DIPLOMAMUNKA

Baracsi Pál

Kovács Zoltán

Terdik Sándor

Debrecen

2010

Debreceni Egyetem
Informatika Kar

MPLS ALAPÚ VIRTUÁLIS MAGÁNHÁLÓZATOK NAPJAINKBAN

Készítették:
Baracsi Pál
Programtervező informatikus (MSc)
Kovács Zoltán
Programtervező matematikus
Terdik Sándor
Programtervező matematikus

Témavezető:
Dr. Almási Béla
egyetemi docens

Debrecen
2010

Tartalomjegyzék

Bevezetés

Manapság napjaink szerves részévé vált a kommunikáció. Gyakorlatilag az élet minden területén jelen van: magánéletben, munkában, szórakozásban és még sorolhatnánk a teljesség igénye nélkül, ahol az internet, mint kommunikációs csatorna szerepel.

Több szinten beszélhetünk kommunikációról. Kezdve otthonunk internetes kapcsolatától vagy épp munkahelyünk helyi hálózatától, egy telephely épületei közötti hálózati összeköttetésen át, az internet szolgáltatók között zajló kapcsolatrendszer megvalósításával bezárva.

Eddigi tanulmányaink során a hálózatokkal való ismerkedés és tudásfejlesztés folyamatosan jelen volt. Számos tanfolyam, verseny és projekt kapcsán helyi hálózatok szintjén magabiztos tudásra tettünk szert, mellyel úgy éreztük megálljuk majd a helyünket az egyetem elvégzése után is.

Kaptunk azonban egy nagyon jó lehetőséget, melynek kapcsán alkalmunk nyílt betekinteni a telekommunikációs internetszolgáltatók világába és rájöttünk, hogy tudásunk ezen a területen jelentős fejlesztésre szorul.

Az IT Services Hungary Kft. adott otthont ezen új ismeretek megszerzéséhez és alapját képezte ezen diplomamunka megírásának.

A diplomamunka témáját illetően elképzeléseink nagymértékben megegyeztek, ezért konzulensünk javaslatára egy nagyobb terjedelmű dolgozatot készítettünk.

Témáinkkal igyekeztük lefedni napjaink WAN kommunikációinak legfontosabb fogalmait, technikáit.

Elsőként bemutatjuk az autonóm rendszerek közötti – és azon belüli – forgalomirányítási információk cseréjét biztosító külső határátjáró protokollt, a BGP-t. A fejezet részletes kidolgozását Terdik Sándor készítette.

A következő fejezetben, - amelyet Baracsi Pál dolgozott ki - az MPLS architektúra fogalmainak és működésének tárgyalása következik. Az MPLS architektúra egy címkekapcsolási módszert ír le, ami a Layer2 switching és a Layer3 routing előnyeit integrálja. A Layer2 (ATM, Frame Relay) hálózatokhoz hasonlóan az MPLS is címkét rendel az adategységekhez, a csomag-, illetve cellaalapú hálózatokon való továbbításhoz. A továbbítási mechanizmus címkecserén (label swapping) alapul, amely igen gyors adattovábbítást tesz lehetővé.

Bemutatjuk a virtuális magánhálózatok (VPN) kialakítását MPLS alapú hálózatban. A módszer nagy előnye és egyben népszerűségének oka, hogy rugalmas és egyben jól

skálázható megoldást kínál a szolgáltatók számára. Lehetővé téve ezzel nagyszámú ügyfelek igényeinek kielégítését költséghatékony módon. A fejezet részeként bemutatjuk a különböző VPN topológiákat. Ezen fejezet első felét Kovács Zoltán, második felét pedig Baracsi Pál készítette el.

Napjaink hálózataiban egyre nagyobb szükség van szolgáltatásminőségi szintek meghatározására az alkalmazások különböző sávszélesség igénye és adatforgalmának jellege miatt. Különböző elvárásokat támasztunk az IP alapú hangátvitel, videokonferenciák és vállalatok kulcsfontosságú üzleti alkalmazásaival szemben. Az MPLS környezet lehetőséget teremt a különféle szolgáltatásminőségi szintek hatékony megvalósítására, melyet Kovács Zoltán ismertet.

Tárgyaljuk a hibatűrő ügyfélhálózatokat megvalósító HSRP protokollt, mely az alapértelmezett átjáró esetlegesen előforduló meghibásodását küszöböli ki. A fejezet elkészítését Terdik Sándor valósította meg.

Említésre kerül néhány backup megoldás, mely az ügyfelek kiesés nélküli hálózatelérését biztosítja. Elsőként a talán legszélesebb körben elterjedt VPN technológiát, az Ipsec alapjait ismerteti röviden Kovács Zoltán, amely hálózati szinten nyújt lehetőséget arra, hogy a kommunikációban résztvevők hitelesen azonosítsák egymást és kódolják adatforgalmukat. Továbbá Baracsi Pál röviden összefoglalja a telefonhálózatok igen nagy lefedettségét kihasználó ISDN technológiát.

Dolgozatunk részét képezi a GNS3 szimulációs környezet és az ebben felépített bemutató hálózat működésének és konfigurációs lépéseinek ismertetése. A GNS3 bemutatását és a reprezentatív modell felépítését, működésének leírását közösen készítettük el.

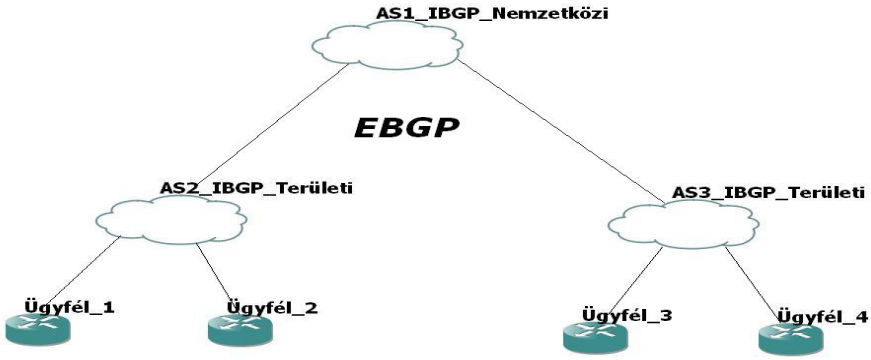
Leírásaink során az RFC szabványdokumentumok jelentik a fő irányvonalat, továbbá a világ egyik legnagyobb hálózati cégének, a Cisco-nak a leírásai, megvalósításai.

1. Határ átjáró protokoll: BGP-4

A BGP (Border Gateway Protocoll) első verziója 1989 júniusában jelent meg [RFC1105], a második 1990-ben [RFC1163], a harmadik 1991-ben [RFC1267]. Ez a harmadik verzió működött az Interneten 1991-től 1994-ig. A BGP negyedik verziója is elkészült 1995 márciusában [RFC1771], legfőbb újítása a CIDR (Classless Inter-Domain Routing). Ennek értelmében a fejezet vázát az RFC1771 szabvány dokumentum képezi, melynek ismérvei kiegészítésre kerülnek Cisco specifikus fogalmakkal, megvalósításokkal.

1.1 BGP (Border Gateway Protocoll) fogalma

A BGP protokoll bemutatásakor nélkülözhetetlen az autonóm rendszer fogalmának definiálása. Az autonóm rendszer (Autonomous System, továbbiakban AS) nem más, mint közös adminisztrációs és fogalomirányítási szabályozás alá eső hálózatok csoportja.
Az AS-eken belül úgynevezett belső átjáró protokollok biztosítják a hálózatelérési információk terjesztését. Ilyen protokoll például a RIPv1, RIPv2, OSPF, EIGRP.
Az Internet azonban meglehetősen sok autonóm rendszerből tevődik össze, ezért van szükség olyan protokollokra, amelyek ezen rendszerek közt igazítanak el. Ezeket hívjuk külső átjáró protokolloknak. Az Interneten jelenleg használt ilyen protokoll a BGP és annak is a 4-es verziója.
A BGP egy autonóm rendszerek közötti (inter-Autonomous System vagy interdomain) forgalomirányító protokoll. A BGP elsődleges funkciója, hogy hálózatelérési információkat cseréljen más BGP rendszerekkel, de ugyanakkor az internet szolgáltatók (Internet Service Provider, továbbiakban ISP) között alkalmazott protokoll is. Mint már említettük az ügyfélhálózatok, mint például egyetemek, vállalatok általában valamilyen belső átjáró protokollt (Interior Gateway protocol, IGP) használnak, mint például RIP vagy OSPF a forgalomirányítási információk hálózaton belüli cseréjére. Az ügyfelek az ISP-hez kapcsolódnak, az ISP-k pedig BGP-t használva oldják meg az ügyfél, és ISP útvonalak cseréjét. Abban az esetben, amikor a BGP-t AS-k között használjuk külső (external) EBGP –ről beszélünk. Ha pedig az ISP használja a BGP-t az útvonalak AS –en belüli cseréjére, belső (interior) IBGP-ről van szó.



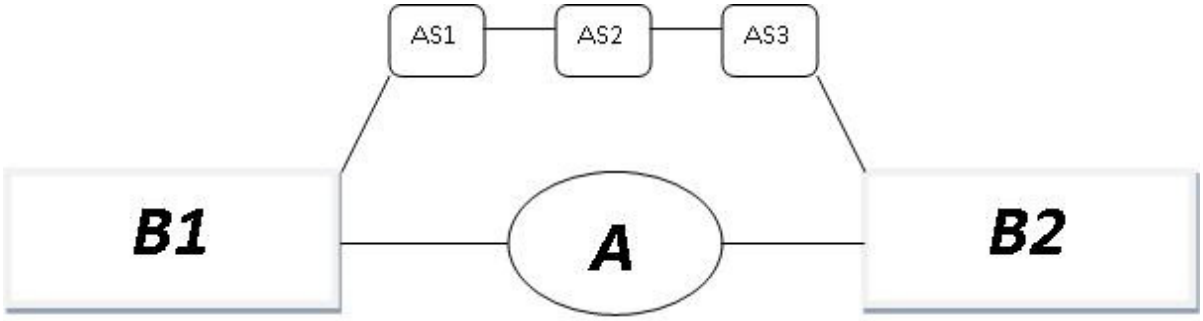
1. ábra

1.1.1 A BGP, mint IGP bemutatása

Igaz, hogy a BGP-t AS-k közötti útvonalterjesztésre tervezték, de számos olyan szolgáltatást nyújt, amelyek miatt AS-eken belül is célszerű használni. Képzeljük el azt a szituációt, hogy egy autonóm rendszer több, BGP–t futtató útválasztón keresztül kapcsolódik a külvilághoz. Ez esetben össze kell hangolniuk működésüket, hogy egységes képet alkossanak az AS-ről a külvilág felé. Erre nyújt megoldást az IBGP néven emlegetett belső BGP protokoll, mellyel a BGP-t futtató forgalomirányítók közvetlenül kapcsolatba léphetnek egymással anélkül, hogy nagy mennyiségű adataikkal valamely belső átjáró protokollt (RIP, EIGRP, OSPF) terhelnék.
Az IBGP és az EBGP működése meglehetősen hasonló egy lényeges eltérést kivéve, mely az útválasztási ciklusokra vonatkozik. Erre későbbiekben kitérünk.

