSR-ek fordított sorrendiségét már egy másik LSP fogja meghatározni. Az ábrából is jól kivehető, hogy az LSR-ek vagy LER-ek a felhő szélén helyezkednek el, egyszerre részei az MPLS felhőnek, de a további hálózatokkal is kapcsolatban állnak. Az LSR-k ezzel szemben csak a felhőben szerepelnek, nem létesítenek közvetlen kapcsolatot a hálózat határain túl.

1.7. Vezérlés

Eddigiekben az MPLS architektúra funkcionális elemeit ismertettük, a továbbiakban vezérlésről, és a címkehirdetésről lesz szó.

Ahogy már eddig is láthattuk, az MPLS-ben a vezérlés feladata a megfelelő címke összekapcsolási tábla létrehozása és karbantartása. Ez a folyamat routing protokollal kezdődik, amellyel kialakulnak az útvonalak. Ezután minden LSR elvégzi az útvonalak és címkék összerendelését, majd ezeket hirdetését a hálózaton. Fontos, hogy a felhőben az összerendelések egyértelműek legyenek. Az összekapcsolási tábla két különböző típusú címkéért felelős, a bejövő (incoming) és a kimenő (outgoing) címkéért. Attól függően, hogy melyik LSR határozza meg a hálózatban a ki- illetve bejövő címkét, beszélhetünk lokális illetve távoli összekapcsolásról. Lokális összekapcsolás esetén az útvonal címke összerendelését az LSR lokálisan végzi el. Távoli összekapcsolás esetén a szomszédos LSR mondja meg a címke, útvonal összerendelését. A címke összekapcsolás folyamata még két féle lehet ezen kívül, attól függően, hogy a bejövő illetve a kimenő címke hogyan képződött. Downstream összekapcsolásról beszélünk, amikor a bejövő címke lokálisan képződik, míg a kimenő címkét a szomszédos LSR határozza meg távoli összekapcsolással. A második esetben upstream összekapcsolásról van szó, amikor a kimenő címke képződik lokálisan, és a bejövő címkét pedig távoli összekapcsolással kapja. Downstream összekapcsolásnál is megkülönböztetünk két esetet, ehhez viszont tekintsük meg a 1.4-es ábrát:



1.4. ábra Példa Unsolicited downstream, Downstream on demand estekre

Unsolicited downstream esetén Router C hirdeti Router B-nek, hogy a 20-as címkét használja 171.68.10.0/24-es hálózathoz, majd Router B hirdeti Router A-nak a 171.68.10.0/24-s hálózatot mondjuk a 28-as címkével, és továbbítja Router C-nek. Ilyenkor minden egyes elérhető úthoz kapunk címkét, attól függetlenül, hogy szükség van-e rá vagy nincs.

Downstream on Demand esetén Router B igényel címkét Router C-től 171.68.10.0/24-es hálózathoz és ezt teszi Router A is. Ebben az esetben csak akkor kérünk címkét egy adott útvonalhoz, amikor szükséges.

A címke összekapcsolás kialakulásánál felmerül még egy kérdés, hogy ez adat vagy kontroll vezérelt legyen. Ez nagymértékben befolyásolja a teljesítményt, a skálázhatóságot és az átvitel robosztusságát. A címke összekapcsolási táblában a bejegyzések mindig valamilyen esemény bekövetkezésének eredményeként jönnek létre. Ez lehet adatcsomag, amit az LSR továbbít, vagy vezérlési információ, például routing update. Adatvezérelt esetben, a címkekapcsolási táblában a bejegyzések fix értékű flow után jönnek létre. Ilyenkor a flow mérete hatással van az MPLS hálózat teljesítményére, ugyanis, ha flow mérete túl kicsi, akkor túl gyakran jönnek létre és törlődnek bejegyzések a táblában, ami igen csak „megeszi” LSR erőforrásait. Viszont ha ez az érték túl nagy, akkor túl sok címke nélküli csomag lép be a hálózatba, ami szintén csökkenti az átviteli kapacitását. Ezen gondolatmenten végig haladva következik az a felismerés, hogy az LSR-knek muszáj routing funkcióval is rendelkezni ahhoz, hogy hagyományos IP csomagokat is továbbítsanak. Ebben az esetben nem használhatunk Layer2-es eszközöket LSR-ként. Ha a bejegyzés a címkekapcsolási táblában valamilyen vezérléstől származó információ hatására jön létre vagy szűnik meg, akkor kontroll vezérelt esetről beszélünk.

A címke összekapcsolási tábla hirdetésére kétféle módszer létezik. Az egyik módszer, hogy egy routing protokoll segítségével hirdetjük meg a címke összerendelést. Ez csak kontroll vezérelt esetben kivitelezhető, ugyanis a címke összekapcsolást a vezérléshez köti. A legfőbb előnye, hogy a címke összekapcsolás nem ellentétes a routing információval, a topológia változás azonnal követhető. Hátránya, hogy a routing információknak tartalmaznia kell a next-hop-t is, így összeköttetés állapot alapú protokollok esetén nem alkalmazható, viszont távolság-vektor alapúaknál már igen.

A második esetben új protokollt kell használni a címketerjesztésre, és hirdetésre, emiatt nő a rendszer komplexitása, de összeköttetés-állapot alapú protokolloknál is használható lesz. Erre a Label Distribution Protocol szolgál.

1.8. LDP ismertetése

Az LDP az IETF által kifejlesztett protokoll, amit az RFC 3036-ban definiáltak. Azt a két LSR-t, amelyek kommunikálnak LDP felett, LDP peer-eknek nevezzük, közöttük LDP session jön létre, ami TCP-t használ.

Az LDP üzeneteknek négy kategóriája van:

- DISCOVERY üzenet szolgál az LSR jelenlétének hirdetésére, és fenntartására. Az LSR-ek változó időközönként HELLO üzeneteket küldenek UDP csomagokkal. Alapvetően két funkciója van. Az egyik az alap (basic), a közvetlenül csatlakoztatott szomszédok felderítésére szolgál, ezért Link-Hello üzenetnek is nevezzük, ami multicast üzenetként az alhálózatban lévő LSR-eket címezi meg. A másik a kiterjesztett (extended), ebben az esetben egy meghatározott IP címre küldi a Targeted-Hello üzenetet.
- SESSION üzenetek két LSR közti session felépítésére, fenntartására és lebontására szolgálnak. Először az Initialization üzenetet küld az aktív LSR a passzívnak. Az üzenet minden információt tartalmaz a session felépítéséhez. Ha a passzív LSR megfelel a követelményeknek, akkor Keepalive üzenet küld, ha nem, akkor Notification üzenetet.
- ADVERTISEMENT üzenetek szolgálnak a címkék és az egyenlő átviteli osztályok közötti leképezés létrehozására, módosítására, és törlésére. Ezek lehetnek Label-request, Label-mapping vagy Label-release üzenetek. A Label-request akkor kerül küldésre, ha az LSR címkét kér az üzenet fogadójától, hogy egy csomagot megfelelő címkével küldhessen tovább. Az üzenet fogadója meghatározza az üzenetben szereplő FEC értékhez tartozó címkét és Label-mapping-gal válaszol a feladónak. Label-release üzenet pedig akkor kerül küldésre, ha már nincs szükség címkére.
- NOTIFICATION üzenetek szolgálnak a hibák jelzésére, valamint a session állapotáról adnak információt.

Az LDP önmagában nem képes a hálózati erőforrások kezelésére, így a Traffic Engineering funkciót sem támogatja, emiatt az IETF elkészítette az LDP bővítését a CR-LDP-t (Constraint-based Routed LDP). A CR-LDP-ben már implementálva lett az erőforrások lefoglalása, így a LER képes előre útvonalak foglalni a hálózatban.

1.9. Traffic Engineering

A Traffic Engineering funkció általában azt takarja bármilyen hálózat esetén, hogy a hálózat képes a megfelelő QoS paraméterekkel kezelni az információt, és ennek megfelelően továbbítja a hálózaton, valamint képes VPN (Virtual Private Network) kezelésére.

Ezt a funkciót csomagkapcsolt gerinc rendszerekben elsőként az ATM biztosította a különböző AAL osztályok létrehozásával. Az IP hálózatokban ugyanezt a feladatot a Resource Reservation Protocols – Traffic Engineering (RSVP-TE) látja el. Ez a protokoll alkalmazható MPLS felhőben is, de az MPLS rendelkezik saját erőforrás kezelő protokollal is amit IETF CR-LDP néven implementált.

1.10. Resource Reservation Protocols – Traffic Engineering (RSVP-TE)

Az RSVP-TE erőforrás kezelésre szolgál IP hálózatokban, viszont kibővítették címkekezeléssel is. Az RSVP-t a RFC 2205-ben definiálták, és IP hálózatok erőforrás kezelésére fejlesztették, hogy biztosítsa a megfelelő QoS paramétereket. Azzal hogy kibővítették címkekezeléssel, lehetőség van, hogy az azonos FEC-hez tartozó csomagok számára erőforrásokat foglaljon le. A forrás és a cél közötti erőforrás lefoglaláshoz az RSVP-TE két üzenetet használ. Ezek a PATH és a RESV üzenetek. A LER PATH üzenet az LSR-k számára. Minden útba eső LSR beilleszti a PATH üzenetbe a saját IP címét, így a kilépő LER tudni fogja, a hozzá érkező csomagok útvonalát. Az RSVP-TE nagyon fontos eleme, hogy az erőforrásokat nem a küldő fogja lefoglalni, hanem a cél eszköz határozhatja meg. Ez azért nagyon fontos, mert a fogadó fél támaszt követelményeket a hálózattal szemben, így neki kell gondoskodnia a az erőforrás igények nyilvántartásáról nem pedig a forrásnak. A belépő LER PATH üzenetére a kilépő LER egy RESV üzenettel fog válaszolni. A RESV üzenet ugyanazon az útvonalon halad végig mint a PATH üzenet, mindegyik LSR ellenőrzi, hogy a kívánt erőforrást biztosítani tudja-e a fogadó fél számára. Ha igen, továbbküldi az üzenetet, ha nem akkor RESV_ERROR üzenettel válaszol. Az erőforrás igény fenntartása miatt bizonyos időközönként a RESV üzeneteket újra kell küldeni.

1.11. MPLS-VPN funkciók

Az MPLS VPN alkalmazásának nagy előnye, hogy az előfizetők egyes telephelyei közt nem kell a hálózatban virtuális áramkört beállítani, elegendő az ügyfél forgalomirányítója és szolgáltató határ eszközei között beállítani a kapcsolatot. A különböző ügyfelek ugyanazt a fizikai közeget használják, de logikailag teljesen elkülönülnek egymástól, így nem láthatják egymás hálózatait. Az MPLS VPN ugyanazzal a biztonsági szinttel rendelkezik, mint a hagyományos VPN technikák. Az MPLS VPN támogatja az átfedő címtartományokat, intraneteket, extraneteket és hub-and-spoke technológiákat. A peer-to-peer topológiából fakadóan az MPLS felhőben nem kell tudni az eszközöknek a VPN létezéséről, elegendő csupán LER-kben implementálni. Az MPLS VPN a gerincszolgáltatók körében igen csak népszerű, ugyanis jóval olcsóbb és ugyanakkora biztonságot tud nyújtani, mint a hagyományos VPN módszerek. Ezek mellett képes biztosítani a skálázhatóságot, és lehetővé teszi az előfizetők hálózatának kisebb részekre bontását.

Az MPLS VPN-ben 3 különböző eszközt különböztetünk meg. A LSR-ek P (provider) router-ek, melyek az MPLS/VPN felhő gerincét alkotják, megmaradnak hagyományos LSR eszköznek. A LER eszközök lesznek a PE (Provider Edge) router-ek amik az MPLS/VPN felhő szélén helyezkednek el. Ezek az eszközök állnak kapcsolatban az ügyfél router-ével amiket CE (Customer Edge) routernek hívunk ebben a terminológiában. A CE routerek már csak hagyományos Layer3 kapcsolattal rendelkeznek, nem tekinthetők a MPLS gerinc részeként, ugyanis nem kell MPLS-t futtatniuk.

A P router-eknek nincs közvetlen kapcsolata a CE router-ekkel így ezek rejtve is maradhatnak, az ügyfél elől. Feladatuk a beérkező információ gyors csomagkapcsolása. A CE eszközök egy, vagy több PE router-rel kapcsolatban állnak, feladatuk az ügyfél csomagjainak forgalomirányítása a PE router-ek felé, ez hagyományos routing segítségével történik, ami lehet például BGP, OSPF, vagy EIGRP.

A PE router-ek, ellátják a hálózatba beérkező csomagokat címkével, és megfelelő VPN azonosítóval, továbbá kilépéskor eltávolítják címkét, és továbbítják az ügyfél hálózatba. Mivel a peer-to-peer topológia egyfajta korlátozást jelent a privát IP címek használatában, emiatt szükséges volt az MPLS/VPN-ben implementálni a Virtuális Router fogalmát.

Tulajdonképpen ez egy virtuális router a router-ben. Mivel egy PE több CE-hez csatlakozhat különböző VPN-ekkel, ezért szükséges a VRF (virtual routing forwarding) táblát létrehozni az egyes CE-ek számára. Az ügyfelek irányító táblái egymástól

különbözőek, de ugyanabban a PE-ben szerepelnek. Ezek a virtuális routerek ügyfél oldalról nézve dedikált router-ként látszanak .

A szolgáltató szempontjából viszont pont az ellenkezője látszik, ugyanis minden PE router-ben, minden előfizetőnek dedikált VRF-je van, így gerinc szempontjából minden ügyfél egy dedikált router-rel rendelkezik.

A VPN létrehozása a PE router-ek feladata, ezért minden ügyfél számára létre kell hozni VRF táblát. Az ügyféllel való kommunikációhoz a PE-n és CE-n is definiálni kell ezeket a táblákat. Az MPLS felhő felé az LSP-k felépítése és a VPN-ek létrehozása címkével történik. Ezt a MPLS/VPN-nél kétszintű címkézéssel oldják meg, amit stack segítségével alkalmaznak.

Ebben az esetben az alsó címke az, ami a PE routerek számára hordoz információt, míg a felső címkével történik meg a továbbítás a hálózatban.

1.12. GMPLS (General-MPLS)

Napjainkban az összes végpont IP csomagokkal kommunikál a külvilággal, éppen ezért szükséges volt egy olyan szabvány kidolgozása, ami kompatibilissé teszi az összes meglévő hálózatunkat az IP-vel. A GMPLS egy keretrendszert ad, ami a különböző multiplexelési képességű rétegeket hierarchiába szervezi, ahol az alsóbb rétegek, felsőbb rétegek forgalmát különböző átviteli csatornába szervezik.

Az ilyen csatornákat általánosított címkével látjuk el és ezeket az IP forgalomirányítás virtuális összeköttetésként értelmezi. A GMPLS szabványosítását IETF dolgozta ki. A definíciók két fontos területre különülnek el. Adatátviteli és Vezérlési síkra.

1.12.1. Adatátviteli sík

A GMPLS-nél a címke fogalma általános, nem csupán virtuális áramkör jelöl az IP csomagkapcsolás számára, hanem bármilyen forgalmi egységre levetíthető, amit egy adott hálózati réteg egységesen kezel. A GMPLS-ben az LSP (címkekapcsolt útvonal) két azonos rétegbeli eszköz között létrehozott virtuális útvonal. Ezzel megvalósítható a különböző adatkapcsolattal rendelkező hálózatok azonos menedzsmentje. A hálózat kezelő szoftvereknek nem kell foglalkozni az egyes hálózatok sajátosságaival, hanem elegendő a címkekapcsolt útvonal azonosítása. Az MPLS-hez képest a GMPLS lehetővé teszi az információ átvitel és a jelzés átvitel elkülönítését logikai és fizikai szinten egyaránt. Ennek jelentősége az optikai útvonalakon van, ugyanis a technológiák többsége nem képes egy sávon belül kezelni a kettőt.

1.12.2. Jelzési sík

Ahhoz, hogy lehetőség legyen automatikus LSP kiépítésre, menedzsmentre és lebontásra, függetlenül a hálózati rétegtől vagy annak fizikai megvalósításától, számos nézetben ki kellett egészíteni a fenn ismertetett „hagyományos” MPLS architektúra jelzés protokolljait. A GMPLS kialakításához olyan változtatásokra van szükség, mint a sávon kívüli jelzésrendszer, kétirányú LSP vagy az általános címke bevezetése. A kétirányú LSP a felhasználói igények miatt is felvetődött, továbbá lehetőség van SDH/SONET rendszerek integrálására, melyek alapvetően kétirányúak. Bevezetésre került még az optikai hálózatok hatékonyabb kezelése miatt a javasolt címke (suggested label) és a kiterjesztett hibakezelés. A kiterjesztések legfontosabb kérdésköre a skálázhatóság, hiszen a DWDM rendszerek megjelenésével az útvonalválasztónak képesnek kell lenni párhuzamos csatornák százait kezelni.

A GMPLS útvonalválasztás kiterjesztésének lényege, hogy az LSP-k virtuális összeköttetésként kerülnek hirdetésre a multiplexelési képességekkel és a védelmi képességekkel együtt. A forgalomirányító algoritmusuk ezek után különösebb módosítás nélkül képes kiválasztani a legmegfelelőbb útvonalat a továbbítás számára.

A sávon kívüli jelzés átvitel és a nem csomagkapcsolt hálózati rétegek egységes menedzsmentje olyan problémákra vet fényt, mint a dedikált jelzéscsatornák kezelése, vagy az általános címkék egyértelműségének biztosítása. Erre szolgál LMP protokoll (Link Management Protocol). A kétirányú dedikált jelzéscsatornák felépítéséről, fenntartásáról a LMP Hello protokoll gondoskodik. Továbbá a címkekiosztást, a védelmi eljárásokat, és az LSP-k felépítését is a LMP végzi.

1.13. Összehasonlítás más hálózatokkal, összegzés

Az MPLS hálózatok alapvetően abban különböznek a hagyományos csomagkapcsolt hálózatoktól, eljárásoktól, hogy OSI modell szerint se nem a második se nem a harmadik rétegben nem realizálható igazán. Hasonlóan a Frame Relay hálózatokhoz, címke alapján történik az útvonalválasztás, így az eszközök lehetnek Layer2 switch-ek, de a továbbításhoz szükséges a hálózati rétegbe is „fel kell nyúlni”, hiszen szükséges a címke összerendelési táblában az IP hálózatok ismerete is a megfelelő továbbításhoz. Tehát az MPLS-nek sikerült egy olyan modell létrehozása, amely egyesíti Layer2 switching és Layer3 routing előnyeit, valamint a különféle erőforrás kezelő protokollok segítségével, az összekötésmentes, és összeköttetés alapú rendszerek előnyeit összegyűrni egy komplett hálózatba. Mint láthattuk, az MPLS-nek nem csak az IP hálózatokban van jelentősége, hanem a

GMPLS megjelenésével bármilyen hálózatban alkalmazhatunk címkekapcsolást.

Tehát a hagyományos IP hálózatokkal szemben nagy előnye, hogy gyors csomagkapcsolást tesz lehetővé Layer3 switching segítségével. Az ATM-mel szemben alkalmazhatunk IP címezést, mégis összeköttetés orientált a kapcsolatunk. Továbbá biztosítani képes a QoS forgalomirányítást ellentétben más címkekapcsolt rendszerekkel.

Belátható, hogy ha szeretnénk egy olyan IP gerinc hálózatot használni, amely gyors kapcsolású, támogatja a QoS-t, VPN-t valamint olcsón implementálható meglévő hálózatba az eszközpark folyamatos fejlesztésével járó nagyobb beruházás nélkül, akkor az MPLS a legjobb választás.

A továbbiakban bemutatok egy olyan szoftvert, amellyel real-time tudjuk az ilyen hálózatokat szimulálni. Bemutatásra kerül egy MPLS gerinc VPN támogatással együtt.

2. Hálózatok tervezése és implementációja GNS3 szimulációs környezetben

2.1. GNS3 (Graphical Network Simulator) bemutatása

A hálózatok tervezéséhez manapság már bárkinek rendelkezésére állnak különböző szoftverek, amik más és más hálózati eszközöket támogatnak. Ebben az esetben a GNS3 nevű ingyenes szoftverre került a választás, ugyanis ez a szoftver támogatja a hálózatokban is igen nagy népszerűségnek örvendő Cisco IOS firmware-t, továbbá oktatási célra alkalmas ingyenes szoftver, aminek köszönhetően kooperatív képzésben résztvevő hallgatóként az Óbudai Egyetem Kandó Kálmán villamosmérnöki kar Híradástechnikai intézet számára egy laborfejlesztést tettem lehetővé, aminek témája a szoftverben szimulált hálózatok összeállítása, és analízálása. A GNS3 lehetővé teszi, a Cisco IOS real-time futását. Egyidejűleg nem csak egy CLI-ben (Console Line Interface) lehet dolgozni, hanem akár több eszközt összekötve, több is megnyitható és konfigurálható. A szoftver használata viszont igen erőforrás igényes, ugyanis a szoftverben szimulált eszközöknek, (amik lehetnek router-ek, tűzfalak, host-ok) biztosítani kell processzoridőt és memóriát a futáshoz. Ebből fakadóan azt, hogy mekkora és milyen hálózat szimulálható, befolyásolja az a gép, amin a szoftvert fut. GNS3 estén fontos az, hogy milyen operációs rendszeren fut, ugyanis elérhető Windows és Linux platformra egyaránt. Mint említettem, a GNS3 igen csak erőforrás igényes szoftver, ezért szerencsésebb Linux disztribúció alá feltelepíteni. Ebben az esetben az Ubuntu 13.10-es verziójára került a választás. A GNS3 legfrissebb verziója a 0.8.6-os, ami elérhető a <http://www.gns3.net/> honlapon. A tervezni kívánt MPLS hálózat is ezen a verzión kerül bemutatásra.

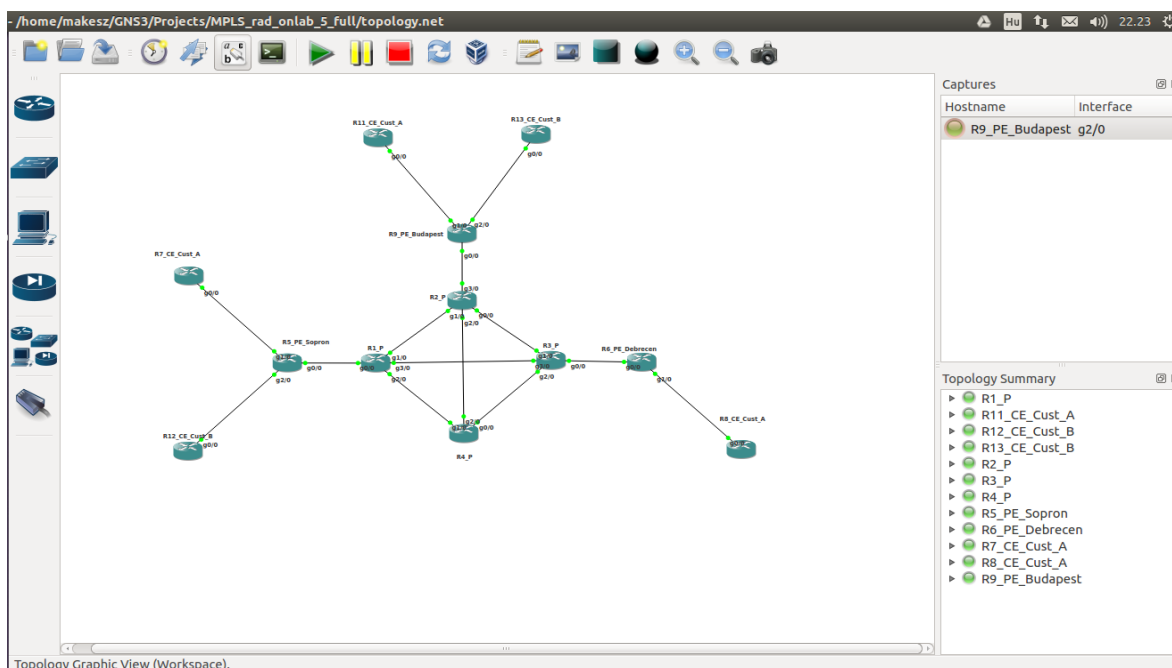
A GNS3 tulajdonképpen egy emulátorokból összeállított szoftvercsomag, egy grafikus interface. A Cisco IOS-ek futását a Dynamips nyílt forráskódú program teszi lehetővé. A GNS3-ba integrálható még Virtual Box is, ami a host-ok bekötését teszi lehetővé a hálózatba. A GNS3 telepítése terminálból a legegyszerűbb, amit Ubuntu-n a CTRL és T megnyomásával hozhatunk fel. Ahhoz, hogy legfrissebb verzió fusson, frissíteni kell az Ubuntu függőségi listáját, amit az alábbi parancsokkal lehet megtenni:

```
#:~$ sudo add-apt-repository ppa:gns3/ppa  
#:~$ sudo apt-get update
```

Miután frissült az adatbázis, megtörténhet a telepítés a következő paranccsal.

```
#:~$ sudo apt-get install dynamips gns3
```

A szoftver telepítése önmagában nem elég a munka elkezdéséhez, szükség van még IOS image fájlra is, amit be kell állítani a szoftverben. A GNS3 Cisco eszközökből hatféle eszközt támogat. Ezek Cisco C1700, C2600, C2691, C3600, C3700, C7200, PIX, ASA eszközök lehetnek. Az MPLS hálózatot érdemes olyan eszközökből összerakni, ami támogatja a gerinchálózati megoldásokat, így a hálózat tervezésnél a C7200-es router-ek lettek felhasználva. A szoftver felépítése az alábbi ábrán látható:



2.1. ábra GNS3 felépítése

A baloldali sávban az All Devices-ben található az összes felhasználható eszköz. Azok az eszközök, amik nem szürke színnel vannak jelen, szabadon a munkaterületre „húzhatóak” egér segítségével. Fontos megjegyezni, hogy a továbbiakban csak azok az eszközök és menüpontok kerülnek bemutatásra, amik elengedhetetlenek a hálózatunk megvalósításához. Így nem kerül tárgyalásra a program minden eleme.

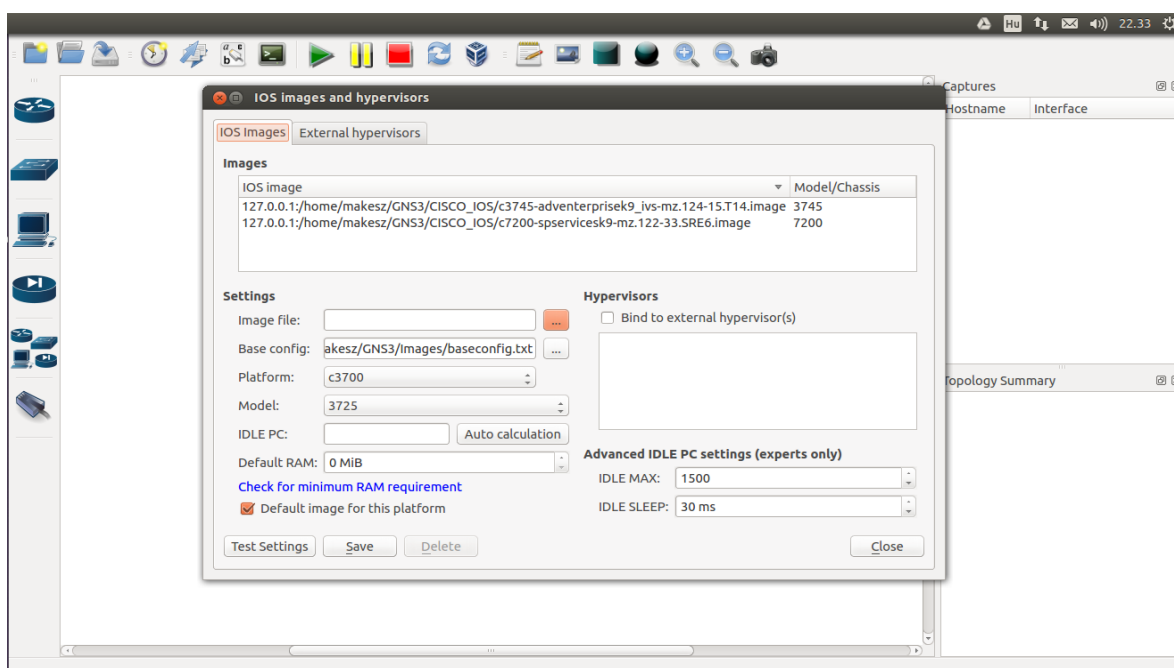
A Captures sávban található annak az összeköttetésnek egy tetszőleges interface-e, amin szeretnénk csomag analízist végrehajtani Wireshark segítségével. Ugyanis a GNS3 szoftvercsomag tartalmazza a Wireshark csomag analízátor szoftvert is, amit így külön nem kell telepíteni. A Topology Summary-ban találhatjuk az összes használt eszközt illetve az összes összeköttetés megtekinthető. Most, hogy a szoftver „felépítése” bemutatásra került, következzen az IOS image beállítása. Ezt az Edit menüpont alatt az

IOS Images and hypervisors menüben tudjuk megtenni. Fontos kiemelni a GNS3 szempontjából, hogy nem csak IOS-t támogat, hanem a Juniper által fejlesztett JunOS is beállítható, így akár nem csak homogén eszközparkból felépített hálózatot tervezhetők. Ehhez a lépéshez le kell tölteni egy az Interneten is szabadon elérhető IOS-t.

Megfelelő IOS-t Cisco C7200 VXR-hez a

<ftp://ftp.unikon-ua.net/./../pub/Cisco/IOS/7200/c7200-advipservicesk9-mz.122-33.SRE6.bin> oldalon található, így a további munka ezzel történik.

A következő lépés a beállítás, ezt az IOS and hypervisor ablakban az IOS Images fülön a settings alatt lehet megtenni, ahogy az a 2.2-es ábrán látható. Image fájlnál ki kell választani a letöltött image fájlt. A Base config-nál érdemes a GNS3 által kreált alap config fájlból többet is létrehozni és átnevezni a fájlt, arra a típusra amilyen router-hez kerül beállításra. A Base config tartalma általános beállításokat tartalmaz, mint a host neve.



2.2. ábra IOS image-k beállítása

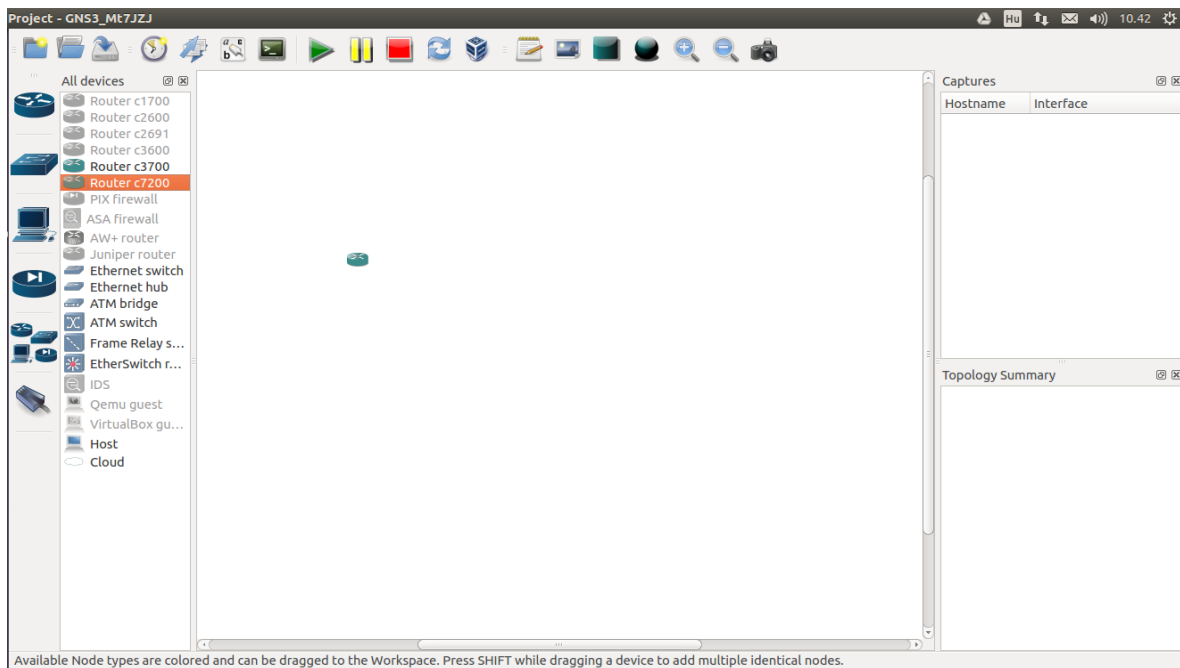
Ez az alábbi módon néz ki:

```
!  
hostname %h  
!  
no ip domain lookup  
no ip icmp rate-limit unreachable  
ip tcp synwait 5  
!  
line con 0  
exec-timeout 0 0
```

```
logging synchronous
privilege level 15
no login
line aux 0
exec-timeout 0 0
logging synchronous
privilege level 15
no login
!
!
end
```

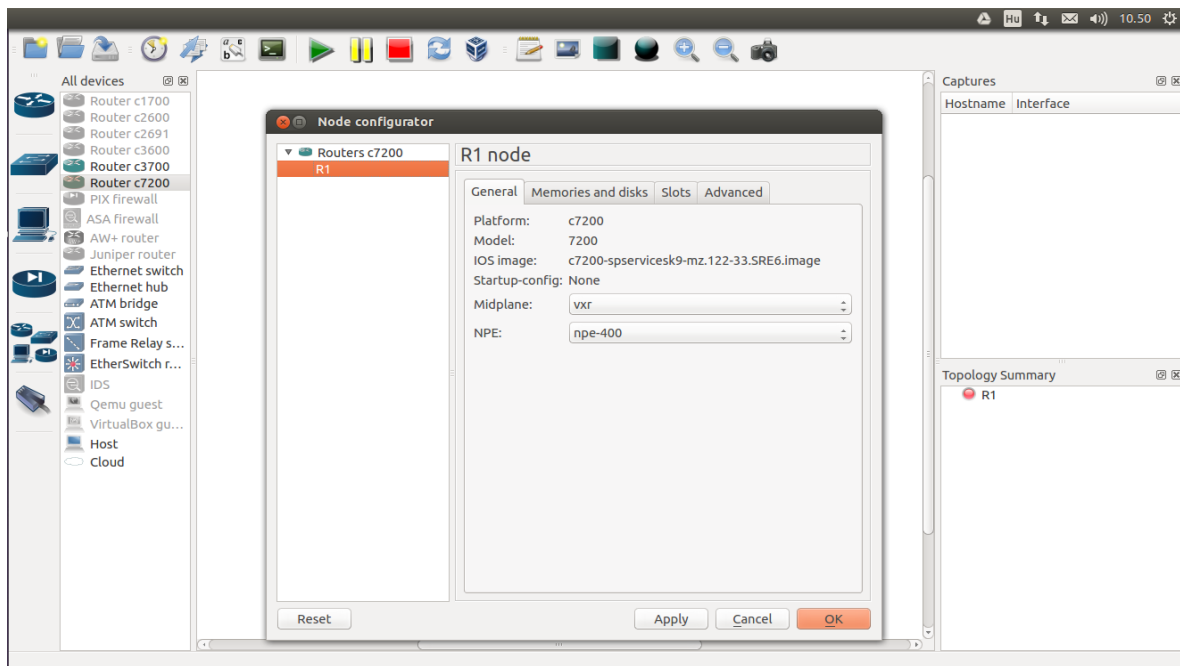
Ez után ki kell választani a megfelelő platform-ot, ebben az esetben ez a c7200-es lesz, és a modellt a szerint, hogy milyen feature-eket szükséges használni.