1.1.2 BGP jellemzői, különbözőségei, útvonaltükrözők

Ahhoz, hogy megértsük a BGP és a belső átjáró protokollok működése közti alapvető különbséget, vázoljuk fel az alábbi szituációt.
Legyen adott az „A” vállalat és legyen adott a „B” vállalat két adott alvállalata „B1” és „B2”, mely két alvállalat kapcsolatban áll az „A” vállalattal marketing szempontból.



2. ábra

A szituáció legyen a következő: az „A” vállalat szeretné, ha kapcsolatban lenne mind „B1” mind „B2”-vel de azt szigorúan ellenzi, hogy „B1” és „B2” egymás forgalmát rajta keresztül bonyolítsák. Ehelyett az AS1-AS2-AS3 útvonal legyen preferált.

Ebben a megvilágításban most már a cél nem csak az, mint a belső átjáró protokollok esetében, hogy a legkedvezőbb úton eljussunk egyik helyről a másikra, hanem hogy figyelembe vegyük az egyes területek tiltásait. Erre kiváló lehetőséget nyújt a BGP a házirend alapú útválasztás támogatásával. A házirendek lényege a kapcsolatok korlátozása, melyre a hálózatok legalább három különböző módon képesek:

- A legegyszerűbb módszer a sávszélesség korlátozása. Akár tiltani is lehet ezzel a módszerrel, ha a tiltást 0 Kbit/s-os sávszélességnek fogjuk fel. A házirend megvalósításának meglehetősen durva módja ez, amely azonban nem tesz különbséget a különböző típusú adatok forgalma között.
- A második módszer a továbbítási pontok által alkalmazott csomagszűrők, melyek a házirendeket IP rétegbeli továbbítási döntések korlátozásával alkalmazzák. Hátránya, hogy minden egyes adategységet ellenőrizni kell, hogy teljesítik-e a feltételeket.
- A BGP a házirendek harmadik típusú megvalósításán alapszik, amely a házirend alapú útválasztás. Használatakor arra teszünk korlátozásokat, ahogy az útválasztók az adatokat terjesztik. Az elgondolás a 2. ábrára vonatkozóan a következő. Ha a „B1” alvállalat nem tudja, hogy a „B2” alvállalat az „A” vállalat másik végén található, akkor értelemszerűen nem is küldi arra a csomagokat. A módszer nem igényel egyedi számításokat, az útválasztók a szokásos továbbítási döntéseiket hozzák meg az adategységekre vonatkozóan igaz, hogy nem a teljes hálózati kép ismeretében. A forgalomirányító protokollok nyelvére lefordítva ez nem jelent mást, minthogy az „A”

vállalat forgalomirányítói kihagyják a „B2” felé vezető útvonalakat a „B1” számára küldött hirdetésekből.

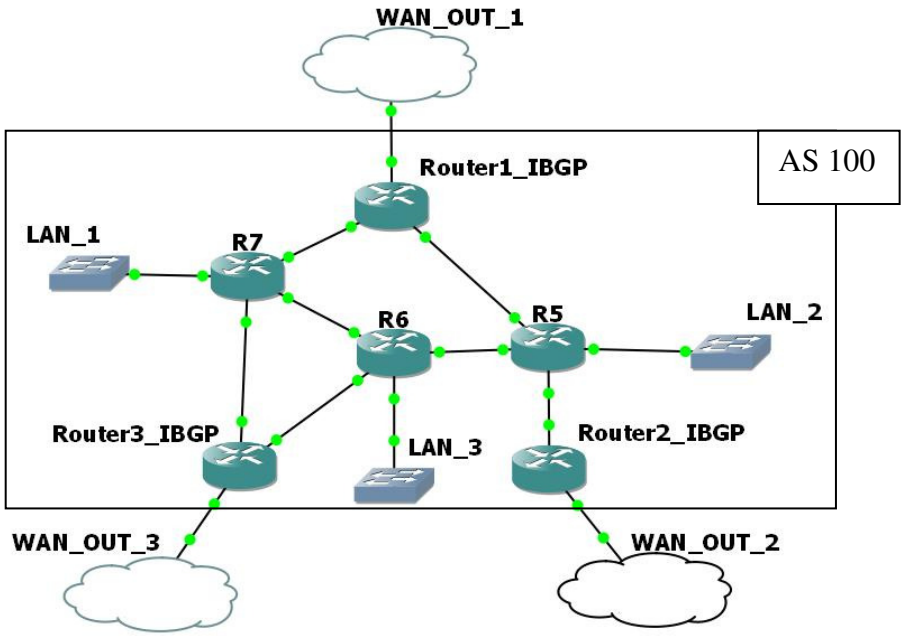
A BGP egy robosztus és méretezhető hálózati protokoll, amely nyilvánvaló, hiszen az interneten használják. A méretezhetőség eléréséhez számos útvonal paramétert használ, melyeket attribútumoknak nevezünk (ld.: később), annak érdekében, hogy meghatározza a forgalomirányítási szabályokat, házirendeket és stabil forgalomirányítási környezetet tartson fenn.

A BGP az attribútumokon túl osztály nélküli forgalomirányítást (Classless InterDomain Routing, CIDR) használ az internetes forgalomirányítási táblák méretének csökkentésére. Ez lehetővé teszi az IP prefixek hirdetését, és szükségtelenné teszi a hálózati osztályok értelmezését a BGP-n belül.

Jellemzője, hogy a BGP –t futtató minden forgalomirányító felépíti az AS irányított gráfját, amelyen az optimalizálás történik. A távolságvektor alapú működés fejlesztéseként megbízható frissítéseket használ a forgalomirányítási információk cseréjére, de csak triggerelt módon, azaz csak változások esetén küld frissítést. A megbízható frissítések a TCP szállítási rétegbeli protokoll használatán alapszik (TCP 179-es port), továbbá ezt támogatják a periodikus keepalive üzenetek és a kötegelt frissítések is.

A BGP és az IGP –k közti szintén nagyon fontos eltérés, hogy míg az IGP- k általában valamilyen költséget rendelnek az útvonalakhoz, a BGP megszabadul a mértékektől és leírja a pontos útvonalat. A BGP egy távolság vektor alapú protokoll és egyben útvonal (vektor) protokoll is, mely szerint egy forgalomirányító nem csak azt tudja, hogy egy távoli hálózat hol van, hanem azt is, hogy a távoli hálózatot min keresztül lehet elérni. Ennek megvalósítása az útvonal (path) attribútum segítségével történik.

Itt térnénk ki rá miért is okoz ez a fajta útvonaltárolás problémát IBGP esetén. Tekintsük a következő ábrát!



3. ábra

Az AS-en belül az IBGP forgalomirányítók a BGP értelmezése szerint minden frissítést ciklusként értelmeznének, hiszen az útvonal attribútumban szerepelne saját AS azonosítójuk, mivel egy autonóm rendszeren belül helyezkednek el. Ezért a BGP működésének azon szabályát, hogyha egy forgalomirányító olyan frissítést kap, amelyben szerepel saját AS száma ciklust feltételezve el kell vetni, módosítani kell IBGP esetén. Tehát az ilyen jellegű frissítéseket nem vetheti el, de ugyanakkor nem is hirdetheti tovább, hiszen ez ciklus kialakulásához vezethet. Például a Router2_IBGP forgalomirányító továbbítja a Router1_IBGP útvonalait a Router3_IBGP útválasztónak. Így Router3_IBGP azt hiszi, hogy a WAN_OUT_1 célcímeit elérheti mind az első mind a második útválasztón keresztül. Ha a Router3_IBGP hasonlóan hibásan visszahirdeti a Router2_BGP –nek az első forgalomirányító útvonalait, annak (Router1_IBGP) meghibásodása esetén mindkét útválasztó azt gondolja a másikon keresztül elérhető a fenti külvilág. Az üzenet a végtelenségig fog bolyongani. Ennek a problémának egyik lehetséges megoldása a teljes háló (full mesh) topológia, amelynél nem lehetséges az, hogy egy IBGP forgalomirányító ne tudjon egy adott útvonalról, egy frissítés az előbb említett okokból történő elmaradása miatt.

A teljes háló topológia kisszámú útválasztók esetén kivitelezhető, ám a forgalomirányítók számának növekedése miatt az elvárások teljesíthetetlenek lesznek. Ilyenkor vehetjük hasznát a BGP egyik újdonságának, az útvonaltükrözőnek (route reflector).

Az útvonaltükröző nem tesz mást, mint megszegi az IBGP előbb említett szabályát, tehát újrahirdeti a kapott útvonalakat az AS-en belül. Használata akkor ajánlott, ha a teljes háló topológia nem teljesíthető. Az ábrán mivel a Router2_IBGP és a Router3_IBGP nincs közvetlen IBGP kapcsolatban ezért nem értesülhetnek fontos útvonalakról. Azonban ha Router1_IBGP útvonaltükröző szerepet tölt be, a probléma megoldódik, hiszen a két útvonalválasztó útvonalait kölcsönösen továbbítja egymásnak. Ennek nyilván ára van, hiszen így ismét forgalomirányítási hurkok alakulhatnak ki. Erre szolgál a fürt (cluster) fogalmának bevezetése.

A fürtök egy útvonaltükrözőből és rá támaszkodó IBGP útválasztókból állnak. Az útvonaltükrözőket irányító szabályok igen egyszerűek. A fürt tagjaitól érkező útválasztási adatok eljutnak minden kapcsolódó rendszerhez, míg a kívülről érkezők csak a fürt tagjainak továbbítódnak. Annak biztosítására, hogy az esetleges hibák ne eredményezzenek ciklusokat az útvonaltükrözők adataikat mindig egy eredetazonosítóval jelölik meg, amely egyértelmű az IBGP útválasztók között. Ha az útválasztó olyan útvonalhoz jut, amely már meg van jelölve saját eredetazonosítójával, tudja, hogy az üzenet ciklusba került és elveti.

Ezen túl a tükrözött útvonalak tartalmaznak egy fürtlistát is, amely azonosítja azokat az útvonal tükrözési fürtöket, amelyeken az üzenet már áthaladt. Ha egy útvonaltükröző saját fürtjét észleli egy ilyen listában, felismeri a ciklust és elveti az üzenetet.

1.2 Működési elv

Számos más forgalomirányító protokollal szemben a BGP nem detektálja automatikusan a szomszédjait. A szomszédok manuális konfigurációja szükséges a már említett TCP viszony kialakításához.

Mivel a BGP kapcsolat TCP fölött épül ki a protokoll egyszerűbbé vált, mert nincs szükség annak figyelésére, hogy elküldött üzenetünket nyugtázta-e már a vevő fél.