Az IDLE PC értéke megadja azt a memória címet, amit az eszköz lefoglal magának a futáshoz, ez kézzel is szerkeszthető ám ez nem javasolt, az Auto Calculation gomb segítségével automatikusan megtörténik. A Default RAM-hoz beállított érték lesz az, amit az eszköz maximálisan igénybe vehet a futtató gép memóriájából. Ezen a paraméterek megadása a legfontosabb ahhoz, hogy elkezdődhessen a munka a programban. Ezután a Test Setting gombbal lehet tesztelni a beállításokat, majd a Save gombbal elmenthető. Van lehetőség több eszköz hozzáadására, ebben az esetben mindegyiket külön-külön el kell menteni. Egy router típushoz több IOS image is használható, ilyenkor a szoftver figyelmeztet a szimuláció indulásakor és megadja a lehetőséget arra, hogy melyik image-t használjuk. További lépés az eszköz hozzáadása a munkaterületre. Egyszerűen rá kell húzni a munkaterületre a kiválasztott eszközt, amint az a 2.3-as ábrán látható.



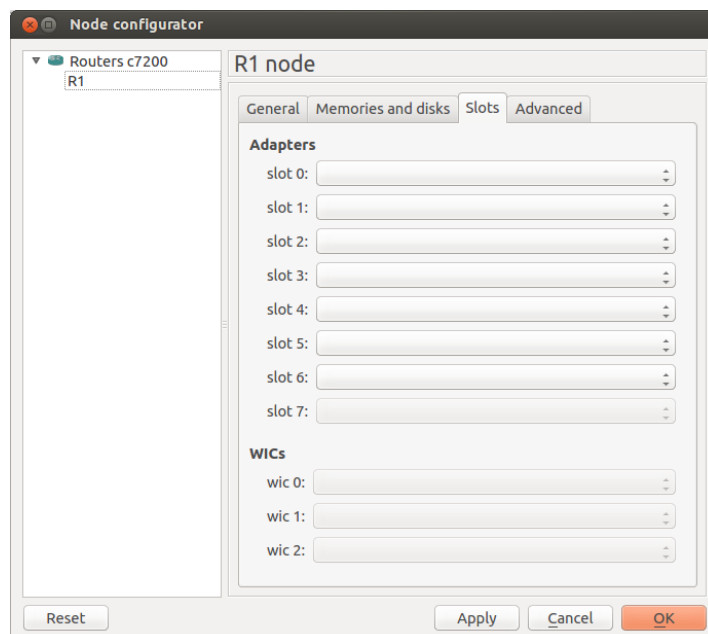
2.3. ábra Eszköz elhelyezése munkaterületen

Nyilván egy hálózathoz több eszközre van szükség, annak függvényében, hogy milyen hálózat kerül tervezésre, annyi eszközt kell hozzáadni a munkaterülethez. Ezeknek az eszközöknek első lépésben a default hostname-t kell beállítani R1, R2, R3, attól függően, hogy hány db eszköz lett elhelyezve a munkaterületen. Természetesen a hostname megváltoztatható bármire, egyedül arra kell figyelni, hogy két eszköznek nem lehet ugyanazt a host nevet beállítani. A szimuláció megkezdése előtt még van lehetőség az eszköz egyedi beállítására, ami az eszközre jobb egérgombbal kattintva a Configure menüpontban érhető el, amit a 2.4-es ábrán láthatunk.



2.4. ábra Router beállítások

A General fülön az eszköz általános paramétereit lehet megváltoztatni. Mivel a dolgozatban bemutatott hálózat tervezésénél nem kellett változtatni az itt lévő paraméteren ezért ez most nem kerül bemutatásra. Továbbá ez helyzet a Memories and disks és Advanced fülrel is. Ami a tervezésre került MPLS hálózat szempontjából fontos, az a Slots fülön található beállítások.



2.5. ábra Slot-ok hozzáadása

Attól függően, hogy milyen platform kerül a hálózatba, annyi slot áll rendelkezésre, ahányat az eszköz támogat. C7200-es ben 7 slot áll rendelkezésre. A slot0-ban csak négy modul típus állítható be, amik a következők lehetnek:

- C7200-IO-FE (1 FastEthernet port)
- C7200-IO-2FE (2 FastEthernet ports)
- C7200-IO-GE-E (1 GigabitEthernet port, Ethernet port is not functional)
- C7200-IO-JC-PA (Cisco 7200 Series Port Adapter Jacket Card)

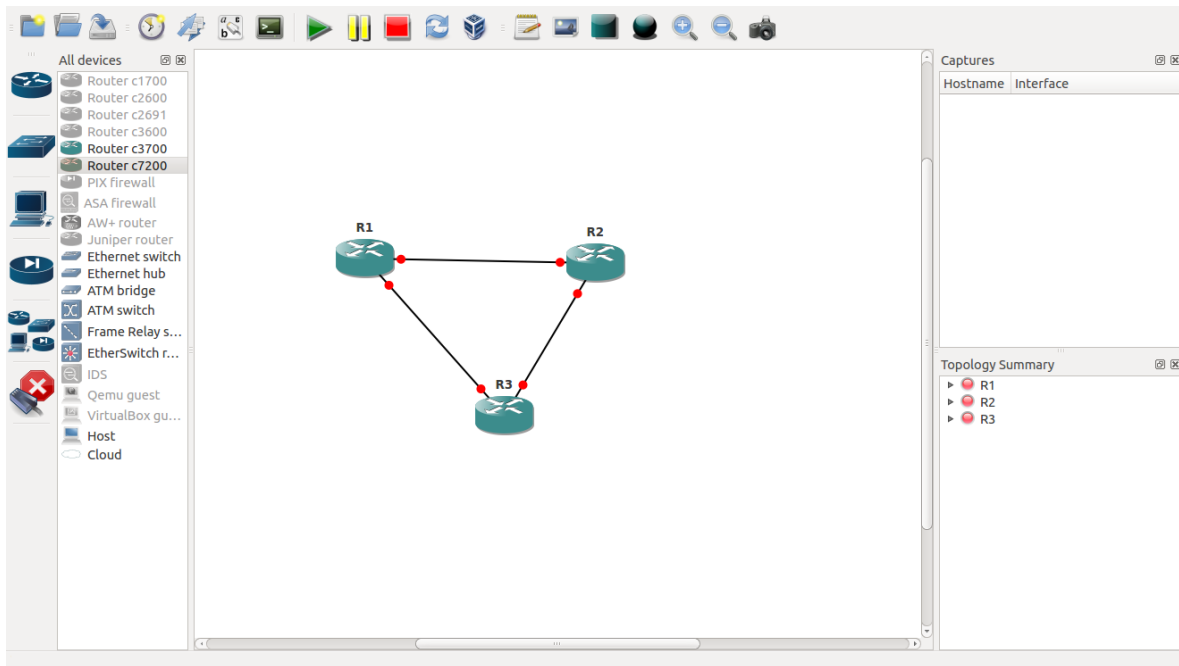
A slot1-től slot7-ig a következő portok állíthatók be:

- PA-FE-TX (1 FastEthernet port)
- PA-2FE-TX (2 FastEthernet ports)
- PA-4E (4 Ethernet ports)
- PA-8E (8 Ethernet ports)
- PA-4T+ (4 serial ports)
- PA-8T (8 serial ports)
- PA-A1 (1 ATM port)
- PA-POS-OC3 (1 Packet-Over-SONET port)
- PA-GE (1 GigabitEthernet port)

A bemutatásra kerülő hálózatban C7200-IO-GE-E és a PA-GE modulokat használtam, ugyanis a modern gerinchálózatok Ethernet kompatibilisek, Carrier Ethernet és Metro Ethernet hálózatok miatt.

Miután megtörtént az eszközök egyedi beállítása, következhet az eszközök összekötése. Fontos, hogy legalább annyi slot kerüljön beállításra (GigabitEthernet esetén 1 modul 1 port), ahány összeköttetést ki kell alakítani.

Az eszközök összekötését a bal sávban található Add a link ikon segítségével lehet megtenni.

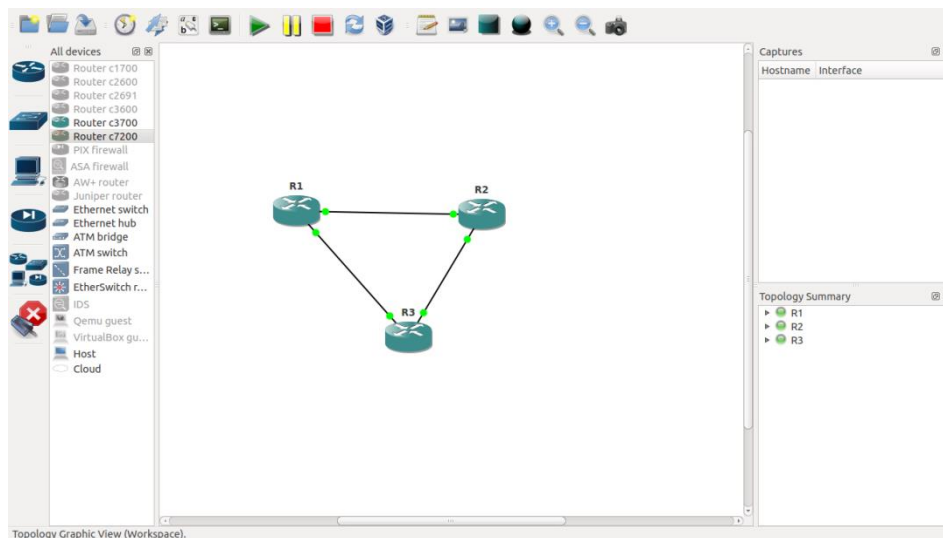


2.6. ábra Összeköttetés kialakítása

Majd eszközre kattintva ki lehet választani az interface-t, ami összeköthető a másik eszköz azonos típusú interface-vel. A szimuláció indításához a felső sorban található

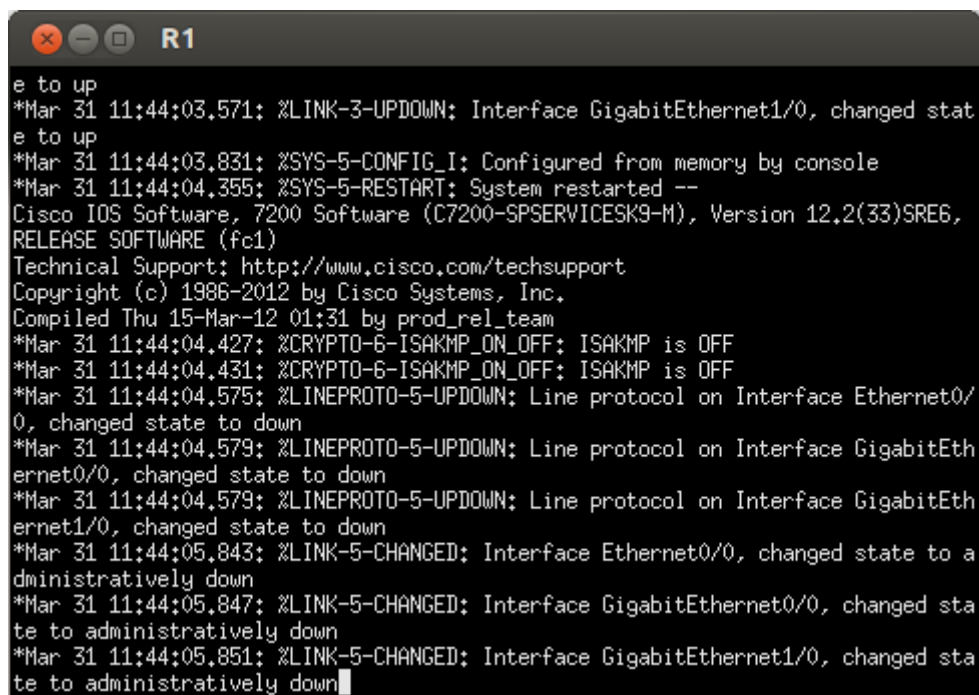
Start/Resume all devices gombra kell kattintani.

Ha elindult a szimuláció, akkor az összes "Node" (2.7-es ábra) zöld státuszt kap és megkezdődhet az egyes eszközök konfigurálása.



2.7. ábra Node státuszok

Az eszközök CLI konfigurálása a jobb egér, Console menüvel elérhető. Innentől szabadon állíthatjuk az eszközöket annak megfelelően, hogy milyen hálózatot akarunk szimulálni.

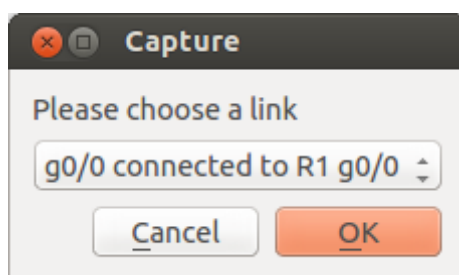


```
e to up
*Mar 31 11:44:03.571: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed stat
e to up
*Mar 31 11:44:03.831: %SYS-5-CONFIG_I: Configured from memory by console
*Mar 31 11:44:04.355: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-SPSERVICESK9-M), Version 12.2(33)SRE6,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 15-Mar-12 01:31 by prod_rel_team
*Mar 31 11:44:04.427: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Mar 31 11:44:04.431: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Mar 31 11:44:04.575: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
0, changed state to down
*Mar 31 11:44:04.579: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet0/0, changed state to down
*Mar 31 11:44:04.579: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet1/0, changed state to down
*Mar 31 11:44:05.843: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to a
dministratively down
*Mar 31 11:44:05.847: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed sta
te to administratively down
*Mar 31 11:44:05.851: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed sta
te to administratively down
```

2.8. ábra CLI felépítése

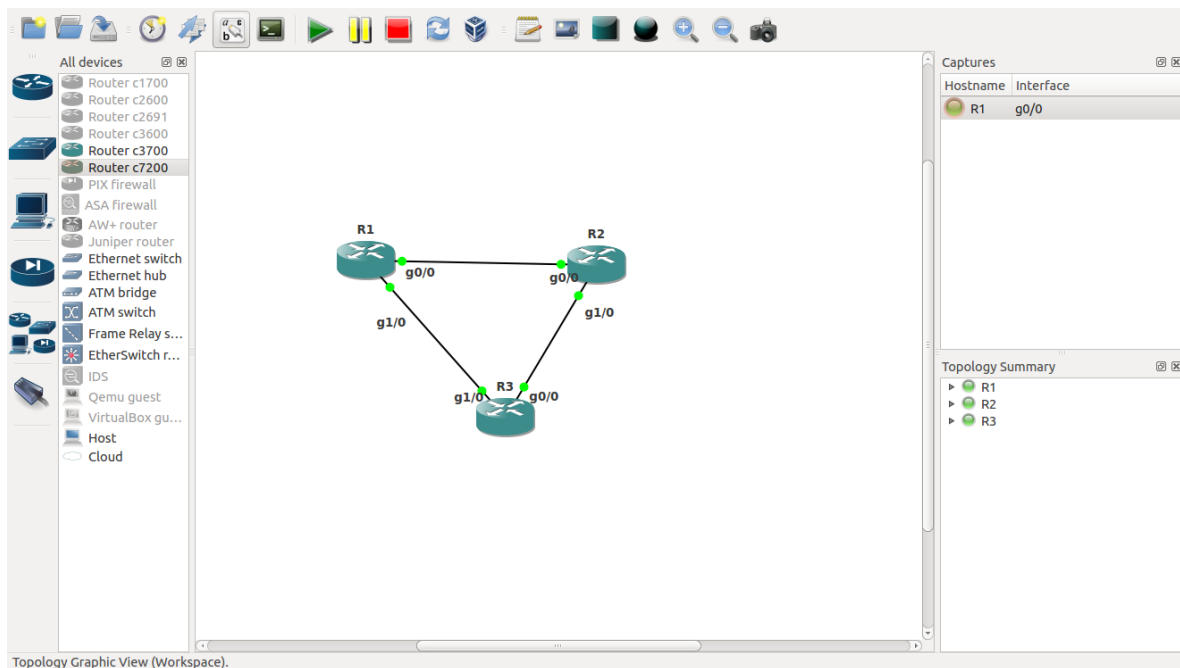
A 2.8-as ábrán láthatjuk a real-time CLI szimulációt, mivel eszközök éppen most lettek indítva, a default konfiguráció fut le az R1-es router-en. Ennek a legnagyobb előnye, hogy a használt IOS parancsok, azonnal kiértékelésre kerülnek, így real-time lehet tesztelni a hálózatot.