Hátránya viszont, hogy így a titkosítás megoldhatatlan a TCP titkosítása nélkül, hiszen elég egy a TCP kapcsolat bontására szolgáló csomagot küldeni az egyik oldalnak és a kapcsolat lebomlik. Így könnyű zavart okozni. Ennek orvoslására hozták lére az MD5 aláírási beállítást. Az MD5 működése egy titkos kulcson alapszik melyet a BGP konfigurálása során kell megadnunk a megfelelő forgalomirányítókon ugyanazon titkos kulccsal. A forgalomirányító a küldés előtt, meghatározott adatokra lefuttatja az MD5 algoritmust, a másik forgalomirányító szintén és összeveti a kapott eredményt az általa generált eredménnyel. Ha nem egyezik,

elveti. Ugyanígy jár el, ha egy forgalomirányítón engedélyezve van az MD5 aláírás és olyan üzenetet kap, amely nem tartalmazza ezt a beállítást.

Másik hátrány, hogy a TCP meglehetősen érzékeny a torlódásokra (sok újraküldés), ám ez okos TCP implementációkkal kivédhető. Minthogy a TCP byte-stream jellegű kommunikációt tesz lehetővé, a BGP pedig üzenetekben kommunikál, minden üzenet elején 16 byte szolgál az üzenet elejének azonosítására és a jövendőbeli hitelesítésre. Ezen felül az üzenet tartalmazza saját hosszát, így a vevőnek nincs más dolga, mint bevárni a megadott számú byte-ot, utána rögtön a következő üzenet eleje jön majd.

A BGP forgalomirányítók a TCP kapcsolat kiépítése után egy OPEN üzenetet küldenek el, melyben megegyeznek a BGP használt verziójáról, közlik saját AS-ük számát, saját azonosítójukat és azt, hogy ők milyen gyakran küldenek majd KEEPALIVE üzeneteket. Ha mind a két forgalomirányító egy időben épített ki TCP kapcsolatot (ütközés), akkor a kisebbik azonosító által létrehozottat törlik. A kapcsolat véglegesítése előtt mindkét fél leellenőrzi, hogy az adminisztráció engedélyezte-e a másikkal való kapcsolattartást. Ha nem, bontják a kapcsolatot.

Ha a kapcsolat kiépült, a BGP forgalomirányítók a terjesztésre szánt összes útvonalukat közlik egymással. Miután ez megtörtént, már csak a változásokról értesítik egymást. A hibákról (például helytelen szintaktikájú üzenet, rossz BGP verziószám, stb.) egy harmadik fajta üzenetben értesítik egymást (NOTIFICATION).

A TCP használata miatt a partnereknek nincs lehetőségük a kapcsolat meglétének tesztelésére, hisz ha nincs átküldendő adat, a TCP nem bocsát ki IP csomagokat. Ennek érdekében, ha már egy jó ideje se UPDATE, se NOTIFICATION üzenet elküldésére nem volt szükség, akkor egy rövid KEEPALIVE üzenetet küldenek egymásnak, a kapcsolat kiépítésekor megállapított időközönként. Ha ez alatt az idő alatt semmiféle üzenet nem jön, vagy a TCP a kapcsolat megszakadása következik be, akkor a BGP elérhetetlennek tekinti partnerét.

Ismert még a ROUTE-REFRESH üzenet, mellyel a forgalomirányító a teljes útválasztási adatbázist kérheti partnerétől.

1.2.1 Útvonalhirdetés és tárolás

Ezen protokoll értelmezésében az útvonal nem más, mint a cél és a célhoz vezető útvonal attribútumainak együttese. A cél az NLRI (Network Layer Reachability Information)

mezőben bejegyzett rendszerek IP címének valamelyike, az útvonal pedig ugyanazon UPDATE üzenet útvonal attribútum mezőjében szereplő információ.

Az útvonalak a BGP párok között UPDATE üzenetek formájában hirdetődnek, míg a Routing Information Bases (RIBs) – ben tárolódnak, nevezetesen az ADJ-RIB-IN, LOC-RIB és ADJ-RIB-OUT információs adatbázisokban.

A BGP viszony kiépülése után az útvonalhirdetések elkezdenek beérkezni az adott forgalomirányítóra. Minden frissítés egy vagy több bejegyzésből állhat. Minden útvonal egy IP címmel és hálózati maszkkal van leírva számos attribútum kíséretében. A NEXT-HOP, AS-PATH, ORIGIN attribútumok kötelezően szerepelnek, de más opcionális attribútumok is jelen lehetnek.

Minden kapott útvonal eltárolódik az útválasztó memóriájában, ezért szükségtelen a módosítás nélküli információk újraküldése, frissítése. Mivel több lehetséges út is létezik a megfelelő hálózathoz a helyi fogalomirányító kiválaszt egyet, mint legjobb útvonalat.

A lokális forgalomirányító csakis a legjobbnak választott útvonalat hirdeti a szomszédok felé, de mivel az összes alternatív útvonal le van tárolva a memóriájában, képes új legjobb útvonal választására amennyiben a szomszédos forgalomirányító elérhetetlenné válik vagy eltávolítja az addig legjobb útvonalat.

Fontos szabály, hogy egy útválasztó sosem küld vissza útvonalat oda, ahonnan azt kapta egy BGP kapcsolat alatt (látóhatár-megosztás). Éppen ellenkezőleg amikor kiválasztódik a legjobb „következő ugrás”, a forgalomirányító meggyőződik róla, hogy a szomszéd nem mutat-e vissza a lokális forgalomirányítóra. Ha mégis, ezt az úgynevezett útvonalmérgezés módszerrel valósítja meg (az útvonalat elérhetetlennek jelöli meg) és küld egy WITHDRAW („forgalomból kivon”) üzenetet a megfelelő szomszédjának.

Amikor a BGP partner kiválasztja a hirdetendő útvonalat a hirdetés előtt hozzáadhat vagy módosíthat az útvonal attribútumain.

Többféle mechanizmus létezik arra, hogy egy partner, hogyan informálhatja szomszédját arról, hogy az előzőleg hirdetett útvonal nem használható a továbbiakban. Három fajta metódusról beszélhetünk, amellyel egy BGP viszonyban részt vevő, forgalomból kivontnak jelezhet egy útvonalat:

- IP előtagban az UPDATE üzenet WITHDRAW ROUTES mezőjében van jelölve.
- Helyettesítő útvonal hirdetése azonos hálózati rétegbeli elérhetőségi információval (Network Layer Reachability Information, NLRI).

- A BGP kapcsolat lezárása, mely az összes útvonalat eltávolítja, amelyet a párok hirdetnek egymásnak

Az útvonalak feldolgozásának részleteit az UPDATE üzenetek kezelés kapcsán még részletesen tárgyaljuk.

1.2.2 RIB fogalma

A BGP 3 listát (Route Information Base, RIB) tart nyilván, amelyben az olyan utakat tárolja, melyeket szomszédjaitól kapott (ADJ-RIB-IN), melyeket terjeszt (ADJ-RIB-OUT) és azokat, amelyeket az adott AS használ (LOC-RIB). Az első két listából minden szomszédos AS-hez tartozik egy-egy, azoknak az utaknak, amit onnan kapott és oda terjeszt.

A BGP döntési folyamata az adatbázisok vonatkozásában a következő:

1. Az ADJ-RIB-IN-ből egy helyi függvény alapján kiszámolja minden útvonal preferenciáját, a legjobbnak ítélt utakat a belső BGP kapcsolatokon keresztül terjeszti.
2. A belső szomszédok az ily módon kapott információt hozzáveszik saját adatbázisukhoz és ebből az összegzett úthalmazból meghatározzák minden célponthoz az AS számára rendelkezésre álló legjobb útvonalat, ezeket elhelyezik a LOC-RIB-be.
3. A LOC-RIB-ben levő információt csoportosítják és a helyi politikának megfelelő részét áthelyezik az ADJ-RIB-OUT-ba.

1.3 Üzenet formátumok

A fejezet a BGP által használt üzenetek funkcióját és formátumát írja le. Mint már említettük az üzenetek megbízható szállítási protokoll által kiépített kapcsolaton keresztül továbbítódnak. Feldolgozása nem kezdődik el a teljes üzenet vétele előtt. Maximális méretük 4096 byte. A legkisebb küldhető üzenet a BGP fejrészt tartalmazza adat rész nélkül vagy 19 byte.

1.3.1 Üzenet fej formátum

MARKER(16 BYTE)	LENGTH (2 BYTE)	TYPE (1 BYTE)

Marker: Egy olyan értéket tartalmaz, melyet az üzenet vevője meg tud becsülni. Ha az üzenet típusa OPEN vagy az üzenet authentikálatlan üzenetet hordoz a marker értéke csupa egyes. Egyébként az értéke becsülhető bizonyos authentikációs mechanizmusokra alapuló számításokkal. A marker alkalmas szinkronizációs problémák detektálására és a beérkező BGP üzenetek authentikálására.
Length: a teljes hossz, mely 19-4096 byte lehet.
Type:

- open (BGP kapcsolat kezdeményezése)
- update (útvonal információ cseréhez)
- notification (hibaüzenetek küldésére)
- keepalive (ha nincs információ csere, akkor is tudni kell, hogy a szomszéd él)
- route-refresh (teljes útválasztási adatbázis küldése)

1.3.2 OPEN üzenet formátum

Az üzenet szolgál a BGP útválasztók bemutatkozására. Ezt adják át elsőként a TCP kapcsolaton.

VERSION (1 BYTE)	MY AS NUMBER (2 BYTE)	HOLD TIME (2 BYTE)	BGP IDENTIFIER (4 BYTE)	OPT PAR LEN (1 BYTE)	OPT PAR

Version: BGP verzió száma. Ez a jelenlegi BGP esetén 4.
My AS Number: Küldő autonóm rendszer azonosítója (száma).
Hold Time: Milyen időközönként (másodpercek száma) kell a szomszédtól üzenetet kapni, hogy élőnek tekintse. Azaz mennyit várjon az üzenet vevője, míg a forgalomirányító újra életjelet küld magáról valamilyen üzenet formájában.
BGP identifier: A küldő BGP azonosítója.
Opt Par Len: Opcionális paraméter mező teljes hossza.
Opt par: Opcionális paraméterek listáját tartalmazza, melyben minden paraméter egy <Paraméter típus; Paraméter hossz; Paraméter érték> hármassal van kódolva.

1.3.3 UPDATE üzenet formátum

Ezek az üzenetek szolgálnak a BGP partnerek közötti forgalomirányítási információ továbbítására. Az UPDATE csomagokban szereplő információ szükséges azon gráf megkonstruálásához mely leírja az autonóm rendszerek közötti kapcsolatokat. Ezt az üzenetet használja a BGP egy lehetséges útvonal hirdetésére, illetve a nem lehetséges útvonalak kivonására is.