A hálózat analizálását támogató Wireshark eléréséhez ki kell választanunk az analizálni kívánt összeköttetés egyik interface-ét, amit az eszközre jobb egérgombbal kattintva a Capture menüben tudunk megtenni. A 2.9-es látható, hogy ezután kiválasztható a megfelelő interface.



2.9. ábra Interface szelekció

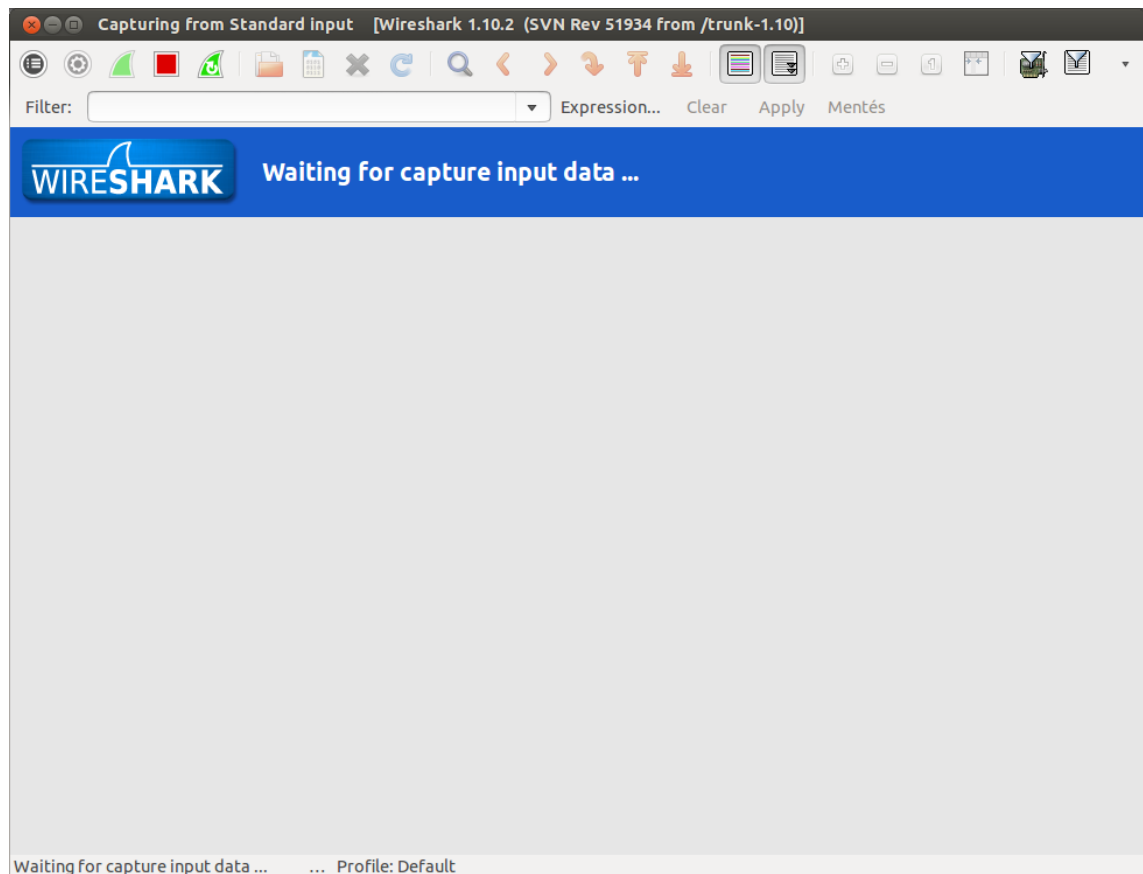
Például R1 és R2 közti kapcsolatot kell analizálni, akkor azt az R1-es interface-t kell kiválasztani ami R2 felé "néz". Ezt megkönnyíthető azzal, ha bekapsolásra kerül az interface címke, ami a felső sorban az Show/Hide interface label gombbal elérhető (5.10-es ábra).



2.10. ábra Interface feliratok

Ha meg van megfelelő interface, akkor a Captures részen megjelenik az analizálni kívánt összeköttetés kiválasztott interface-e. Ebben a részben lehet beállítani, eltávolítani, hogy éppen melyik interface csomagjait "kapja el" a csomag analízátor. Az analízis, - ami szintén real-time - az interface-re jobb egérgombbal kattintva a Start Wireshark menüpontban érhető el. Továbbá itt érhető el a korábban említett többi lehetőség is.

Addig, amíg az összeköttetésen nincs hálózati forgalom, a Wireshark várja a beérkező adatokat. Amint megtörténik az eszközök konfigurálása, elindul az adatforgalom az összeköttetésen. A továbbiakban a Wireshark nem kerül bemutatásra, csak a hálózat analíziséhez lesz használva.

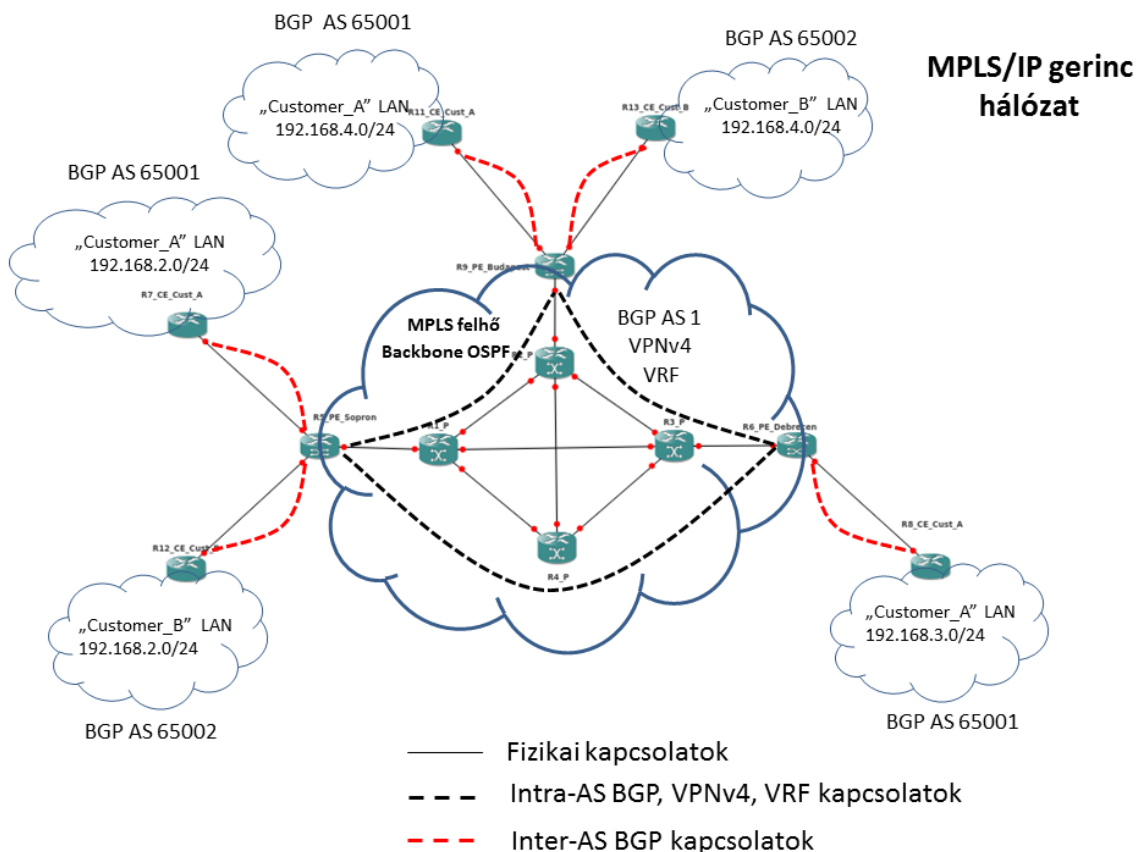


2.11. ábra Wireshark

A fejezet további részében egy példahálózaton keresztül bemutatom az MPLS eljárás működését, a hálózat felépítését, a tervezés lépéseinek, egyes elemeit.

2.2. MPLS példahálózat tervezése és implementációja GNS3-ban

A GNS3 beállítása és konfigurálása után, a most következő fejezet részben bemutatásra kerül egy példahálózat összeállítása lépésről-lépésre. A tervezett hálózat képét a 2.12-es ábrán láthatjuk.



2.12. ábra Tervezésre szánt példahálózat

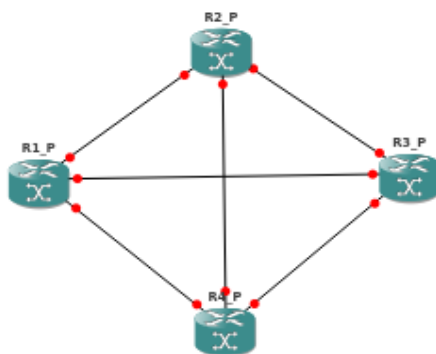
A hálózat felépítése szerint alapvetően négy nagy csoportba osztható:

- MPLS gerinc magjának kialakítása, OSPF rotuing kialakítása LSR-k, létrehozása,
- LER-ek hozzáadása, intra-AS Multiprotocol BGP kapcsolatok kialakítása,
- VPN hozzáadása gerinc hálózathoz, PE router-ek kialakítása,
- Customer Edge eszközök csatlakoztatása, PE eszközökben VRF-ek létrehozása, iner-AS BGP kapcsolatok beállítása, ügyfelek telephelyeinek csatlakoztatása.

2.3. MPLS gerinc magja, LSR-ek létrehozása

Az MPLS gerinc kialakításánál, több szempontot is figyelembe kell venni. Az egyik ilyen szempont, hogy milyen Layer2 protokoll kerül kiválasztásra a hálózathoz. Mivel GigabitEthernet-en lesz az adatkapcsolat, ezért különösebben nem érdekes. A következő szempont a hálózati topológia megválasztása. Ebben, példahálózatban a full-mesh topológiára került a választás, ami megfelel a gerinchálózati paramétereknek, tartalmaz elegendő hibavédelmet, továbbá jól szemléltethető az MPLS címkekapcsolás lényege és az OSPF előnye. Ehhez első lépésben négy darab eszközre van szükség. Ezek alkotják a gerinchálózatunk magját. A router-ek OSPF-et és MPLS-t is futtatnak egyszerre. Az OSPF-re azért van szükség, hogy az LDP TCP felett létre tudja hozni, az MPLS címke összekapcsolási táblákat az egyes router-ekben. Onnantól, hogy ez megtörtént, ezek az eszközök már LSR-ként üzemelnek.

Tehát szükség van négy router-re, ezeket a munkaterületen kell elhelyezni.



2.13. ábra MPLS mag eszközök

A hálózati elemek hostname-k megválasztásánál az MPLS/VPN terminológia lett alkalmazva, tehát nem LSR-k, hanem P router-ek vannak az ábrában és a konfigurációban. Első lépésben, minden router kapcsolatát meg kell adni, tehát minden router-nek be kell lépni a CLI-jébe, és be- kell állítani az egyes interfészek IP címét. Ehhez érdemes /30-as hálózati címeket használni ugyanis minden összeköttetés egy-egy pont-pont kapcsolatot ad, aminek két végpontja van. Az interface konfiguráció az alábbi módon néz ki:

R1_P:

!

```
interface GigabitEthernet1/0
description >>> to R2_P interface <<<
ip address 10.10.10.5 255.255.255.252
```

```

!  

interface GigabitEthernet2/0  

description >>> to R4 interface <<<  

ip address 10.10.10.13 255.255.255.252  

!  

interface GigabitEthernet3/0  

description >>> to R3 interface <<<  

ip address 10.10.10.9 255.255.255.252

```

Látható, hogy R1_P router-nél felhasználásra került R2_P felé a 10.10.10.4/30-as, R3_P felé a 10.10.10.8/30-as és R4_P felé a 10.10.10.12/30-as hálózat. Ezt a konfigurációt kell kialakítani minden routeren, természetesen figyelembe véve a már felhasznált IP tartományokat.

R2_P esetén, az alábbi módon néz ki:

```

!  

interface GigabitEthernet0/0  

description >>> to R3 interface <<<  

ip address 10.10.10.17 255.255.255.252  

!  

interface GigabitEthernet1/0  

description >>> to R1 interface <<<  

ip address 10.10.10.6 255.255.255.252  

!  

interface GigabitEthernet2/0  

description >>> to R4 interface <<<  

ip address 10.10.10.21 255.255.255.252

```

R3_P esetén:

```

!  

interface GigabitEthernet1/0  

description >>> to R2 interface <<<  

ip address 10.10.10.18 255.255.255.252  

!  

interface GigabitEthernet2/0  

description >>> to R4 interface <<<  

ip address 10.10.10.25 255.255.255.252  

!  

interface GigabitEthernet3/0  

description >>> to R1 interface <<<  

ip address 10.10.10.10 255.255.255.252

```

R4_P esetén:

```
!  
interface GigabitEthernet0/0  
description >>> to R3 interface <<<  
ip address 10.10.10.26 255.255.255.252  
!  
interface GigabitEthernet1/0  
description >>> to R1 interface <<<  
ip address 10.10.10.14 255.255.255.252  
!  
interface GigabitEthernet2/0  
description >>> to R2 interface <<<  
ip address 10.10.10.22 255.255.255.252
```

Az IOS-ben a konfigurációhoz a *conf t* paranccsal lehet belépni. Az interface-ket pedig az *interface gx/x* paranccsal lehet elérni. A *des* paranccsal lehet az interface-nek leírást adni, ami szabadon megválasztható. Az *ip address* segítségével tudjuk megadni a kívánt hálózatunkat a 255.255.255.252-es maszk jelöli a /30-as pont-pont hálózatot.

Alapértelmezetten a C7200-esknél az interfészek lekapcsolt *down* státuszt kapnak ezért mindegyiket be kell kapcsolni a konfigurációjuk végén. Ezt a *no shutdown* paranccsal tudjuk megtenni. Ezen kívül minden forgalomirányítóhoz definiálni kell egy loopback interface-t, amit az *int loX* paranccsal tudunk létrehozni, ahol az X tetszőleges decimális valós szám lehet. Ennek az interfésznek is kell adni hálózat címet, ami ebben az esetben egy tetszőleges /32-es IP cím. Az IOS támogatja tabulátor funkciót, tehát a parancsokat nem kell végig írni az értelmezhetőséghez. Ezért a dolgozatban sok helyen csak rövidítve lesznek leírva.

A következő lépés a routing beállítása. A példahálózatban OSPF routing alkalmazása a célszerű, ugyanis ez támogatja az MPLS Traffic Engineering funkciókat. A routing beállítása szintén konfigurációs üzemmódban (*conf t*) a router parancs segítségével történik. Ez az alábbi módon néz ki az egyes forgalomirányítóknál:

R1_P esetén:

```
!  
router ospf 1  
log-adjacency-changes  
network 1.1.1.1 0.0.0.0 area 0  
network 10.10.10.4 0.0.0.3 area 0  
network 10.10.10.8 0.0.0.3 area 0  
network 10.10.10.12 0.0.0.3 area 0  
network 10.10.10.28 0.0.0.3 area 0  
!
```

R2_P esetén:

```
!  
router ospf 1  
log-adjacency-changes  
network 2.2.2.2 0.0.0.0 area 0  
network 10.10.10.4 0.0.0.3 area 0  
network 10.10.10.16 0.0.0.3 area 0  
network 10.10.10.20 0.0.0.3 area 0  
!
```

R3_P esetén

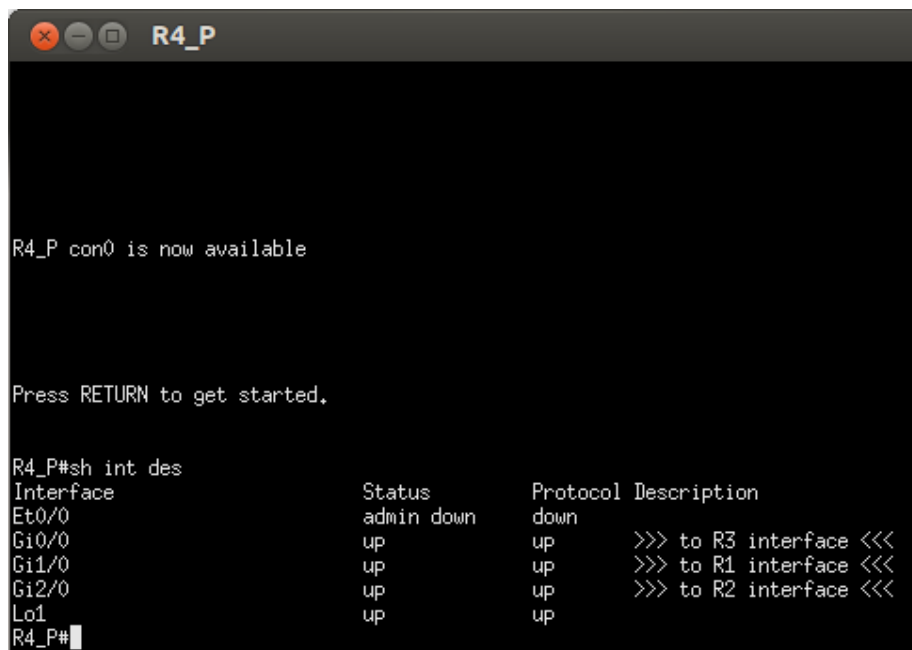
```
!  
router ospf 1  
log-adjacency-changes  
network 3.3.3.3 0.0.0.0 area 0  
network 10.10.10.8 0.0.0.3 area 0  
network 10.10.10.16 0.0.0.3 area 0  
network 10.10.10.24 0.0.0.3 area 0  
network 10.10.10.32 0.0.0.3 area 0  
!
```

R4_P esetén:

```
!  
router ospf 1  
log-adjacency-changes  
network 4.4.4.4 0.0.0.0 area 0  
network 10.10.10.12 0.0.0.3 area 0  
network 10.10.10.20 0.0.0.3 area 0  
network 10.10.10.24 0.0.0.3 area 0  
!
```

A routing beállításánál nem konkrét interfész címek kerülnek megadásra, hanem hálózati címek, amit a router hirdet a hálózaton. A hálózati címek után jön a kiegészítő maszk vagy más néven *wildcard*, aminek értéke 255.255.255.255 - x.x.x.x attól függően milyen hálózati maszkot használtunk. Ez biztosítja a változó hosszúságú alhálózatok használatát a protokollban. Az area 0 jelöli, hogy backbone hálózat lesz kialakítva, az OSPF ehhez mértén fogja kezelni a kapcsolatokat. Ezek a beállítások tesztelhetők az alábbi parancsokkal:

sh int des



```
R4_P con0 is now available

Press RETURN to get started.

R4_P#sh int des
Interface          Status      Protocol Description
Et0/0              admin down  down
Gi0/0              up          up      >>> to R3 interface <<<
Gi1/0              up          up      >>> to R1 interface <<<
Gi2/0              up          up      >>> to R2 interface <<<
Lo1                up          up
R4_P#
```

2.14. ábra sh int des kimenete

sh ip route

```
R1_P

R1_P con0 is now available

Press RETURN to get started.

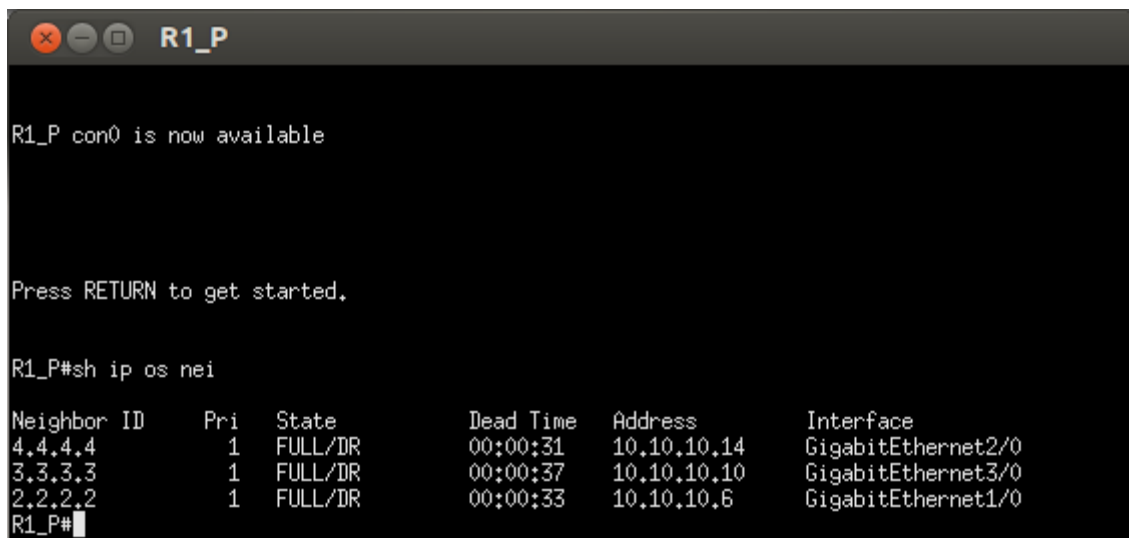
R1_P#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP
        + - replicated route, % - next hop override

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback1
    2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 10.10.10.6, 00:00:32, GigabitEthernet1/0
    3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/2] via 10.10.10.10, 00:00:42, GigabitEthernet3/0
    4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/2] via 10.10.10.14, 00:00:42, GigabitEthernet2/0
   10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
C       10.10.10.4/30 is directly connected, GigabitEthernet1/0
L       10.10.10.5/32 is directly connected, GigabitEthernet1/0
C       10.10.10.8/30 is directly connected, GigabitEthernet3/0
L       10.10.10.9/32 is directly connected, GigabitEthernet3/0
C       10.10.10.12/30 is directly connected, GigabitEthernet2/0
L       10.10.10.13/32 is directly connected, GigabitEthernet2/0
O       10.10.10.16/30 [110/2] via 10.10.10.10, 00:00:42, GigabitEthernet3/0
O       10.10.10.20/30 [110/2] via 10.10.10.14, 00:00:42, GigabitEthernet2/0
        [110/2] via 10.10.10.6, 00:00:32, GigabitEthernet1/0
O       10.10.10.24/30 [110/2] via 10.10.10.14, 00:00:42, GigabitEthernet2/0
        [110/2] via 10.10.10.10, 00:00:42, GigabitEthernet3/0
R1_P#
```

2.15. ábra sh ip route kimenete

Valamint az *sh ip os nei* szolgál erre a feladatra.



```
R1_P con0 is now available

Press RETURN to get started.

R1_P#sh ip os nei
Neighbor ID    Pri   State       Dead Time   Address        Interface
4.4.4.4        1     FULL/DR     00:00:31    10.10.10.14    GigabitEthernet2/0
3.3.3.3        1     FULL/DR     00:00:37    10.10.10.10    GigabitEthernet3/0
2.2.2.2        1     FULL/DR     00:00:33    10.10.10.6     GigabitEthernet1/0
R1_P#
```

2.16. ábra *sh ip os nei* kimenete

Az első paranccsal, csak az interfészek állapota ellenőrizhető, a beállított IP címek nem. Ha az interface konfigurációját kell ellenőrizni, akkor a *sh run int <interface száma>* paranccsal lehet megtenni. A második parancs a route tábla ellenőrzésére szolgál, meg lehet nézni, hogy az eszköz milyen hálózatokat lát. A harmadik parancs adja meg az OSPF szomszédjait, itt már csak a loopback interfészek hálózat címeit látja, az interfész nevét és azt, hogy milyen IP címen keresztül éri el az eszközt. Az eddigi beállítások definiálják a mag router-eken az OSPF hálózatot, ami az MPLS és az LDP számára szükséges a címke összerendelési táblák kialakításához az LSR-ekben.

Az MPLS futtatásához minden router, minden interfésze alá a következő két parancsot kell megadni:

mpls ip

mpls label protocol ldp

Ezt az összes eszközbe belépve az *int range* parancs segítségével egyszerre minden interface-be be lehet állítani. Ez a lépés roppant fontos ahhoz, hogy a hálózatunk MPLS-t használjon. Az eredményt le lehet ellenőrizni az *sh mpls forwarding-table* parancs segítségével.

```

R1_P#sh mpls forwarding-table
Local   Outgoing Prefix      Bytes Label  Outgoing  Next Hop
Label   Label    or Tunnel Id Switched     interface
16      Pop Label 3.3.3.3/32   0           Gi3/0      10.10.10.10
18      Pop Label 10.10.10.24/30 0           Gi3/0      10.10.10.10
        Pop Label 10.10.10.24/30 0           Gi2/0      10.10.10.14
19      Pop Label 10.10.10.16/30 0           Gi3/0      10.10.10.10
20      Pop Label 4.4.4.4/32   0           Gi2/0      10.10.10.14
21      Pop Label 2.2.2.2/32   0           Gi1/0      10.10.10.6
22      Pop Label 10.10.10.20/30 0           Gi1/0      10.10.10.6
        Pop Label 10.10.10.20/30 0           Gi2/0      10.10.10.14
R1_P#

```

2.17. ábra Címkekapcsolási tábla

Ez tulajdonképpen a címke összerendelési tábla, ami tartalmazza a kimenő és a lokális címkét, a hozzátartozó hálózati címet és interfészt, valamint a következő ugrási címet. Továbbá le lehet még ellenőrizni az MPLS kapcsolatot a LSP ping segítségével, ami az alábbi módon néz ki:

ping mpls ipv4 <IP cím> <alhálózati maszk>

```

R1_P#ping mpls ipv4 4.4.4.4 255.255.255.255
Sending 5, 100-byte MPLS Echos to 4.4.4.4/32,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
       'L' - labeled output interface, 'B' - unlabeled output interface,
       'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
       'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
       'P' - no rx intf label prot, 'p' - premature termination of LSP,
       'R' - transit router, 'I' - unknown upstream index,
       'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/21/36 ms
R1_P#

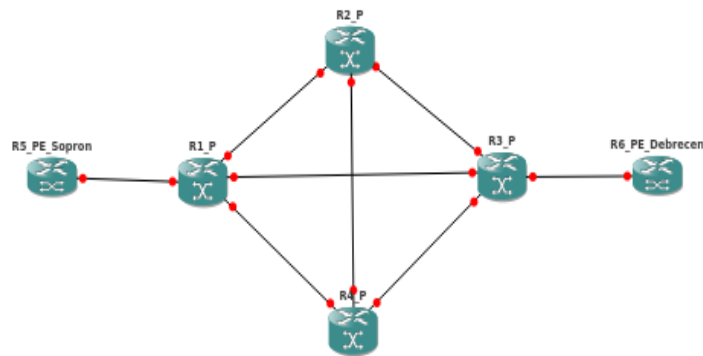
```

2.18. ábra Ping parancs kimenete

Ha megfelelőek a hálózat beállításai, akkor a LSP ping sikeres lesz. Ezzel be is fejeződött az első fázis, el lehet kezdeni a LER-ek hozzáadását a jelenlegi hálózathoz.

2.4. Edge-LSR-ek, LER-ek hozzáadása

Ahhoz, MPLS hálózat magja használható legyen átvitelre, szükség van belépő eszközökre. Ezek az MPLS terminológiában LER-ek vagy E-LSR-k. Mivel a LER-eknek "ismernie" kell egymást a hálózaton ezért érdemes inter-AS routing-ot alkalmazni. Erre a célra a Multiprotocol BGP szolgál, ami tulajdonképpen hagyományos BGP router ami, MPLS-t is futtat. Az MPLS hálózat felé pedig OSPF-fel hirdeti a hálózatait. A hálózat képe az alábbi módon változik:



2.19. ábra MPLS hálózat LER-ekkel kiegészítve

Könnyen belátható, hogy a LER-ek konfigurációja az MPLS hálózat felé nem tartogat meglepetéseket. A hálózathoz kifelé néző interfészen, illetve a routing-ban található különbség.

```

interface R5_PE_Sopron:
!
interface GigabitEthernet0/0
description >>> to R1 interface <<<
ip address 10.10.10.29 255.255.255.252
mpls ip
mpls label protocol ldp
!
interface GigabitEthernet1/0
description >>> to CE interface <<<
ip address 192.168.1.1 255.255.255.252
interface R6_PE_Debrecen:
!
interface R6_PE_Debrecen
!
interface GigabitEthernet0/0
description >>> to R3 interface <<<
ip address 10.10.10.34 255.255.255.252
mpls ip
mpls label protocol ldp
!
interface GigabitEthernet1/0
description >>> to CE interface <<<
ip address 192.168.1.5 255.255.255.252
!

```

A CE interfészekhez nem kell MPLS-t beállítani, hiszen a gerincen túl már nem címkekapcsolt a hálózat.

A routing az alábbi módon néz ki:

```

R5_PE_Sopron:
!
router ospf 1
log-adjacency-changes
network 5.5.5.5 0.0.0.0 area 0
network 10.10.10.28 0.0.0.3 area 0
!
router bgp 1
no synchronization
bgp log-neighbor-changes
neighbor 6.6.6.6 remote-as 1
no auto-summary
!

```

R6_PE_Debrecen:

!

router ospf 1

log-adjacency-changes

network 6.6.6.6 0.0.0.0 area 0

network 10.10.10.32 0.0.0.3 area 0

!

router bgp 1

no synchronization

bgp log-neighbor-changes

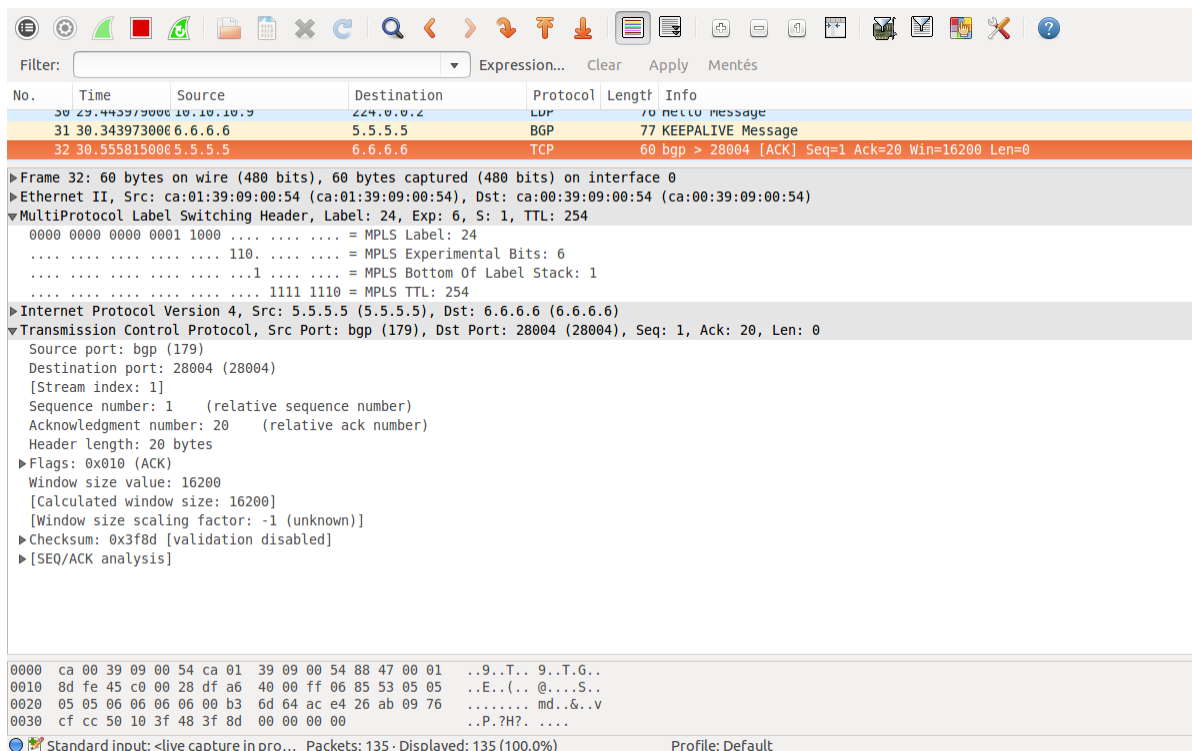
neighbor 5.5.5.5 remote-as 1

neighbor 5.5.5.5 update-source Loopback1

no auto-summary

!