UNF. ROUTES LENGTH (2 BYTE)	WITHDRAWN ROUTES (VÁLTOZÓ)	TOTAL PATH ATTR. LENGTH (2 BYTE)	PATH ATTRIBUTES (VÁLTOZÓ)	NLRI (VÁLTOZÓ)

Unfeasible routes lenght: kivont utak mező hossza. A 0 jelenti, hogy nincs ilyen út, azaz a WITHDRAWN ROUTES mező az UPDATE üzenetben üres.
Withdrawn routes: A kivont útvonalak IP címének előtagjainak listáját tartalmazza, minden ip cím előtagot a következő alakban kódolva: < előtag hossz; előtag >.
Total Path Attribute lenght: Path attributes mező teljes hossza.
Path attributes: Útvonal attribútumok listája. Minden útvonal attribútum a következő hármassal írható le: <attribútum flag; attribútum hossz; attribútum érték>.
NLRI: A hirdetett hálózatokat sorolja fel a következő formában:
1 byte előtag hossz (subnet maszk hossza) + 1 byte előtag (hálózat azonosítása)

1.3.4 KEEPALIVE üzenet formátum

A BGP nem használ semmilyen szállítási protokoll alapú mechanizmust arra, hogy meghatározza a partner elérhető-e. Helyette KEEPALIVE üzenetek cserélődnek a partnerek között elég gyakran ahhoz, hogy ne járjon le a hold-time időzítő. Az indokolható leghosszabb idő a KEEPALIVE üzenetek között a hold-time intervallum egyharmadának kell lennie. KEEPALIVE üzenetek nem küldhetők gyakrabban másodpercenként.
Ez a BGP legegyszerűbb üzenete ezért formátumát tekintve nem más, mint a BGP üzenetek közös fejléce.

1.3.5 NOTIFICATION üzenet formátum

Ezt az üzenetfajtát hibadetektálás esetén küldik el a hálózaton, amely után a BGP kapcsolat azonnal bezárásra kerül.

ERROR CODE (1 BYTE)	ERROR SUBCODE (1 BYTE)	DATA (VÁLTOZÓ)

Error code(hiba kód): *Error subcode (hiba alkód)*

1. Message Header Error
    a) A kapcsolat nem épült fel
    b) Rossz üzenethossz
    c) Rossz üzenettípus
2. OPEN Message error
    a) Nem támogatott verziószám
    b) Bad peer (egyenrangú) AS
    c) Rossz BGP azonosító
    d) Nem támogatott opcionális paraméter
    e) Authentikációs hiba
    f) Nem elfogadható hold-time intervallum
3. UPDATE Message Error
    a) Rosszul formázott attribútum lista
    b) Nem felismerhető Well-Known attribútum
    c) Hiányzó Well-Known attribútum
    d) Attribútum flag hiba
    e) Attribútum hossza rossz
    f) Érvénytelen Origin attribútum
    g) Forgalomirányítási hurok
    h) Érvénytelen NEXT_HOP attribútum
    i) Opcionális attribútum hiba
    j) Érvénytelen hálózat mező
    k) Helytelen AS_PATH

4. Hold Timer Expired — Türelmi idő lejárt
5. Finite State Machine error — Váratlan esemény következett be
6. Cease Error — Kapcsolat lezárása következik

Az utolsó három hibakód esetén alkód nem használható.

A hibakezelés részleteit az 5. alfejezet (Hibakezelés) tartalmazza.

1.3.6 ROUTE-REFRESH üzenet formátum

Ez az üzenet arra kéri a másik felet, hogy küldje át teljes útválasztási táblázatát. Tartalma mindösszesen egy címcsalád-azonosító (AFI), egy foglalt, 0 értékű bájt és egy címcsalád-részazonosító (SAFI). IPv4 esetén az AFI és SAFI értéke 1.

1.4 Útvonal attribútumok

Az útvonalak mellett a frissítési üzenetek fontos részét képezik az útvonal attribútumok.

JELZŐK (1 BYTE)	KÓD (1 BYTE)	HOSSZ(1 v. 2 byte attól függően, hogy a negyedik jelző 0 vagy 1)
TULAJDONSÁG ADATAI		

A jelzők a tulajdonság jellemzőit határozzák meg:

Bitek	Jelentés
7	Ha értéke 1, a tulajdonság értelmezése nem kötelező
6	Ha értéke 1, az attribútum továbbadása kötelező, még ha az útválasztó nem is képes értelmezni
5	Ha értéke 1, a tulajdonság csak részleges adatokat tartalmaz
4	Ha értéke 1, a tulajdonság hossza 2 bájtot foglal el
3-0	Foglalt, küldéskor 0-val kell feltölteni, fogadáskor pedig figyelmen kívül hagyni

A kód mező azonosítja magát a tulajdonságot, melyek a következők lehetnek:

1. ORIGIN

Kötelező, továbbítandó attribútum. Azt tárolja, hogyan értesült a BGP útválasztók hálózata először az útvonalról. Értékét az első BGP útválasztó állítja be, amely az útvonalat a többiekkel közli.

Lehetséges értékei:

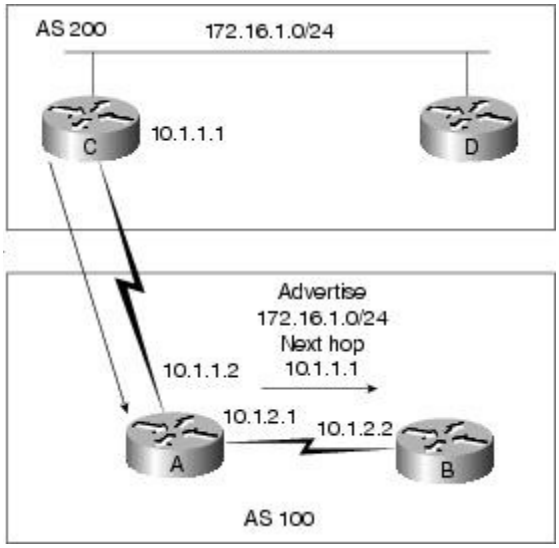
- IGP
- EGP
- Egyéb módon tanulja meg a forgalomirányító, valamilyen ismeretlen forrásból.

2. AUTONOM SYSTEM PATH

Kötelező attribútum. Autonóm rendszerek számának sorozatát tartalmazza, amelyen a forgalomirányítási információ áthalad. Nagy előnye, hogy a forgalomirányítási hurkok igen hamar kiderülhetnek.

3. NEXT HOP

Kötelező attribútum. Megadja azt az útválasztót, mely az útvonalakon a következő állomás lesz. Ez általában megegyezik azzal, amely az üzenetet küldte, de el is térhet.



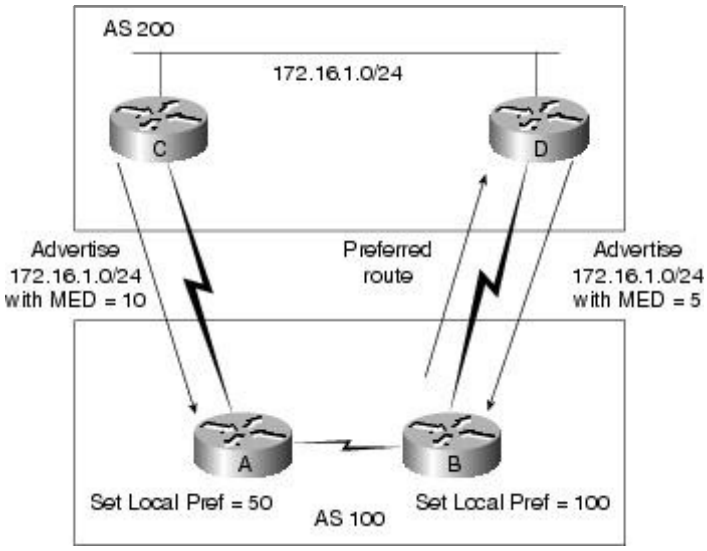
4. ábra

A C forgalomirányító hirdeti a 172.16.1.0/24 hálózatot 10.1.1.1 következő ugrással. Amikor az A útválasztó propagálja ezt az útvonalat a next-hop információ megőrződik, és ha a B forgalomirányítónak nincs útvonal elérési információja a next-hop-ra vonatkozóan, az

útvonal eldobódik. Ezért fontos, hogy fusson valamilyen IGP az AS en belül, hogy terjessze a next-hop elérési információkat.

4. MULTI_EXIT_DISC

Opcionális attribútum. Ha egy autonóm rendszer egy másikkal több ponton kapcsolódik, akkor megmondja, hogy melyik a preferált pont. Röviden MED–nek nevezzük és lehetővé teszi, hogy az útválasztók tetszési sorrendet állíthassanak fel a különböző, egyező végpontú útvonalak között.

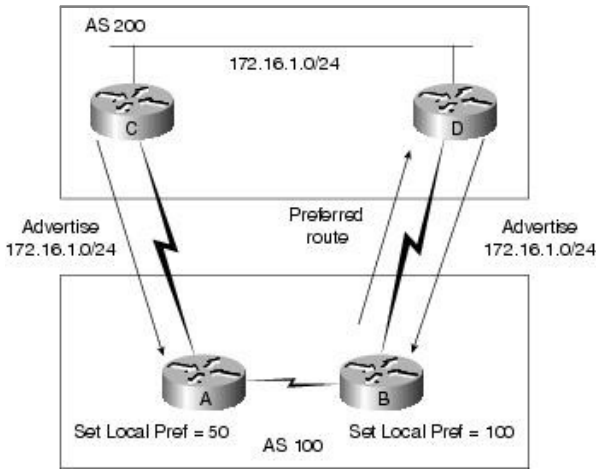


5. ábra

A C forgalomirányító hirdeti a 172.16.1.0/24 –es hálózatot 10–es metrikával, miközben a D forgalomirányító szintén hirdeti ugyanezt a hálózatot 5-ös metrikával. A kisebb metrikájú útvonal a preferált, tehát az AS100 a D felé vezető útvonalat fogja választani a 172.16.1.0 –ás hálózat elérésére.

5. LOCAL_PREF

Míg a MULTI_EXIT_DISC attribútum az autonóm rendszerek közti útvonalak kiválasztásában segít, ez az attribútum ugyanezt teszi az AS-en belül. A hálózatfelügyelők helyi prioritási értékeket rendelhetnek a külső útvonalakhoz, a BGP útválasztók pedig ezeket is hozzáfűzik majd az AS-en belülre küldött frissítési üzenetekhez. Döntési helyzetben a nagyobb lokális preferenciával rendelkező útvonal lesz kiválasztva.



6. ábra

Az AS100 két hirdetést kap a 172.16.1.0 –s hálózat elérésére az AS200 -ból. Amikor az A forgalomirányító megkapja ezt a hirdetést, ellátja egy 50-es lokális preferencia értékkel. A B ugyanígy tesz csak 100 lokális preferencia értékkel. Ezek a preferencia értékek kicserélődnek az A és B forgalomirányítók között és megállapításra kerül, hogy a magasabb preferenciával rendelkező B forgalomirányító lesz a kilépési pont az AS100-ból a 172.16.1.0 AS200-beli hálózat felé.

6. ATOMIC_AGGREGATE

Opcionális attribútum. Amelyik forgalomirányító ezt küldte, ott útvonal összevonás történt, ezért információ veszett el, nem egy teljes, hanem egy összegzett hálózat látszik.