A BGP routing-ban ebben az esetben, nem hálózatok lettek meghirdetve, hanem az MPLS felhőben lévő más BGP routerek loopback címeit kellett megadni. Ezek után a BGP az MPLS-en belül TCP protokoll felett "megkeresi" a szomszédjait. Wiresharkban ez az alábbi módon néz ki:



2.19. ábra MPLS hálózaton BGP kapcsolat TCP felett

Látható, a hálózatba belépő TCP csomag címkézve lép be a hálózatba, tehát működik a címkekapcsolás. A címkekapcsolási táblában is lesz változás:

```

R1_P#sh mpls fo
Local   Outgoing   Prefix      Bytes Label   Outgoing   Next Hop
Label   Label      or Tunnel Id Switched      interface
20      Pop Label  3.3.3.3/32  0             Gi3/0      10.10.10.10
21      Pop Label  10.10.10.32/30 0             Gi3/0      10.10.10.10
22      Pop Label  10.10.10.24/30 0             Gi3/0      10.10.10.10
        Pop Label  10.10.10.24/30 0             Gi2/0      10.10.10.14
23      Pop Label  10.10.10.16/30 0             Gi1/0      10.10.10.6
        Pop Label  10.10.10.16/30 0             Gi3/0      10.10.10.10
24      24        6.6.6.6/32  1655          Gi3/0      10.10.10.10
25      Pop Label  5.5.5.5/32  2019          Gi0/0      10.10.10.29
26      Pop Label  4.4.4.4/32  0             Gi2/0      10.10.10.14
27      Pop Label  2.2.2.2/32  0             Gi1/0      10.10.10.6
28      Pop Label  10.10.10.20/30 0             Gi1/0      10.10.10.6
        Pop Label  10.10.10.20/30 0             Gi2/0      10.10.10.14
R1_P#

```

2.20. ábra MPLS címkekapcsolási tábla LER-ekkel

Látható hogy az eddigiekkel szemben a 24-es és 25-ös címkéhez tartozó interfészein – a TCP kommunikációnak köszönhetően megjelent, hogy hány bájt lett továbbítva az adott hálózatok felé. További lépés a VPN kapcsolatok kialakítása a LER-eken amit innentől fogva PE router-nek kell nevezni, illetve az LSR-ket pedig P router-nek.

2.5. VPN hozzáadása MPLS gerinchez, P és PE router-ek kialakítása

A VPN hozzáadása már nem MPLS funkció, a BGP routing-ban kell meghatározni az addressfamily-t, tehát a LER-ben kell beállítani. Ezzel tulajdonképpen azt határozzuk meg, hogy mely BGP router-ek futtassanak VPN-t, valamint ezeknek a kapcsolatát írja le. A routing konfiguráció az alábbi módon változik az előzőekhez képest.

R5_PE_Sopron:

!

router bgp 1

no synchronization

bgp log-neighbor-changes

neighbor 6.6.6.6 remote-as 1

no auto-summary

!

address-family vpnv4

neighbor 6.6.6.6 activate

neighbor 6.6.6.6 send-community extended

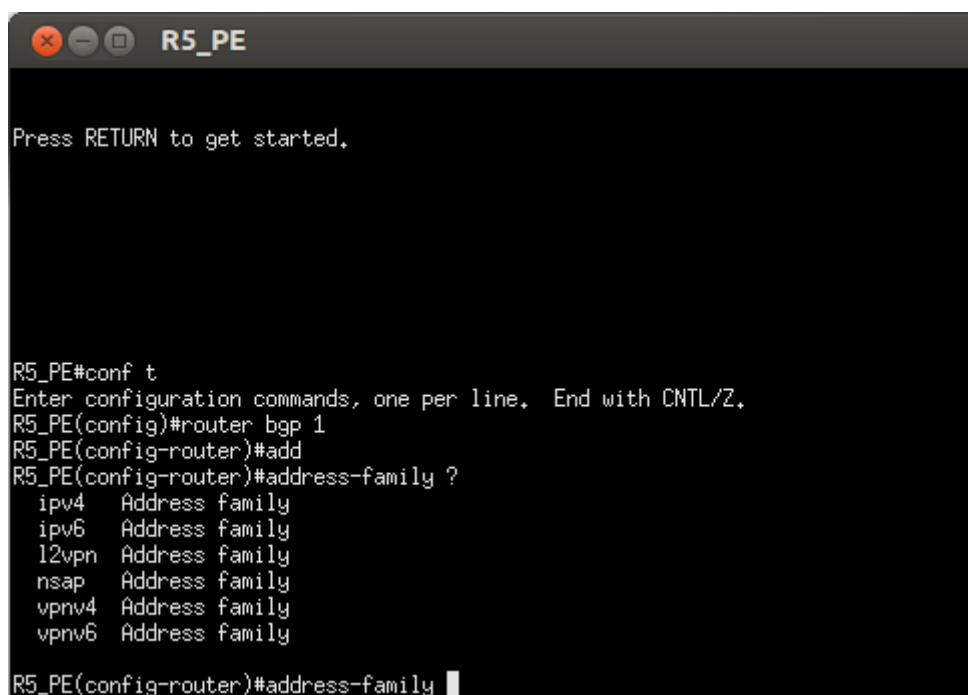
exit-address-family

!

R6_PE_Debrecen:

```
!  
router bgp 1  
no synchronization  
bgp log-neighbor-changes  
neighbor 5.5.5.5 remote-as 1  
neighbor 5.5.5.5 update-source Loopback1  
no auto-summary  
!  
address-family vpnv4  
neighbor 5.5.5.5 activate  
neighbor 5.5.5.5 send-community extended  
exit-address-family  
!
```

Az address-family-nél meghatározhatjuk azt, hogy milyen alkalmazás futtasson BGP routing-ot a globális BGP mellett. Mivel VPN IPV4 kerül alkalmazásra ezért, szükséges kiválasztani a vpnv4-es opciót. Az address-family további lehetőségei:



```
R5_PE  
Press RETURN to get started.  
  
R5_PE#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R5_PE(config)#router bgp 1  
R5_PE(config-router)#add  
R5_PE(config-router)#address-family ?  
  ipv4  Address family  
  ipv6  Address family  
  l2vpn  Address family  
  nsap   Address family  
  vpnv4  Address family  
  vpnv6  Address family  
R5_PE(config-router)#address-family
```

2.21. ábra address-family opciók

Az address-family-n belül szükséges is meghatározni, BGP-peer-eket. Erre szolgál a neighbor parancs. A VPN-t ellenőrizni a *sh bgp vpnv4 unicast all* paranccsal tudjuk. A parancs kimenete a következő:

```

R5_PE

R5_PE con0 is now available

Press RETURN to get started.

R5_PE#sh bgp vpv4 unicast all
BGP table version is 4, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, x best-external
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 1:100 (default for vrf Cust_A)
*> 192.168.2.0     192.168.1.2             0             0 65001 i
*>i192.168.3.0     6.6.6.6                 0      100      0 65001 i
R5_PE#

```

2.22. ábra sh bgp vpv4 unicast all kimenete

Mivel az itt látható kimenetben már VRF-k is szerepelnek, ezért nem minden sor ennek a lépésnek a következménye. További megértéshez szükséges a következő pont, a CE eszközök csatlakoztatása.

2.6. Customer Edge eszközök csatlakoztatása, PE eszközökben VRF-ek létrehozása

Customer Edge eszköz bármilyen router lehet, ami tudja a BGP routing protokollt. A példahálózatban megmaradt a Cisco C7200-es sorozat, így megmarad a Cisco specifikus CLI konfiguráció. Az eszközben célszerű legalább két interface beállítani, egyet a PE oldalán, egyet vagy többet pedig LAN-ra. A CE eszközök konfigurációja az alábbi módon néz ki:

R7_CE_Cust_A:

```

!
interface GigabitEthernet0/0
description >>> to PE interface <<<
ip address 192.168.1.2 255.255.255.252
!
interface GigabitEthernet1/0
ip address 192.168.2.1 255.255.255.0
negotiation auto

```

```

!
router bgp 65001
no synchronization
bgp log-neighbor-changes
network 192.168.2.0
neighbor 192.168.1.1 remote-as 1
no auto-summary
!

```

A konfigurációból jól látszik, hogy a BGP szomszédnak a PE router IP címét kell megadni és az ügyfél LAN hálózatát kell hirdetni BGP routing-gal. A BGP AS száma ebben az esetben privát AS lesz, ami 65000 felett található. A példa a következő: az „A” ügyfélnek két telephelye van, egy Sopronban, egy pedig Debrecenben, és szeretné, ha a két telephely LAN hálózata „látná” egymást. Továbbá később szeretne még egy budapesti telephelyet létesíteni, és ehhez a meglévő MPLS gerinc hálózatot szeretné használni. Továbbá „B” ügyfél is szeretne egy soproni telephely és egy budapesti telephely közt VPN kapcsolatot létre hozni. A két ügyfél konkurens, tehát kicsit sem lenne szerencsés, hogy lássák egymás hálózatait.

A gerinc hálózaton meg kell valósítani:

- a PE router-ekben az ügyfelekhez tartozó VRF létrehozását,
- PE router-ekben a VRF-khez tartozó address-family-k meghatározását,
- a CE eszközök BGP routing konfigurálását.

A PE routerek VRF-jének létrehozása az alábbi módon néz ki:

```

R6_PE_Sopron:
!
ip vrf Cust_A
description Customer_A
rd 1:100
route-target export 1:100
route-target import 1:100
!

```

Ahol az *ip vrf <név>*-el lehet létrehozni a virtuális routing táblát, az *rd* az útvonal-megkülönböztető, különbséget tesz az ügyfelek között, tehát ügyfelenként változik, a *route-target* sorok pedig lehetővé teszik az útvonalak importálását a VRF számára.

Ezen túl be kell állítani a VRF-ben használni kívánt interface-t is, ami az alábbi módon néz ki:

R5_PE_Sopron:

```
!  
interface GigabitEthernet1/0  
description >>> to CE interface <<<  
ip vrf forwarding Cust_A  
ip address 192.168.1.5 255.255.255.252  
!
```

A „normál” interfész beállításától csak a *ip vrf forwarding <név>* különbözik, amivel megadható az interfésznek, hogy melyik VRF-et használja, a beérkező csomagok továbbítására.

Ha megvan a VRF akkor ezt hozzá kell adni a BGP routing-hoz is, ami a következőképpen néz ki:

```
!  
address-family ipv4 vrf Cust_A  
no synchronization  
neighbor 192.168.1.6 remote-as 65001  
neighbor 192.168.1.6 activate  
neighbor 192.168.1.6 as-override  
exit-address-family  
!
```

Ez a BGP 1-es AS globális routing alatt létrehozott, routing konfiguráció. Itt meg kell határozni a CE eszköz címét, és AS számát. Mivel az ügyfél hálózata privát, ezért a 65001-es AS számot kapja. Az as-override attribútum szolgál arra, hogy a 65001-ről beérkező csomagok az 1-es AS-ben legyenek továbbítva, amit a gerincen alkalmazunk, valamint ezzel elkerüljük a routing hurkok kialakulását. Könnyen belátható, hogy ez a BGP-n belül csak a „Cust_A” hálózatra vonatkozó routing lesz.

Az R6_PE_Debrecen következőképpen néz ki:

```
!  
ip vrf Cust_A  
description Customer_A  
rd 1:100  
route-target export 1:100  
route-target import 1:100  
!  
Interface-nél:  
!  
interface GigabitEthernet1/0  
description >>> to CE interface <<<  
ip vrf forwarding Cust_A  
ip address 192.168.1.5 255.255.255.252  
negotiation auto  
!
```

Routing-nál pedig:

!

```
address-family ipv4 vrf Cust_A
no synchronization
neighbor 192.168.1.6 remote-as 65001
neighbor 192.168.1.6 activate
neighbor 192.168.1.6 as-override
exit-address-family
```

!

Mint látható, a különbség csak az interfésznél, az IP címek megadásában van, egyértelmű hogy a debreceni PE-nek a debreceni CE router-t kell ismernie. Fontos, hogy az *ip vrf Cust_A* paraméterei ugyanazok legyenek a más-más telephelyekhez tartozó PE-ken.

A CE eszközök konfigurációja már egyszerű, hiszen a PE oldali interface „alá” csak másik /30-as IP-t kell megadni, a LAN felé pedig olyat, amilyen méretű hálózata szükség van. Ebben az esetben /24-es tartomány lett megadva.

A további telephely hozzáadásához a jelenlegi gerinc hálózatot ki kell egészíteni MPLS szinten egy LER-el, valamint a telephely „közelében” lévő LSR-nek egy új interface-t, hálózatot (pont-pont) kell megadni, amit hirdetni kell OSPF-en és ez alapján az MPLS címkekapcsolási tábla is változik.

A hálózat képe az alábbiak szerint módosul:



2.23.ábra Plusz telephely csatlakoztatása meglévő hálózathoz

Mivel változott a LER-ek száma, ezt a BGP-nek is meg kell adni, fel kell venni az új 1-es AS-hez tartozó BGP eszközt, tehát konfigurációk az alábbiak szerint módosulnak:

R9_PE_Budapest:

```
!  
ip vrf Cust_A  
description Customer_A  
rd 1:100  
route-target export 1:100  
route-target import 1:100  
!  
!  
interface Loopback1  
ip address 7.7.7.7 255.255.255.255  
!  
interface GigabitEthernet0/0  
ip address 10.10.10.42 255.255.255.252  
mpls ip  
mpls label protocol ldp  
!  
interface GigabitEthernet1/0  
ip vrf forwarding Cust_A  
ip address 192.168.1.9 255.255.255.252  
! add  
network 10.10.10.40 0.0.0.3 area 0  
!  
router bgp 1  
no synchronization  
bgp log-neighbor-changes  
neighbor 5.5.5.5 remote-as 1  
neighbor 5.5.5.5 update-source Loopback1  
neighbor 6.6.6.6 remote-as 1  
neighbor 6.6.6.6 update-source Loopback1  
no auto-summary  
!  
address-family vpnv4  
neighbor 5.5.5.5 activate  
neighbor 5.5.5.5 send-community extended  
neighbor 6.6.6.6 activate  
neighbor 6.6.6.6 send-community extended  
exit-address-family  
!  
address-family ipv4 vrf Cust_A  
no synchronization  
neighbor 192.168.1.10 remote-as 65001  
neighbor 192.168.1.10 activate  
neighbor 192.168.1.10 as-override  
exit-address-family  
!  
!  
mpls ldp router-id Loopback1  
!
```

A budapesti PE-ben meg kell határozni a globális BGP-ben és VPN-ben egyaránt a soproni és a debreceni BGP routerek IP címét, illetve létre kell hozni „A” előfizető VRF tábláját és konfigurálni kell „A”-hoz tartozó interfészt is. Továbbá hozzá kell adni az OSPF-hez az MPLS felhőbe tartozó hálózatot is, amit szintén el kell végezni R2_P LSR esetén.

A debreceni és soproni telephelyeken csak kis módosítások szükségesek:

R5_PE_Sopron:

```
!  
router bgp 1  
no synchronization  
bgp log-neighbor-changes  
neighbor 6.6.6.6 remote-as 1  
neighbor 6.6.6.6 update-source Loopback1  
neighbor 7.7.7.7 remote-as 1  
neighbor 7.7.7.7 update-source Loopback1  
no auto-summary  
!  
address-family vpnv4  
neighbor 6.6.6.6 activate  
neighbor 6.6.6.6 send-community extended  
neighbor 7.7.7.7 activate  
neighbor 7.7.7.7 send-community extended  
exit-address-family  
!
```

R6_PE_Debrecen:

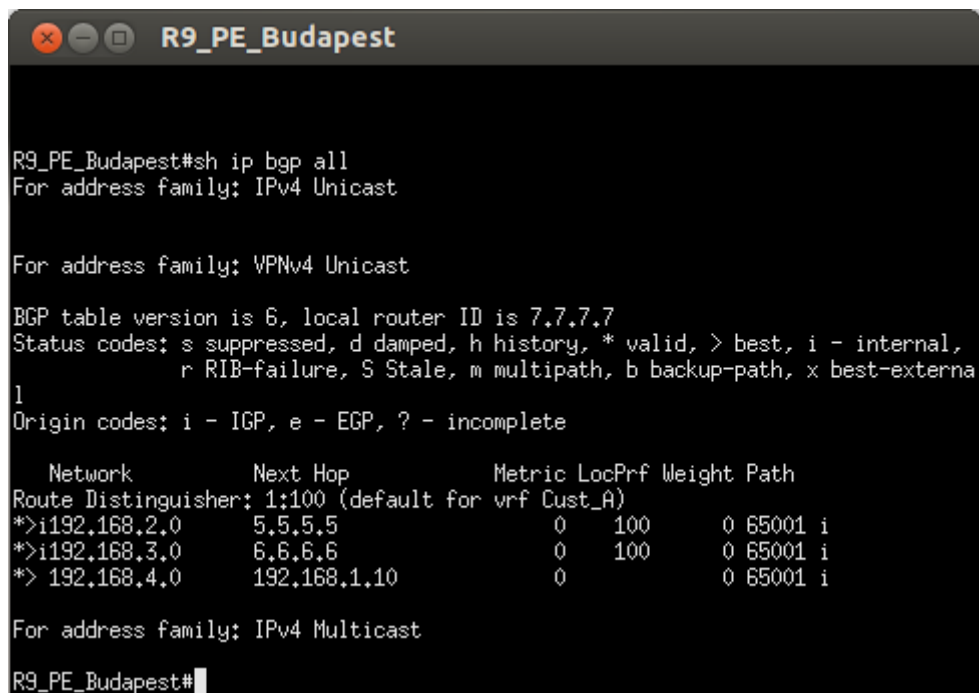
```
!  
router bgp 1  
no synchronization  
bgp log-neighbor-changes  
neighbor 5.5.5.5 remote-as 1  
neighbor 5.5.5.5 update-source Loopback1  
neighbor 7.7.7.7 remote-as 1  
neighbor 7.7.7.7 update-source Loopback1  
no auto-summary  
!  
address-family vpnv4  
neighbor 5.5.5.5 activate  
neighbor 5.5.5.5 send-community extended  
neighbor 7.7.7.7 activate  
neighbor 7.7.7.7 send-community extended  
exit-address-family  
!  
address-family ipv4 vrf Cust_A  
no synchronization  
neighbor 192.168.1.6 remote-as 65001  
neighbor 192.168.1.6 activate  
neighbor 192.168.1.6 as-override  
exit-address-family  
!
```