7. AGGREGATOR

Opcionális attribútum. Annak a forgalomirányítónak az azonosítója, aki az útvonal összefogást elvégezte.

8. COMMUNITY

Mindenképp továbbítandó attribútum. Tehát értelmezési probléma esetén is tovább adódik. Ez a tulajdonság az útválasztási adatokat felhasználók meghatározott közösségével hozza összefüggésbe. Ezek a közösségek általában valamilyen közös tulajdonsággal rendelkeznek, a hozzájuk tartozó útvonalak megjelölése pedig segít e tulajdonság felismerésében és a megfelelő kezelés megvalósításában. A tulajdonság adatait négybájtos közösségi értékek alkotják. Ezeket bármely AS önállóan határozhat meg és tetszés szerint értelmezhet.

9. ORIGINATER_ID

Nem kötelezően továbbítandó attribútum. Ez az egyik olyan tulajdonság, amely segít az útvonaltükrözőknek a ciklusok elkerülésében. Ha egy BGP forgalomirányító egy útvonalat tükröz, mindig beállítja ezt az tulajdonságot, amely az útválasztó BGP azonosítóját tartalmazza. Ha ezután az útvonaltükröző újra megkapja az üzenetet, ebből az értékből már tudja, hogy ismeri az útvonalat, így elveti.

10.CLUSTER_LIST

A másik attribútum, mely segíti az útvonaltükrözők működését. Tartalma azon fürtök listája, amelyeken az útvonal már áttükröződött. Ha egy olyan útválasztóhoz jut el az útvonal, amely saját fürtjét látja a listában, elveti.

1.5 Hibakezelés

A fejezet a BGP üzenetek feldolgozása közben detektált hibák kezelését írja le. A már említett lehetséges hibák előfordulása esetén egy NOTIFICATION üzenet a megfelelő hibakóddal, hiba alkóddal és adat mezővel elküldésre kerül és a BGP kapcsolat lezárul. A BGP kapcsolat lezárása azt jelenti, hogy a szállítási protokoll kapcsolat lezárul és a BGP kapcsolat összes erőforrása felszabadul. A forgalomirányítási tábla távoli partnerhez rendelt bejegyzési érvénytelenként lesznek megjelölve.

1.5.1 Hibakezelés az egyes üzenetformátumok esetén

1.5.1.1 Message Header

Az üzenet fejrész feldolgozása alatt bekövetkező minden hiba „Message header error” hibakódú NOTIFICATION üzenet kiváltását eredményezi melynek hiba alkódját a hiba jellege specifikálja. A marker mező már említett értékei közül (OPEN üzenet vagy authentikációs mechanizmusfüggő) ha egyik sem teljesül szinkronizációs hibáról beszélünk és a hiba alkód értéke „kapcsolat nem szinkronizált (Connection not Synchronized)” lesz.

Ha az üzenetben ez egyes hosszértékek nem egyeznek, vagy meghaladják pozitív, ill. negatív irányban a hossz mező lehetséges határait a hiba alkódja „rossz üzenet hossz (Bad Message Length)” lesz.
Ha a típusmező értéke nem ismert, a hiba alkód „rossz üzenet típus (Bad Message Type)”.

1.5.1.2 Open Message

OPEN típusú üzenet feldolgozása közben fellépő hiba „OPEN message error” hiba kódú NOTIFICATION üzenetet vált ki. Lehetséges alkódok:

- Verzió szám nem támogatott: „Unsupported Version Number”
- AS mező elfogadhatatlan: „Bad Peer AS”
- Hold time mező elfogadhatatlan: „Unacceptable Hold Time”
- BGP identifier mező szintaktikailag helytelen: „Bad BGP identifier”
- Valamelyik opcionális paraméter nem felismerhető: „Unsupported Optional Parameters”
- Authentikációs eljáráshiba: „Authentication Failure”

1.5.1.3 Update Message

UPDATE üzenetek feldolgozása alatt előforduló hiba esetén a NOTIFICATION üzenet hiba kódja „Update Message Error”.
Ezen üzenetek hibaellenőrzése az útvonal attribútumok vizsgálatával kezdődik.
Lehetséges alkódok:

- Ha az „Unfeasible Routes Length” vagy a „Total Attribute Length” értéke túl nagy akkor az alkód: „Malformed Attribute List”
- Ha a felismert attribútum rendelkezik flaggekkel, amelyek ütköznek az attribútum típus kódjával: „Attribute Flags Error”
- Ha a felismert attribútum hossza nem egyezik az elvárt hosszal: „Attribute Length error”
- Ha bármilyen kötelező közismert attribútum nincs jelen: „Missing Well-known Attribute”

- Ha bármilyen kötelező, továbbítandó attribútum nem felismerhető: „Unrecognized Well-known Attribute”
- ORIGIN attribútum nem definiált: „Invalid Origin Attribute”
- NEXT_HOP attribútum szintaktikailag inkorrekt: „Invalid NEXT-HOP attribute”
- AS_PATH attribútum szintaktikailag helytelen: „Malformed AS_PATH”
- Opcionális attribútumok értékeinek ellenőrzése közben fellépő hiba: attribútum eldobása és „Optional Attribute Error” alkód.
- Többször előforduló attribútum: „Malformed Attribute List”
- NLRI mező szintaktikailag helytelen: „Invalid Network Field”

1.5.1.4 NOTIFICATION Message

Ez egy nagyon speciális eset, amikor a hiba jelzésére szolgáló üzenetek feldolgozása közben következik be hiba, de természetesen előfordulhat. Bármilyen nem felismerhető hiba vagy alhiba kód esetén ezt fel kell ismerni, naplózni és jelezni a partner adminisztrációja számára, de ennek a problémának a kifejtése még szabvány szinten sem szerepel.

1.5.2 Visszatartási időzítő lejártának kezelése

Ha a rendszer nem kap sikeres KEEPALIVE és/vagy UPDATE és/vagy NOTIFICATION üzeneteket az OPEN üzenet Hold Time mezőjében rögzített idő intervallumon belül „Hold Timer Expired” hibakódú NOTIFICATION üzenet kerül kiküldésre és a BGP kapcsolat bezárul.

1.5.3 Megszakítás

Bármikor előfordulhat olyan eset is, amikor bármilyen kritikus, detektált hiba nélkül a BGP partner úgy dönt, hogy bezárásra készteti BGP kapcsolatát, ha annak speciális oka van. Ezt egy „Cease” hibakódú NOTIFICATION üzenet formájában tudja megtenni.

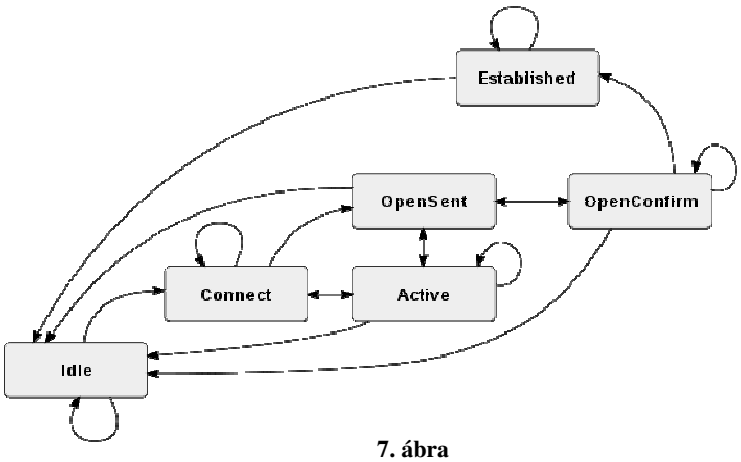
1.5.4 Kapcsolatütközés detektálás

Amikor a BGP partnerek egy időben próbálják kiépíteni a TCP kapcsolatot egymás közt ütközésről beszélünk. Ebben az esetben az egyik kapcsolat megszakításra kerül. Ennek kezelése során a BGP azonosítóra alapozva megállapításra kerül, hogy mely kapcsolat élvez védelmet az ütközés előfordulása esetén. A konvenció az, hogy az ütköző BGP partnerek BGP azonosító értékei közül a nagyobb értékkel rendelkező kapcsolat marad aktív.

1.6 FSM fogalma, BGP állapotok

Annak érdekében, hogy a BGP partnerek döntéseket tudjanak hozni működésük során egy úgynevezett véges állapotú gépet (Finite State Machine, FSM) használnak, melynek hat különböző állapotát különböztetünk meg:

- Idle
- Connect
- Active
- OpenSent
- OpenConfirm
- Established



7. ábra

Mindegy egyes peer-to-peer kapcsolathoz a BGP implementáció fenntart egy állapotváltozót, mely nyomon követi, hogy a kapcsolat éppen melyik állapotban van, továbbá a protokoll definiál üzeneteket, melyeket minden partnernek ki kell cserélnie az állapotváltás kezdeményezésére. Nézzük az egyes állapotokat részletesen:

Idle:

- Erőforrások lefoglalása a BGP folyamat számára
- TCP kapcsolat kiépítésének kísérlete a konfigurált partnerrel
- TCP kapcsolat figyelése
- Állapotváltás CONNECT-re

- Bármilyen hiba esetén a BGP kapcsolat azonnal terminálódik és az állapot visszakerül IDLE-be.
- Néhány ok, ami miatt a forgalomirányító nem hagyja el az IDLE állapotot:
    - 179-es TCP port nincs megnyitva
    - 1023 feletti véletlen TCP port nincs megnyitva
    - Szomszéd címe helytelenül van konfigurálva valamelyik forgalomirányítón
    - AS szám helytelen konfigurációja

Connect:

- Várakozás a sikeres TCP kapcsolat kiépülésére
- Ebben az állapotban kevés időt tartózkodik a BGP ha a TCP kapcsolat sikeresen kiépül.
- OPEN message küldése a partnernek és állapotváltás OpenSent-be.
- Hiba előfordulása esetén a BGP ACTIVE állapotba kerül. Ennek lehetséges okai ugyanazok, mint az IDLE állapot esetén.

Active:

- Ha a forgalomirányító nem képes a TCP kapcsolat felépítésére ebben az állapotban ragad.
- Új TCP kapcsolat kezdeményezése. Ha sikerül OPEN üzenet küldése a partnernek.
- Ha újra sikertelen az FSM visszaáll IDLE állapotba
- Az ismételt hibák egy ciklikus kört indítanak el az IDLE és az ACTIVE állapot között.

OpenSent:

- A forgalomirányító OPEN üzenetre várakozik a partnerektől.
- Az üzenet átvétele után ellenőrzi az OPEN üzenet érvényességét.
- Ha hiba fordul elő, mert az OPEN üzenetben szereplő valamely mező értéke nem egyezik a partner által várt értékkel (pl. Verziószám eltérés, MD5 jelszó eltérés, különböző saját AS szám stb.) a forgalomirányító NOTIFICATION üzenetet küld a hiba jellegével.
- Sikeres OPEN üzenetvétel esetén KEEPALIVE üzenet küldése, időzítők beállítása és állapotváltás OpenConfirm –ra.