Az előző CE-hez hasonlóan R11_CE_Cust_A az alábbiak szerint néz ki:

```
!  
interface GigabitEthernet0/0  
ip address 192.168.1.10 255.255.255.252  
media-type gbic  
!  
interface GigabitEthernet1/0  
ip address 192.168.4.1 255.255.255.0  
!  
router bgp 65001  
no synchronization  
bgp log-neighbor-changes  
network 192.168.4.0  
neighbor 192.168.1.9 remote-as 1  
no auto-summary  
!
```

Most következhet a telephelyek közti „átláthatóság” tesztelése. Ehhez a *traceroute <IP cím>*, *sh ip route vrf <vrfazonosító>* és a *sh ip route* és a *sh ip bgp all* parancsok kellenek. Először le kell ellenőrizni, hogy PE oldalon létrejöttek-e a BGP kapcsolatok:

sh ip bgp all



```
R9_PE_Budapest#sh ip bgp all  
For address family: IPv4 Unicast  
  
For address family: VPNv4 Unicast  
  
BGP table version is 6, local router ID is 7.7.7.7  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, x best-external  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
Network          Next Hop          Metric LocPrf Weight Path  
Route Distinguisher: 1:100 (default for vrf Cust_A)  
*>i192.168.2.0    5.5.5.5           0      100      0 65001 i  
*>i192.168.3.0    6.6.6.6           0      100      0 65001 i  
*> 192.168.4.0    192.168.1.10      0              0 65001 i  
  
For address family: IPv4 Multicast  
R9_PE_Budapest#
```

2.24. ábra *sh ip bgp all* kimenete

Mint látható, az összes LAN hálózat megjelenik a táblában. Láthatjuk, hogy ezek a hálózatok azonos AS-be tartoznak, illetve az is kivehető hogy milyen IP-n keresztül érhetők el.

A következő „A” előfizető VRF routing táblájának ellenőrzése:

```
R9_PE_Budapest

R9_PE_Budapest#sh ip route vrf Cust_A

Routing Table: Cust_A
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route, % - next hop override

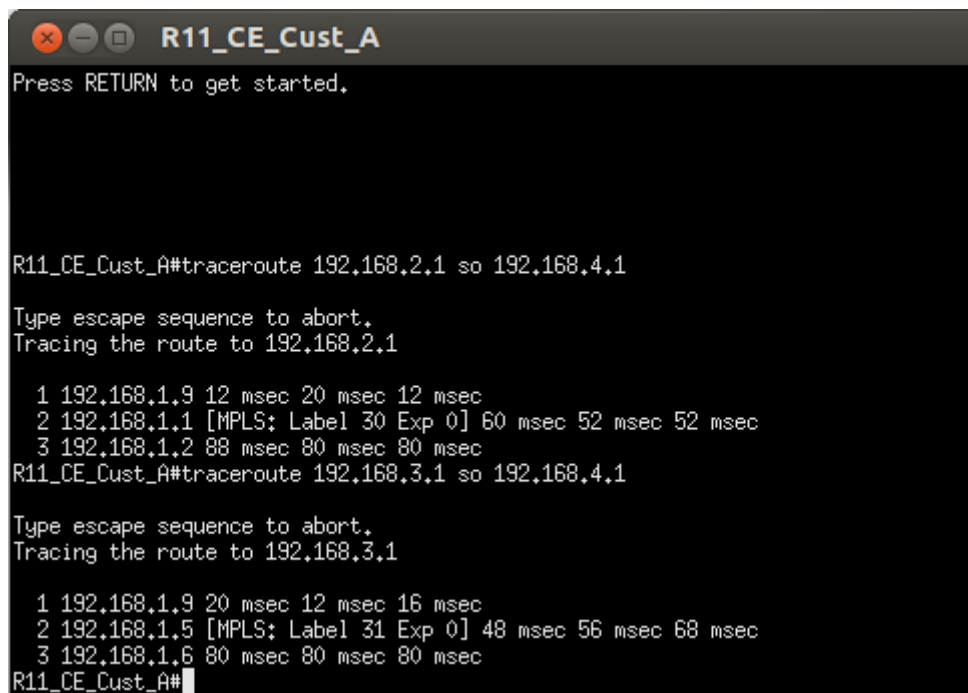
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.8/30 is directly connected, GigabitEthernet1/0
L       192.168.1.9/32 is directly connected, GigabitEthernet1/0
B       192.168.2.0/24 [200/0] via 5.5.5.5, 00:15:32
B       192.168.3.0/24 [200/0] via 6.6.6.6, 00:15:32
B       192.168.4.0/24 [20/0] via 192.168.1.10, 00:15:32
R9_PE_Budapest#
```

2.25. ábra sh ip route vrf Cust_A kimenete

Látható, hogy az összes CE eszköz BGP-vel látja a telephelyeket. Most már tesztelhető a kapcsolat, következzen az útvonalkövetés, R11_CE_Cust_A-ból.

traceroute <IP> source <IP>



```
R11_CE_Cust_A
Press RETURN to get started.

R11_CE_Cust_A#traceroute 192.168.2.1 so 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.2.1

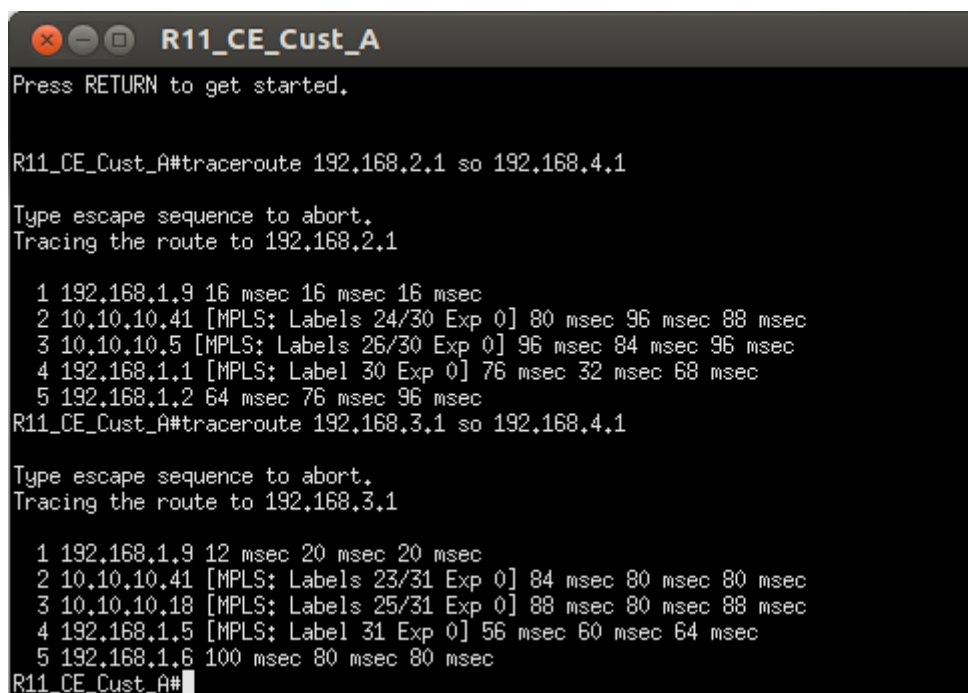
  1 192.168.1.9 12 msec 20 msec 12 msec
  2 192.168.1.1 [MPLS: Label 30 Exp 0] 60 msec 52 msec 52 msec
  3 192.168.1.2 88 msec 80 msec 80 msec
R11_CE_Cust_A#traceroute 192.168.3.1 so 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.3.1

  1 192.168.1.9 20 msec 12 msec 16 msec
  2 192.168.1.5 [MPLS: Label 31 Exp 0] 48 msec 56 msec 68 msec
  3 192.168.1.6 80 msec 80 msec 80 msec
R11_CE_Cust_A#
```

2.26. ábra traceroute kimenete

Látható, hogy mindkét /24-es hálózat elérhető és még az is kivehető, hogy 192.68.1.1 és 192.168.1.5-ös kilépő LSR-nél milyen címkét alkalmaz a hálózat. A hálózat többi tagja nem látható, hiszen a gerinc szolgáltató szempontjából nem előnyös, ha az ügyfél „belelát” a hálózatba, ezért az összes gerinc router-en be lett állítva a *no mpls ip propagate-ttl*. Ha ezt visszaállítjuk, akkor a teljes hálózat lekövethető egy traceroute paranccsal.



```
R11_CE_Cust_A
Press RETURN to get started.

R11_CE_Cust_A#traceroute 192.168.2.1 so 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.2.1

  1 192.168.1.9 16 msec 16 msec 16 msec
  2 10.10.10.41 [MPLS: Labels 24/30 Exp 0] 80 msec 96 msec 88 msec
  3 10.10.10.5 [MPLS: Labels 26/30 Exp 0] 96 msec 84 msec 96 msec
  4 192.168.1.1 [MPLS: Label 30 Exp 0] 76 msec 32 msec 68 msec
  5 192.168.1.2 64 msec 76 msec 96 msec
R11_CE_Cust_A#traceroute 192.168.3.1 so 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.3.1

  1 192.168.1.9 12 msec 20 msec 20 msec
  2 10.10.10.41 [MPLS: Labels 23/31 Exp 0] 84 msec 80 msec 80 msec
  3 10.10.10.18 [MPLS: Labels 25/31 Exp 0] 88 msec 80 msec 88 msec
  4 192.168.1.5 [MPLS: Label 31 Exp 0] 56 msec 60 msec 64 msec
  5 192.168.1.6 100 msec 80 msec 80 msec
R11_CE_Cust_A#
```

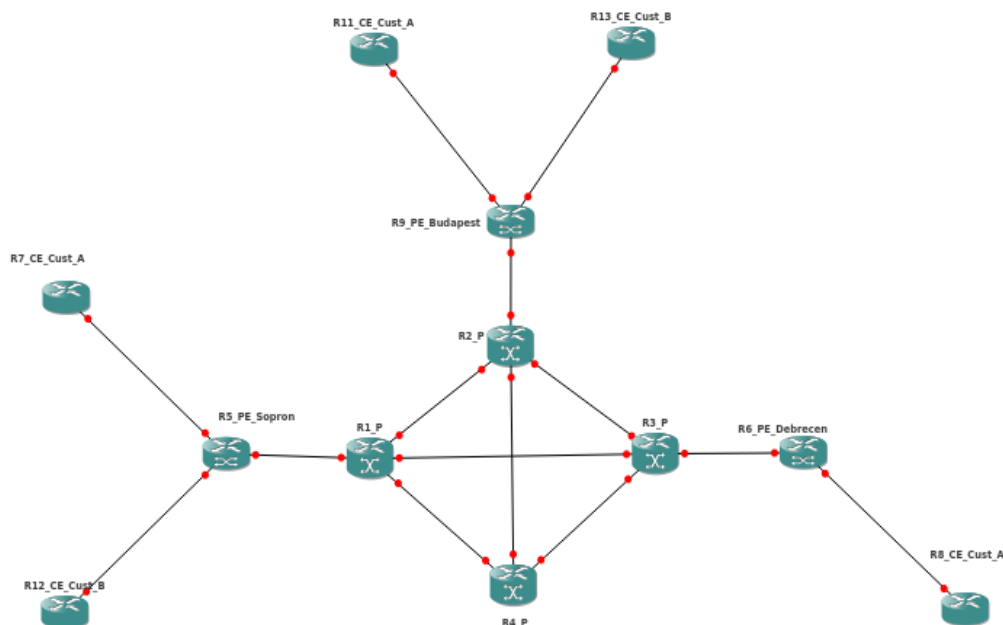
2.27. ábra traceroute kimenete, mpls ip propagate-ttl után

Ebben az esetben az összes MPLS útvonal, címke lekövethető a hálózaton. A továbbiakban ezt a beállítást már nem lesz alkalmazva, most csak a szemléltetés kedvéért lett beállítva az *mpls ip propagate-ttl*.

A traceroute-ból az is látszik, hogy az MPLS hálózatban többszintű címkézés valósul meg, erre azért van szükség, mert VPN is címkével kerül azonosításra.

Még hátra van „B” előfizető csatlakoztatása a gerinchez. „B” előfizető csak két telephely /24-es hálózatát szeretné összekötni, és fontos számára, hogy „A” előfizető ne láthassa, hogy egy hálózaton vannak, ez amúgy „A” számára is fontos. B-nek Sopronban és Budapesten van telephelye.

A végleges hálózati kép a következő:



2.28. ábra Teljes példahálózat

A hálózathálóból látható, hogy a „B” előfizető CE router-ei arra a PE router-re csatlakoznak fizikailag, mint „A” előfizetőé. Viszont a két előfizető nem látja egymást a hálózaton, mert más VRF-be tartoznak. Az előfizetők oldaláról csak egy BGP router látszik, míg a szolgáltatói oldalról két virtuális BGP valósul meg.

A konfiguráció PE router-ekben az alábbiak szerint változik:

R9_PE_Budapest:

```
!  
ip vrf Cust_A  
description Customer_A  
rd 1:100  
route-target export 1:100  
route-target import 1:100  
!  
ip vrf Cust_B  
description Customer_B  
rd 100:200  
route-target export 100:200  
route-target import 100:200  
!
```

BGP-n belül:

```
!  
address-family ipv4 vrf Cust_A  
no synchronization  
neighbor 192.168.1.10 remote-as 65001  
neighbor 192.168.1.10 activate  
neighbor 192.168.1.10 as-override  
exit-address-family  
!  
address-family ipv4 vrf Cust_B  
no synchronization  
neighbor 192.168.101.1 remote-as 65002  
neighbor 192.168.101.1 activate  
neighbor 192.168.101.1 as-override  
exit-address-family
```

!
R5_PE_Sopron:

```
!  
ip vrf Cust_A  
description Customer_A  
rd 1:100  
route-target export 1:100  
route-target import 1:100  
!  
ip vrf Cust_B  
description Customer_B  
rd 100:200  
route-target export 100:200  
route-target import 100:200
```

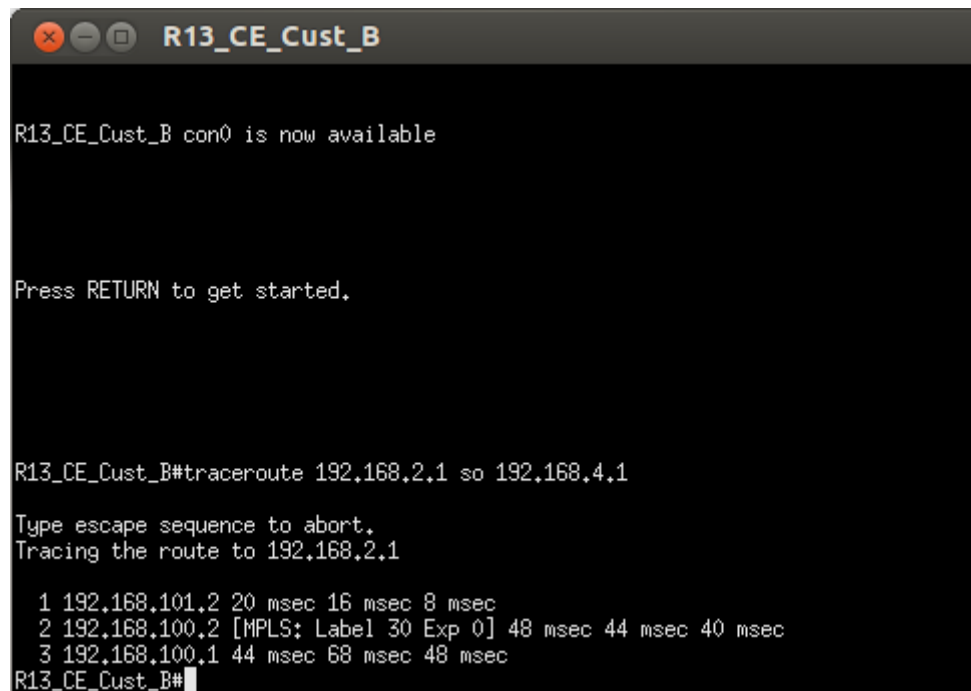
!
BGP-n belül:

```
!  
address-family ipv4 vrf Cust_A  
no synchronization  
neighbor 192.168.1.2 remote-as 65001  
neighbor 192.168.1.2 activate  
neighbor 192.168.1.2 as-override  
exit-address-family  
!  
address-family ipv4 vrf Cust_B  
no synchronization  
neighbor 192.168.100.1 remote-as 65002  
neighbor 192.168.100.1 activate  
neighbor 192.168.100.1 as-override  
exit-address-family  
!
```

„B” előfizető CE oldala csak az AS-ben különbözik. A VPN nagy előnye, hogy az előfizetők használhatják ugyanazt az IP tartományt a LAN-ukra így a szemléltetés kedvéért „A” és „B” Sopronban és Budapesten ugyanazt a /24-es hálózatot használja a LAN-on belül.