OpenConfirm:

- A partner KEEPALIVE üzenetre várakozik
- Ha a KEEPALIVE üzenet megérkezett az időzítők lejárta előtt állapotváltás Established- re.
- Ha az időzítő lejárt a KEEPALIVE üzenet előtt vagy hiba fordult elő a forgalomirányító IDLE állapotba kerül.

Established:

- Ebben az állapotban a partnerek UPDATE üzeneteket küldenek a társaknak, melyekben a szomszéd felé publikált útvonalakat hirdetik.
- Hiba esetén NOTIFICATION üzenet és IDLE állapotba állás.

1.7 UPDATE üzenet kezelése

Mint az előző fejezetben említettük UPDATE üzenetek vétele csak ESTABLISHED állapotban lehetséges. UPDATE üzenet érkezése esetén elsőként minden mező érvényességének ellenőrzése következik a hibakezelésnél említett kritériumoknak megfelelően.

Nézzük mi is történik az UPDATE üzenetek feldolgozása során:

- Felismerhetetlen opcionális nem tranzitív (továbbítandó) attribútum esetén „csendes” ignorálás történik
- Felismerhetetlen opcionális tranzitív attribútum esetén paritás bit értékét az attribútum flag-ben 1-re állítja és továbbítja.
- Felismerhető érvényes értékkel rendelkező opcionális attribútum esetén az attribútum típusától függően megtörténik lokális feldolgozása, visszatartása, frissítése illetve esetleges propagálása más BGP partnerek felé
- Nem üres WITHOUT ROUTES mező esetén a tartalmazott útvonalak törlésre kerülnek az ADJ-RIB-IN lokális adatbázisból. Döntési folyamat futtatása, mivel az előzőleg hirdetett útvonalak többé nem elérhetőek
- Ha az üzenet lehetséges útvonalat tartalmaz, elhelyezésre kerül az ADJ-RIB-IN információs adatbázisba és a következő kiegészítő lépések történnek meg:
    - Ha az útvonal NLRI értéke azonos egy már ADJ-RIB-IN-ben tárolt útvonal NLRI értékével a régi útvonal felülírásra kerül.

- Ha az új útvonalnak egy régivel átfedése van döntési folyamat futtatása
- Ha az útvonal attribútumok azonosak az új és a régi útvonalak esetén és az új útvonal specifikusabb nincs további teendő.
- Ha az új útvonal NLRI értéke nem szerepel egyetlen, már tárolt útvonal esetén sem eltárolásra kerül az Adj-RIB-IN adatbázisban és döntési folyamat futtatása.
- Ha az új útvonal átfedésben van egy régi útvonallal döntési folyamat futtatása a kevésbé specifikus útvonal által leírt célok halmazára.

1.7.1 Döntési folyamat

Ez a folyamat a hirdetésekben szereplő útvonalak kiválasztását jelenti. Ez nem jelent mást, mint a helyi Policy Information Base (PIB)-ben szereplő szabályok alkalmazását az ADJ-RIB-IN-ben tárolt útvonalakra. A folyamat kimenete útvonalak egy olyan halmaza, amely hirdetésre kerül az összes partner számára és eltárolásra kerül a helyi ADJ-RIB-OUT adatbázisban. Tehát az adott AS-ben a célhoz vezető legjobb útvonal bekerül a BGP forgalomirányítási táblájába, kivéve egy esetet. Ellenőrzésre kerül, hogy az adott cél nem-e ismert más forgalomirányító protokoll által. Abban az esetben, ha szerepel, a forgalomirányító az adminisztratív távolság alapján dönt a célhoz vezető útvonalak forgalomirányító táblában történő elhelyezéséről. A kisebb adminisztratív távolsággal rendelkező útvonalat fogja elhelyezni a forgalomirányító táblába.
A döntési folyamat a következő 3 dologért felelős:

- Útvonalak kiválasztása a helyi AS-ben szereplő BGP partnereknek szánt hirdetések számára.
- Útvonalak kiválasztása a szomszédos AS–ekben szereplő BGP partnerekek szánt hirdetések számára
- Útvonal összefogás és útvonal információcsökkentés

A döntési folyamat 3 különálló fázisra bontható, melyek különböző események hatására váltódnak ki. Ezeket a fázisokat a következő 3 alfejezet írja le.

1.7.1.1 Preferencia értékének számítása

Az első fázisban a döntő függvény akkor fut le, amikor a helyi BGP partner UPDATE üzenetet kap a szomszédos AS-ben szereplő partnerétől, melyben egy új útvonalat, útvonal cseréjét vagy kivont útvonalat hirdet. A függvény működése alatt zárolja az ADJ-RIB-IN adatbázist és feloldja működése után.
Minden újonnan kapott vagy cserélt lehetséges útvonal esetén a helyi BGP partner meghatározza a preferencia értékét. Ha az útvonalat helyi AS-ben szereplő BGP partnertől kapta a LOCAL_PREF attribútum értéke lesz a preferencia értéke, vagy a helyi rendszer számolja ki a preferencia értéket előre konfigurált szabályozási információk alapján. Ha az útvonalat a szomszédos AS –ben szereplő partnertől tanulta akkor a lokális preferencia értéke az előre konfigurált szabályozási információk alapján számítódik. Ezt követően a lokális partner egy belső frissítési eljárást folytat, hogy kiválassza és hirdesse a leginkább preferált útvonalat.

1.7.1.2 Útvonalválasztás

A második fázis során az Adj-RIB-IN információs bázisban szereplő útvonalak vizsgálata következik, mely magában foglalja mind a helyi AS-ben mind pedig a szomszédos AS-ben szereplő partnerek által küldött útvonalakat. Amennyiben az útvonal NEXT_HOP attribútuma ismert, de a helyi partnernek nincs útvonala hozzá a LOC-RIB adatbázisban az útvonalat ki kell zárni a második fázisú döntési folyamatból. Minden egyes célhoz, amelyhez létezik útvonal az ADJ-RIB-IN adatbázisban a helyi forgalomirányító kiválasztja a legjobb útvonalat, mely a legmagasabb precedencia értékű útvonal lesz több oda mutató útvonal esetén vagy pedig az egyetlen útvonal az adott célhoz.
Ezt követően a helyi partner felveszi az útvonalat a LOC-RIB –be és lecseréli az ugyanazon célhoz vezető már LOC-RIB –beli útvonalakat. Meg kell határoznia a következő ugrást ahhoz a címhez mely a kiválasztott útvonal NEXT_HOP attribútumában szerepel. Oly módon, hogy végrehajt egy keresést és kiválaszt egy lehetséges útvonalat az AS-ben. Ha a NEXT_HOP attribútum által leírt címhez vezető útvonal megváltozik az útvonalválasztás újraszámítása

szükséges. A kivont útvonalak eltávolításra kerülnek a LOC-RIB–ből és az ADJ-RIB-IN–ből is.

1.7.1.3 Útvonalterjesztés

A harmadik fázisú döntési folyamat a második fázis végeztével, vagy a következő események előfordulásának hatására következik be:

- A helyi célokhoz vezető LOC-RIB -beli útvonalak megváltoznak
- a hirdetett helyileg generált útvonalak megváltoznak
- Új BGP kapcsolat kiépülése esetén

Minden LOC-RIB -beli útvonal feldolgozásra kerül és bekerül egy neki megfelelő bejegyzés az ADJ-RIB-OUT–ba. Az útvonal összefogás és az információcsökkentő technikák alkalmazása itt lehetséges. Az útvonalterjesztés általános leírását az 1.2.1 alfejezet tartalmazza.

2. Multiprotocol Label Switching (MPLS)

2.1 Az MPLS kifejlesztése

Az 1990-es évek elején felmerült az igény egy olyan protokoll létrehozására, amely lehetővé teszi IP-csomagok ATM hálózatokban történő továbbítását. Erre azért volt szükség, mert a felhasználói igényekkel a routerek routing-táblái folyamatosan növekedtek, és a hagyományos IP-forgalomirányítás esetén minden egyes routernek meg kell vizsgálni az IP-csomag fejrészét. Az elsődleges cél a forgalomtovábbítás egyszerűsítése volt.
Az IETF (Internet Engineering Task Force) 1993 és 1994 között két megoldást is kifejlesztett.

- Multiprotocol Encapsulation over ATM Adaptation Layer 5, amely az 1483-as RFC-ben van leírva. Ez a megoldás lehetővé tette bármely felsőbb rétegbeli protokoll ATM cellák segítségével történő továbbítását. Hátránya, hogy az ATM-felhőben előre kellett definiálni egy tunnel-t.
- Classical IP and ARP over ATM, RFC-1577. Ez a megoldás az ATM hálózatot logikai subnet-ekre (Logical IP Subnet, LIS) bontja, amelyen belül Virtual Circuit létrehozásával kommunikálnak az ATM-komponensek. A LIS-en kívülre a LIS egy routerén keresztül küldi a csomagokat. Minden LIS egy külön IP-alhálózat, és minden LIS-nek egy ATM ARP szervere van.

Egyik megoldás sem hozott sikert, így több gyártó is saját megoldással állt elő.

A Toshiba 1995-ben megpróbált egy úgynevezett ATM-Cell Switching Router-t az IETF-nél standardizálni. A CSR egy olyan ATM-kapcsoló, amely több IP-specifikus funkciót is ellát. Lényegében a CSR az ATM-kapcsolók és a routerek funkcióit egyesítené. Hátránya a nehéz megvalósíthatósága, ezért az IETF nem standardizálta.
A következő kísérlet az Ipsilon nevű cég IP-Switching megoldása 1996-ból. Ez sem lett standard, de kiadtak egy úgynevezett informational RFC-t.
Nem sokkal később (1996) a Cisco Systems nyilvánosságra hozta a Tag-switchinget, amelyet már nem csak ATM alapon, hanem egy sor további második rétegbeli technológiára implementáltak. A Cisco ezt a megoldást megpróbálta az IETF-nél standardizálni, de ezzel majdnem egy időben az IBM is kiadta az Aggregate Route-based IP Switching (ARIS) nevű megoldását, amely sok tekintetben hasonlított a Cisco megoldásához.

Ezért az IETF 1997-ben létrehozta az MPLS munkacsoportot egy szabvány létrehozására, melynél a fent említett megoldásokat is figyelembe vették.

2.2 Az MPLS előnyei a hagyományos IP forgalomirányítással szemben

A hagyományos IP forgalomirányítás esetén a hálózati eszközök a csomag célcím mezője alapján továbbítják a forgalmat a cél állomás felé, minden egyes csomóponton a forgalomirányítási tábla (routing table) alapján a legjobb metrikával rendelkező interfész irányába küldve a csomagokat. Tehát minden egyes hálózati eszköz minden egyes csomag esetén útvonalválasztási döntést hoz, amely az IP fejléc vizsgálatát is magában foglalja.
Az MPLS esetében csak az MPLS-felhő határánál lévő hálózati eszközök vizsgálják meg az IP fejrészt, a felhő belsejében lévő eszközök gyorsabb és ez által hatékonyabb címkekapcsolást végeznek.
De az MPLS nem csupán a gyorsabb forgalomtovábbításra képes, további előnye, hogy támogatja a VPN-eket és a Quality of Service-t.

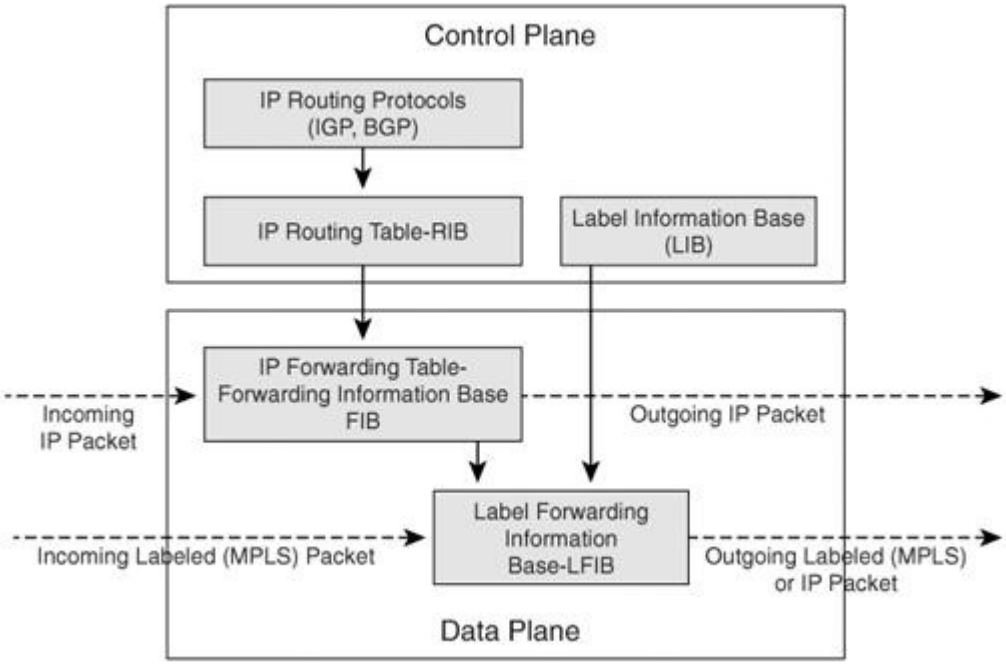
2.3 MPLS alapfogalmai, működése

Az MPLS megalkotásánál az alapötlet az volt, hogy a forgalomtovábbítást úgy tegye egyszerűbbé és gyorsabbá, hogy a hálózatba belépő csomagot egy címkével (label) látja el, így a hálózat belsejében lévő eszközök csak címkekapcsolást végeznek. Ez az adatkapcsolati rétegben (Layer 2) történik, így a csomagok fejrészének vizsgálata nem szükséges.
Az MPLS a 7 rétegű OSI modellben az adatkapcsolati és a hálózati, tehát a második és harmadik réteg között helyezkedik el.
Az MPLS-t úgy alkották meg, hogy mind az adatkapcsolati (ATM, Frame Relay, Ethernet, PPP), mind a hálózati (IPv4, IPv6, IPX, AppleTalk) rétegtől független legyen, tehát ne csak IP-protokollal és ATM technológiával legyen alkalmazható.

2.3.1 Control and Data Plane komponensek

Az MPLS esetén két fő funkcionális rész különböztethető meg:

- Vezérlő sík (control plane): tartalmazza a hálózati réteg forgalomirányítási információit és az ahhoz szükséges folyamatokat, illetve biztosítja a címkék kezelését, és cseréjét a szomszédos routerek között. Például: BGP, OSPF, RSVP és LDP.
- Továbbító vagy adat sík (data plane): feladata az adat csomagok továbbítása, amely lehet IP-csomag és felcímkézett csomag egyaránt. A továbbításhoz szükséges információkat, mint például a címke értékeket, a vezérlő síktól kapja.

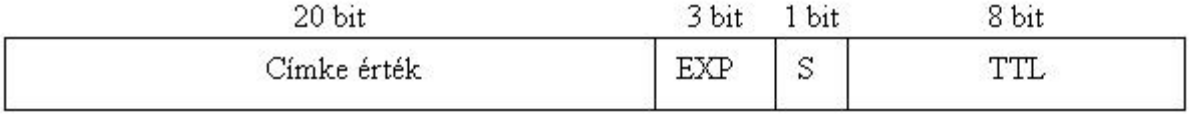


8. ábra

2.3.2 Az MPLS címke (label)

A címke az MPLS hálózaton belüli adattovábbításhoz szükséges fejrész.
Ez a címke a különböző technológiák esetén eltérő lehet. ATM esetén a VPI/VCI (Virtual Path Identifier / Virtual Channel Identifier) mezőbe kerül. Nem ATM esetén a harmadik rétegbeli csomag egy 32 bites címkét kap.

Az MPLS címke szintaxisa:



- 20 bit a címke értéke,
- 3 bit experimental field, Quality of Service használatához
- 1 bit bottom of stack indicator, a címke verem alját jelzi, ha több címkét használunk
- 8 bit Time-to-live field, a csomag „hátralevő életidejének” jelzése.

2.3.3 Label Switch Router (LSR)

Az LSR olyan router, amely támogatja az MPLS-t, két fajtáját lehet megkülönböztetni:

- Edge LSR: az MPLS határán lévő router, amely lehet ingress LSR, amikor egy még felcímkézetlen csomagot kap, ekkor a csomag elé beilleszti a kezdeti címkét, és lehet egress LSR, amikor a felcímkézett csomagról eltávolítja a címkét, és a csomag elhagyja az MPLS-t.
- Intermidiate LSR: az MPLS hálózat belsejében található eszköz, amely a felcímkézet csomagot feldolgozza, a címke és az irányítási tábla alapján kicseréli a címkét és a megfelelő interfészén továbbítja a csomagot az új címkével. Címkekapcsolást végez, lehet ATM kapcsoló, Frame Relay kapcsoló vagy router.

2.3.4 A csomagok címkézése

Amikor egy csomag eléri az MPLS hálózatot az ingress LSR-t, akkor az meghatározza a kezdő címkét, besorolja a csomagot. Ezután a többi LSR-nek már nem kell a besorolást végrehajtania, csak a címkekapcsolt útvonal szerint továbbítania. Az Edge LSR-ek a beérkező csomagokat ekvivalens átviteli osztályokba (Forwarding Equivalence Class - FEC) sorolják, amelyek a csomagok olyan halmazai, amelyeket az MPLS hálózaton belül azonosan kezelnek. Azokat a csomagokat, amelyeket azonos FEC-be sorol az Edge LSR, azonos címkével fogja ellátni.
A hálózat belsejében a címkék szétosztása történhet:

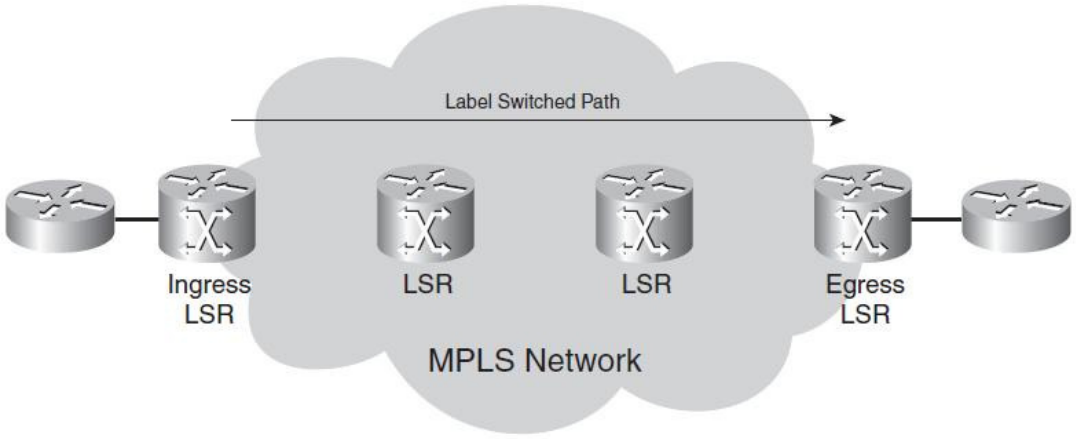
- Az IETF által kidolgozott Label Distribution Protocol (LDP) segítségével. RFC 3031
- A BGP-t kiterjesztették a címke információk terjesztésére.
- Resource Reservation Protocol -Traffic Engineering (RSVP-TE) használatával, RFC-3209 írja le.

- Tag Distribution Protocol (TDP): a Cisco fejlesztette ki, gyártóspecifikus, más gyártók eszközeit nem támogatja.

Az IETF az LDP kifejlesztésénél a Cisco Tag Distribution Protocol-ját használta alapként, a két protokoll nagyon hasonló funkciókkal rendelkezik, de az LDP egy gyártófüggetlen protokoll.

2.3.5 Label Switched Path (LSP)

A címkekapcsolt útvonal, vagyis LSP az LSR-ek egy szekvenciája, egy útvonal az MPLS-felhő egy részén át. A címkézet csomagok ezen az útvonalon továbbítódnak. Az LSP-k mindig egy irányba mutatnak, az LSR-ek fordított sorrendű szekvenciáját egy másik LSP határozza meg.



9. ábra

2.4 Label Distribution Protocol (LDP)

A címke kiosztó protokoll az LSR-ek közötti címkekapcsolt útvonalak (LSP) beállításáért és kezeléséért felelős protokoll. Az IETF által kifejlesztett protokoll, amely a 3036-os RFC-ben van leírva.

Azt a két LSR-t amelyek LDP-t használnak a címke és FEC információk cseréjéhez LDP peereknek nevezzük, közöttük egy LDP session van, és UDP illetve TCP protokoll (a 646-os porton) segítségével kommunikálnak.

Az LDP üzeneteknek négy kategóriája létezik:

- Discovery üzenetek: az LSR-ek jelenlétének hirdetéséhez, és fenntartásához. Az LSR-ek bizonyos időközönként Hello üzeneteket küldenek UDP csomagok formájában. Két Discovery mechanizmust különböztethetünk meg, az alap (basic) és a kiterjesztett (extended) discovery mechanizmust. Az alap discovery a közvetlenül csatlakoztatott szomszédok felfedezésére szolgál, ezért Link-Hello üzenetnek nevezik, amelyet egy multicast üzenet formájában az alhálózatban lévő összes többi LSR-nek címez. A kiterjesztett Discovery mechanizmus esetén csak egy meghatározott IP-címre küldi a Targeted-Hello üzenetet.
- Session üzenetek: két LSR közötti session kiépítéséhez, fenntartásához és lebontásához. Ilyen például az Initalization üzenet. Először ezt az üzenetet küldi az aktív LSR a passzívnak, ez az üzenet minden információt tartalmaz, amely a session létrehozásához szükséges. Ha a passzív LSR megfelel a követelményeknek, akkor egy Keepalive üzenetet küld az aktívnak, amennyiben nem akkor egy Notification üzenetet.
- Advertisement üzenetek: a címkék és az ekvivalens átviteli osztályok közötti leképezés létrehozásához, módosításához és törléséhez. Például a Label-request vagy a Label-mapping üzenet. A Label-request üzenet küldője egy címkét kér az üzenet fogadójától, hogy egy csomagot a megfelelő címkével küldhessen tovább. Az üzenet fogadója meghatározza a Label-request üzenetben megadott FEC értékhez a címkét és egy Label-mapping üzenettel válaszol. A Label-release üzenetet pedig akkor küldi az LSR, ha már nincs tovább szükség a címkére.
- Notification üzenetek: a fellépő hibákról és a session állapotáról szolgáltat információkat. Ilyen például a Keepalive üzenet.