Fontos, hogy „B” előfizetőnek más BGP AS kerüljön beállításra, így ő a 65002-est fogja használni.

A hálózat ellenőrzése az alábbi módon néz ki:



```
R13_CE_Cust_B con0 is now available

Press RETURN to get started.

R13_CE_Cust_B#traceroute 192.168.2.1 so 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1
  1 192.168.101.2 20 msec 16 msec 8 msec
  2 192.168.100.2 [MPLS: Label 30 Exp 0] 48 msec 44 msec 40 msec
  3 192.168.100.1 44 msec 68 msec 48 msec
R13_CE_Cust_B#
```

2.29. ábra traceroute paracs kimenete B esetben

Az előzőekhez képest a 192.168.2.1-es IP a 192.168.100.1-es interfész IP-n keresztül válaszol, tehát nem az „A” ügyfél hálózatából jön vissza a válasz. Ha ping paranccsal próbáljuk elérni „A” előfizető hálózatát, akkor az nem sikerülhet, hiszen más a routing táblája „B”-nek mint „A”-nak.

```

R9_PE_Budapest

R9_PE_Budapest#sh ip route vrf Cust_B

Routing Table: Cust_B
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route, % - next hop override

Gateway of last resort is not set

B    192.168.2.0/24 [200/0] via 5.5.5.5, 00:04:28
B    192.168.4.0/24 [20/0] via 192.168.101.1, 00:04:28
     192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.101.0/30 is directly connected, GigabitEthernet2/0
L    192.168.101.2/32 is directly connected, GigabitEthernet2/0
R9_PE_Budapest#

```

2.30. ábra sh ip route vrf Cust_B esetén

Ping parancsra nem jön válasz, ha az „A” debreceni telephelye felé próbálkozik „B”. Emlékeztetőül, A előfizető telephelyének hálózata 192.168.3.0/24 volt. A ping kimenete a következő:

```

R13_CE_Cust_B

R13_CE_Cust_B con0 is now available

Press RETURN to get started.

R13_CE_Cust_B#ping 192.168.3.1 so 192.168.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.4.1
*****
Success rate is 0 percent (0/5)
R13_CE_Cust_B#

```

2.31. ábra Sikertelen ping A előfizető felé.

Az MPLS felhőben a két előfizetőt a címke különbözteti meg. Példaként mindkét előfizető budapesti telephelye ping-eli a soproni telephelyet. Wiresharkban ez az alábbi módon néz ki R2_P g1/0-ás interface-n:

No.	Time	Source	Destination	Protocol	Length	Info
326	10.793018006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0000, seq=186/47616, ttl=254 (reply in 328)
327	10.823434006	192.168.4.1	192.168.2.1	ICMP	122	Echo (ping) reply id=0x0001, seq=324/17409, ttl=254 (request in 325)
328	10.843608006	192.168.4.1	192.168.2.1	ICMP	122	Echo (ping) reply id=0x0000, seq=186/47616, ttl=254 (request in 326)
329	10.863818006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0001, seq=325/17665, ttl=254 (reply in 331)
330	10.874066006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0000, seq=187/47872, ttl=254 (reply in 333)
▶Frame 327: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0						
▶Ethernet II, Src: ca:06:0a:92:00:1c (ca:06:0a:92:00:1c), Dst: ca:05:0a:92:00:1c (ca:05:0a:92:00:1c)						
▼MultiProtocol Label Switching Header, Label: 26, Exp: 0, S: 0, TTL: 254						
0000 0000 0000 0001 1010 = MPLS Label: 26						
..... 000. = MPLS Experimental Bits: 0						
..... 0 = MPLS Bottom Of Label Stack: 0						
..... 1111 1110 = MPLS TTL: 254						
▼MultiProtocol Label Switching Header, Label: 31, Exp: 0, S: 1, TTL: 255						
0000 0000 0000 0001 1111 = MPLS Label: 31						
..... 000. = MPLS Experimental Bits: 0						
..... 0 = MPLS Bottom Of Label Stack: 1						
..... 1111 1111 = MPLS TTL: 255						
▶Internet Protocol Version 4, Src: 192.168.4.1 (192.168.4.1), Dst: 192.168.2.1 (192.168.2.1)						
▶Internet Control Message Protocol						

2.32. ábra A előfizető VPN azonosító címkéje

Ebből látható, hogy „A” előfizetőnek az MPLS hálózatban a VPN azonosításra a 31-es címke került kiosztásra míg „B” előfizető számára a 32-es a Sopron Budapest szakaszokon.

No.	Time	Source	Destination	Protocol	Length	Info
322	10.712082006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0000, seq=185/47360, ttl=254 (reply in 324)
323	10.742426006	192.168.4.1	192.168.2.1	ICMP	122	Echo (ping) reply id=0x0001, seq=323/17153, ttl=254 (request in 321)
324	10.752535006	192.168.4.1	192.168.2.1	ICMP	122	Echo (ping) reply id=0x0000, seq=185/47360, ttl=254 (request in 322)
325	10.782841006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0001, seq=324/17409, ttl=254 (reply in 327)
326	10.793018006	192.168.2.1	192.168.4.1	ICMP	122	Echo (ping) request id=0x0000, seq=186/47616, ttl=254 (reply in 328)
▶Frame 324: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0						
▶Ethernet II, Src: ca:06:0a:92:00:1c (ca:06:0a:92:00:1c), Dst: ca:05:0a:92:00:1c (ca:05:0a:92:00:1c)						
▼MultiProtocol Label Switching Header, Label: 26, Exp: 0, S: 0, TTL: 254						
0000 0000 0000 0001 1010 = MPLS Label: 26						
..... 000. = MPLS Experimental Bits: 0						
..... 0 = MPLS Bottom Of Label Stack: 0						
..... 1111 1110 = MPLS TTL: 254						
▼MultiProtocol Label Switching Header, Label: 32, Exp: 0, S: 1, TTL: 255						
0000 0000 0000 0010 0000 = MPLS Label: 32						
..... 000. = MPLS Experimental Bits: 0						
..... 0 = MPLS Bottom Of Label Stack: 1						
..... 1111 1111 = MPLS TTL: 255						
▶Internet Protocol Version 4, Src: 192.168.4.1 (192.168.4.1), Dst: 192.168.2.1 (192.168.2.1)						
▶Internet Control Message Protocol						

2.33. ábra B előfizető VPN azonosító címkéje

VPN azonosításra a címkeverem alján található címke szolgál, a PE eszköz már csak a VPN azonosítót kapja meg és az alapján route-olja a tovább az ügyfél hálózatába a csomagot. A felső címke az LSR-ek közti továbbítást szolgálja. Látható a címke felépítéséből, hogy ha S bit 1-es, akkor elértük a verem alját.

A PE eszközök végleges címkekapcsolási táblája pedig az alábbi módon néz ki:

The screenshot displays the GNS3 interface with three PE routers (R9, R5, R6) and their MPLS forwarding tables. The tables show the mapping of local labels to outgoing labels and prefixes, along with the next hop information.

R9_PE_Budapest# show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing Interface	Next Hop
16	16	5.5.5.5/32	0	Gi0/0	10.10.10.41
17	17	5.5.5.5/32	0	Gi0/0	10.10.10.41
18	18	3.3.3.3/32	0	Gi0/0	10.10.10.41
19	19	2.2.2.2/32	0	Gi0/0	10.10.10.41
20	20	1.1.1.1/32	0	Gi0/0	10.10.10.41
21	21	10.10.10.16/30	0	Gi0/0	10.10.10.41
22	21	10.10.10.12/30	0	Gi0/0	10.10.10.41
23	22	10.10.10.24/30	0	Gi0/0	10.10.10.41
24	23	10.10.10.32/30	0	Gi0/0	10.10.10.41
25	24	10.10.10.8/30	0	Gi0/0	10.10.10.41
26	25	10.10.10.28/30	0	Gi0/0	10.10.10.41
27	26	10.10.10.4/30	0	Gi0/0	10.10.10.41
28	27	10.10.10.20/30	0	Gi0/0	10.10.10.41
29	18	4.4.4.4/32	0	Gi0/0	10.10.10.41
30	No Label	192.168.4.0/24[V]	\	Gi1/0	192.168.1.10
31	No Label	192.168.4.0/24[V]	\	Gi2/0	192.168.101.1

R5_PE_Sopron# show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing Interface	Next Hop
17	20	6.6.6.6/32	0	Gi0/0	10.10.10.30
18	21	3.3.3.3/32	0	Gi0/0	10.10.10.30
19	22	1.1.1.1/32	0	Gi0/0	10.10.10.30
20	23	10.10.10.24/30	0	Gi0/0	10.10.10.30
21	24	10.10.10.16/30	0	Gi0/0	10.10.10.30
22	25	10.10.10.12/30	0	Gi0/0	10.10.10.30
23	26	10.10.10.32/30	0	Gi0/0	10.10.10.30
24	27	10.10.10.4/30	0	Gi0/0	10.10.10.30
25	28	10.10.10.8/30	0	Gi0/0	10.10.10.30
26	29	7.7.7.7/32	0	Gi0/0	10.10.10.30
27	30	4.4.4.4/32	0	Gi0/0	10.10.10.30
28	31	2.2.2.2/32	0	Gi0/0	10.10.10.30
29	32	10.10.10.40/30	0	Gi0/0	10.10.10.30
30	33	10.10.10.20/30	0	Gi0/0	10.10.10.30
31	No Label	192.168.2.0/24[V]	\	Gi1/0	192.168.1.2
32	No Label	192.168.2.0/24[V]	\	Gi2/0	192.168.100.1

R6_PE_Debrecen# show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing Interface	Next Hop
17	17	3.3.3.3/32	0	Gi0/0	10.10.10.33
18	18	10.10.10.24/30	0	Gi0/0	10.10.10.33
19	19	10.10.10.16/30	0	Gi0/0	10.10.10.33
20	20	10.10.10.8/30	0	Gi0/0	10.10.10.33
21	21	7.7.7.7/32	0	Gi0/0	10.10.10.33
22	22	5.5.5.5/32	0	Gi0/0	10.10.10.33
23	23	4.4.4.4/32	0	Gi0/0	10.10.10.33
24	24	2.2.2.2/32	0	Gi0/0	10.10.10.33
25	25	1.1.1.1/32	0	Gi0/0	10.10.10.33
26	26	10.10.10.12/30	0	Gi0/0	10.10.10.33
27	27	10.10.10.40/30	0	Gi0/0	10.10.10.33
28	28	10.10.10.20/30	0	Gi0/0	10.10.10.33
29	29	10.10.10.4/30	0	Gi0/0	10.10.10.33
30	30	10.10.10.28/30	0	Gi0/0	10.10.10.33
31	No Label	192.168.3.0/24[V]	\	Gi1/0	192.168.1.6

2.34. ábra PE routerek címkekapcsolási táblája

Ezzel elkészült az tervezni kívánt MPLS/VPN példahálózat, a konfigurációkat minden eszközben le kell menteni a *write memory* parancs segítségével, és a GNS3-ban is menthető projektként.

2.7. Összegzés

Az eddigiek alapján bemutatásra került a szimulációs környezet felépítése, a GNS3 szoftver felépítése, és beállításai, valamint az, hogy hogyan kell a hálózatokat összeállítani, illetve az eszközöket konfigurálni. A példahálózaton keresztül látható volt az MPLS gerinc felépítése, LSR-k, LER-k konfigurációja. A hálózat ki lett egészítve VPN funkcióval, ami lehetővé tette, hogy több előfizető csatlakozzon különböző telephelyekkel ugyanahhoz a gerinchez, ugyanarra az eszközre, ami a gerincszolgáltató szempontjából gazdaságosabb, mint külön-külön csatlakoztatni őket. Bemutatásra került az is, hogy meglévő hálózatot miként lehet újabb és újabb telephelyekkel bővíteni. A végső következtetés: az MPLS/VPN alapú hálózatok, olcsók, egyszerűek, könnyen bővíthetők, de mégis biztonságosak. Rendelkezésre állásuk csupán a terheltségtől, és az eszközök minőségétől függ. A példahálózatot még továbbfejleszthető egy menedzsment hálózat létre hozásával, valamint a Traffic Engineering funkciók kihasználásával, és a szövevényes BGP kapcsolatok helyett, lehet csillagpontos BGP-t létrehozni route reflektorok segítségével, továbbá IPv6-ra is lehet fejleszteni. Ezek közül néhány elem bemutatásra kerül az utolsó fejezetben, laborhálózat megvalósításánál.

Összefoglalás

Az ATM és az IP hálózatok előnyeinek egyesítése céljából megjelenő MPLS eljárást részletesen bemutatam. A dolgozat tartalmazza az MPLS továbbítás mechanizmusának, a címke felépítésének és beépülésének menetét különböző adatkapcsolattal rendelkező hálózatokba, ismertetésre került a címketerjesztésért felelős LDP, valamint a hálózat menedzselését szolgáló Traffic Engineering funkció, továbbá a VPN funkció. A GMPLS koncepció irányába hajló szolgáltatók miatt bemutatam ezt az elvet is.

A GNS3 szimulációs környezetnek köszönhetően megterveztem egy teljes gerinchálózatot, amely alkalmazza az MPLS eljárást. A hálózat felépítésének különböző pontjain keresztül betekintheünk a Cisco IOS parancsok funkciójába, az eszközök konfigurálásába.

A hálózat fejlesztése során MPLS/VPN funkcióval lett ellátva és lehetőség lett több előfizetői végpont megvalósítására. A hálózatban további fejlesztési lehetőségek maradtak nyitva, mint például a Traffic Engineering alkalmazása és menedzsment hálózat kialakítása.

A végső konklúzió, hogy az MPLS forradalmasította az átviteltechnikát, ennek és a Carrier Ethernet technológiának köszönhetően az IP alapú gerinchálózatok implementációja könnyebb feladattá vált. Az MPLS-ben további fejlesztési lehetőségek rejlenek, amiket a jövő generációinak érdemes lesz kihasználni.

Irodalomjegyzék

Könyvek:

- [1] **Lobo, Lancy; Lakshman, Umesh:** MPLS configuration on Cisco IOS software. Indianapolis: Cisco Press, 2006.
- [2] **Pepelnjak, Ivan; Guichard, Jim:** MPLS and VPN Architectures. Indianapolis: Cisco Press, 2002.
- [3] **Thomas, Stephen A.:** IP kapcsolás és útválasztás: RIP, OSPF, BGP, MPLS, CR-LDP, RSVP-TE. Budapest: Kiskapu, 2002.
- [4] **Tomsu, Peter; Wieser, Gerhard:** MPLS-based VPNs : designing advanced virtual networks. Upper Saddle River: Prentice Hall, 2002.

RFC dokumentumok:

- [5] **Bates, T.; Chandra, R.; Katz, D.; Rekhter, Y.:** Multiprotocol Extensions for BGP-4. 1998. RFC2283.
- [6] **Blake, S.; Black, D.; Carlson, M.; etc.:** An Architecture for Differentiated Services. 1998. RFC2475
- [7] **Braden, R.; Clark, D.; Shenker, S.:** Integrated Services in the Internet Architecture: an Overview. 1994. RFC1633
- [8] **Braden, R.; Zhang, L.; Berson, S.; etc.:** Resource ReSerVation Protocol (RSVP). 1997. RFC2205
- [9] **Harkins, D. and Carrel, D:** The Internet Key Exchange (IKE). 1998. RFC2409
- [10] **Nichols, K.; Blake, S.; Baker, F.; Black, D.:** Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. 1998. RFC2474
- [11] **Rosen, E.:** Removing a Restriction on the use of MPLS Explicit NULL. 2005. RFC4182.
- [12] **Rosen, E.; Rekhter, Y.:** BGP/MPLS VPNs. 1999. RFC2547.
- [13] **Rosen, E.; Tappan, D., Fedorkow, G.; etc.:** MPLS Label Stack Encoding. 2001. RFC3032.
- [14] **Rosen, E.; Viswanathan, A.; Callon, R.:** Multiprotocol Label Switching Architecture. 2001. RFC3031.

Folyóiratok:

- [15] **Szigeti János, Tapolcai János, Rétvári Gábor, Láposi Levente, Cinkler Tibor** Útvonalkijelölés és forgalomelvezetés több tartományú kapcsolt optikai hálózatokban, Híradástechnika, LIX. évfolyam, 2004/2, 42-49.