Az LDP egy címke kiosztó protokoll, amely nem képes erőforrások lefoglalására és így a Traffic Engineering támogatására sem. Ezért az IETF elkészítette az LDP kiterjesztését a Constraint-based Routed Label Distribution Protocol-t. A CR-LDP lehetővé teszi, hogy a bementi LSR hálózati erőforrásokat foglaljon le. Így a CR-LDP már nem csak a címkék kiosztására képes, hanem a címkekapcsolt útvonalak erőforrásigényeinek biztosítására is.

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

Az RSVP, ahogy a neve is mutatja, erőforrások lefoglalására szolgál, viszont kibővítették a címkekiosztás funkcióval is. Az RSVP-t a 2205-ös RFC írja le, és arra fejlesztették, hogy erőforrásokat foglaljon le IP-hálózatokban, ezzel Quality of Service funkciókat valósítva meg. Az RSVP-TE nagy előnye, hogy a címkekapcsolt útvonalakhoz (LSP-khez), vagyis az azonos FEC-be tartozó csomagok számára foglalja le az erőforrásokat.

Egy forrás és egy cél közötti erőforrások lefoglalásához az RSVP-TE két üzenetet használ, a PATH és a RESV üzenetet. Az ingress LSR egy PATH üzenetet küld az egress LSR-nek, útközben minden közbeeső LSR beilleszti a PATH üzenetbe a saját IP-címét, így az egress LSR ismerni fogja azt az útvonalat, amelyen keresztül hozzá érkeznek majd a csomagok. Az RSVP nagyon fontos tulajdonsága, hogy az adatfolyam fogadója határozza meg a lefoglalandó erőforrásokat, és nem a küldő. Ez azért fontos, mert a különböző fogadó felek különböző igényeket támaszthatnak a hálózattal szemben. Ezért ahelyett, hogy a forrás tartaná nyilván a fogadó felek erőforrásigényeit, sokkal hatékonyabb, ha a fogadó maga gondoskodik az igényeiről.

Tehát az ingress LSR PATH üzentére az egress LSR egy RESV üzenettel fog válaszolni, amely tartalmazza a szükséges erőforrások leírását. Miközben a RESV üzenet halad visszafelé, minden LSR ellenőrzi, hogy a kívánt erőforrások rendelkezésre állnak-e. Ha igen, akkor lefoglalja az erőforrásokat és továbbítja a RESV üzenetet, ha nem, akkor RESV_ERROR üzenettel válaszol és a folyamat ezzel félbeszakad.

Fontos még megjegyezni, hogy az erőforrás-igénylést életben tartsa a fogadó fél, bizonyos időközönként ezt a RESV üzenetet újra kell küldje, különben a lefoglalt útvonal törlődik.

2.6 Az MPLS működésének négy fő lépése

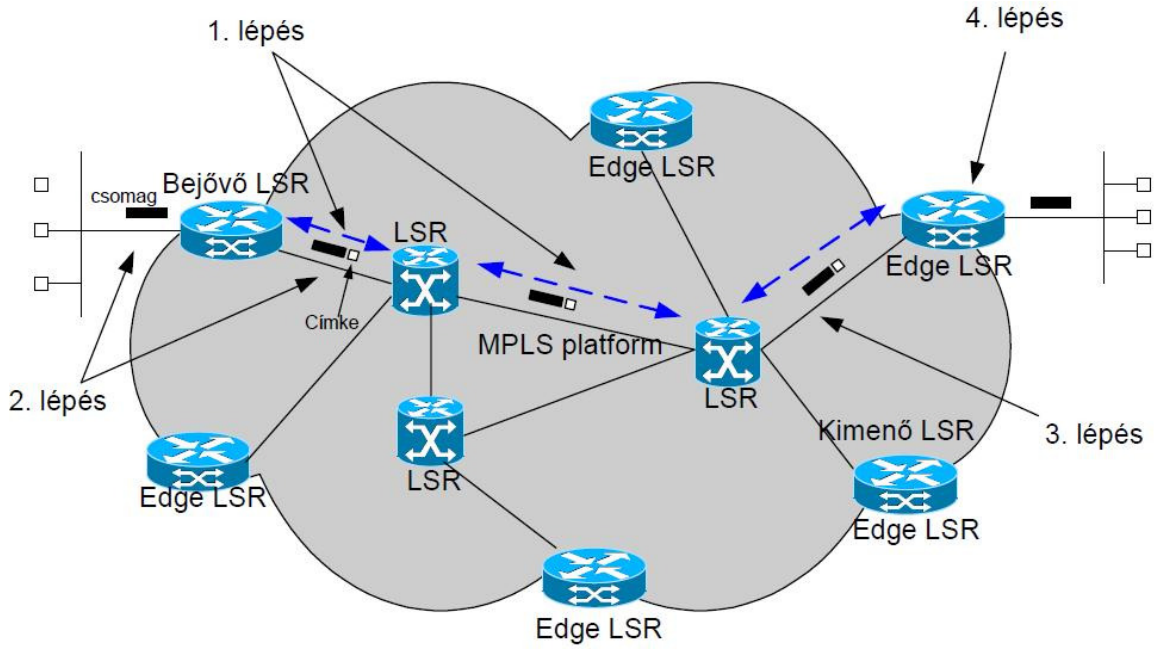
1. lépés: a hálózat feltérképezése valamely forgalomirányítási protokoll segítségével, például OSPF vagy BGP. Az LDP protokoll segítségével beállítódnak a címkék, így kialakulnak a címkekapcsolt útvonalak (LSP-k).

2. lépés: amikor egy csomag belép az MPLS hálózatba az ingress LSR, megvizsgálja a csomag fejrészét és meghatározza a címke értékét, és továbbítja a megfelelő belső LSR-nek.

3. lépés: a belső LSR-ek csak a címkét vizsgálják meg, a csomagot figyelmen kívül hagyják. Az irányítási táblában megkeresi a bemenő interfész és címke pároshoz tartozó kimenő

interfész és címke párost, majd a címkét kicseréli, és a kimenő interfészen továbbítja a csomagot az új címke értékkel.

4. lépés: az egress LSR, eltávolítja a címkét és feldolgozza a csomag fejrészét. A csomagot a hálózati cím alapján továbbítja.



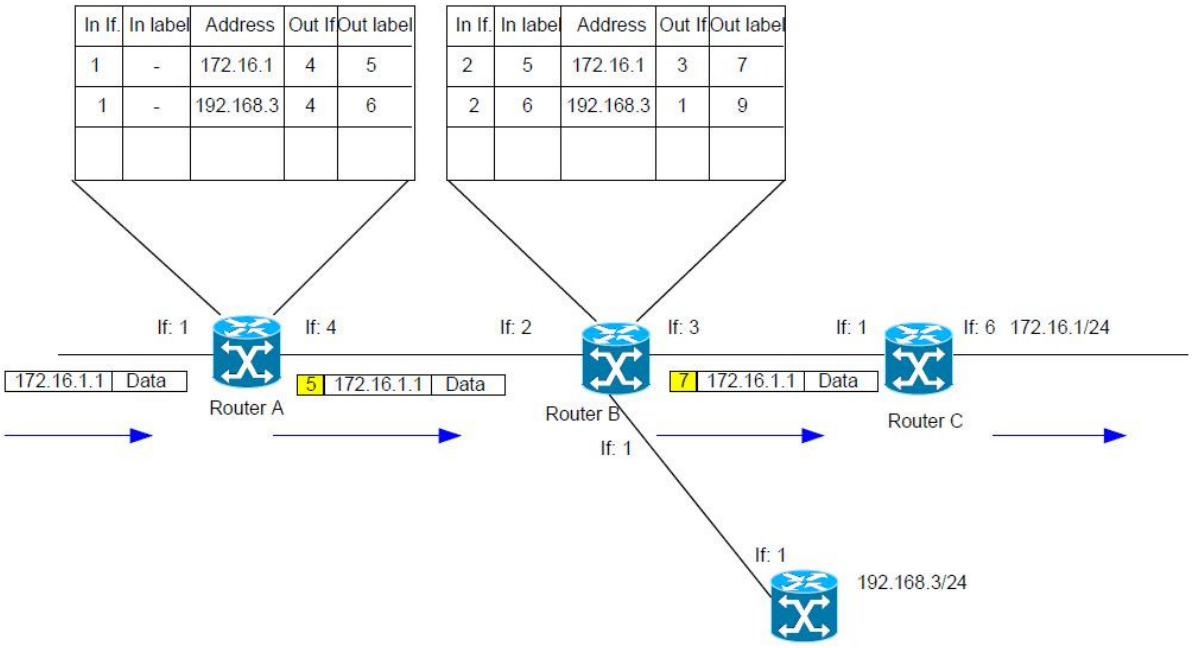
10. ábra

2.7 A címkék hatásköre, a címkék elosztása (distribution), és a címkék cseréje (swapping)

A címkéknek csak lokális jelentése van, ugyanis minden egyes LSR egyedileg kezeli a címkéket, így előfordulhat, hogy két LSR ugyanazt a címkét használja két különböző LSP esetén. Viszont lényeges, hogy az egyes LSR-párok egyeztessék a címkéket. Éppen ezért, hogy egy LSR ne kaphassa ugyanazt a címkét két forrástól, ezért mindig a célhoz közelebbi LSR határozza meg a címke értékét. A címke hirdetése történhet automatikusan a célhoz közelebbi LSR felöl (Unsolicited Downstream), illetve a forráshoz közelebbi LSR kérésére is (Downstream-on-Demand).

A címkék cseréje: Amikor egy LSR egyik interfészén egy MPLS csomagot kap, akkor megvizsgálja a csomag címkéjét, és megnézi a továbbítási táblájában, hogy az adott bemenő interfész és bemenő címke értékhez melyik kimenő interfész és címke érték páros tartozik.

Ezután az LSR kicseréli a csomag címkéjét a megfelelőre, és az adott interfészén továbbítja a csomagot.



11. ábra

2.8 A címke verem

Az MPLS megengedi, hogy egy csomag több címkét is kapjon, így jön létre a label stack vagy címke verem.

Layer 2 Header	Top Label	...	Bottom Label	Layer 3 Header

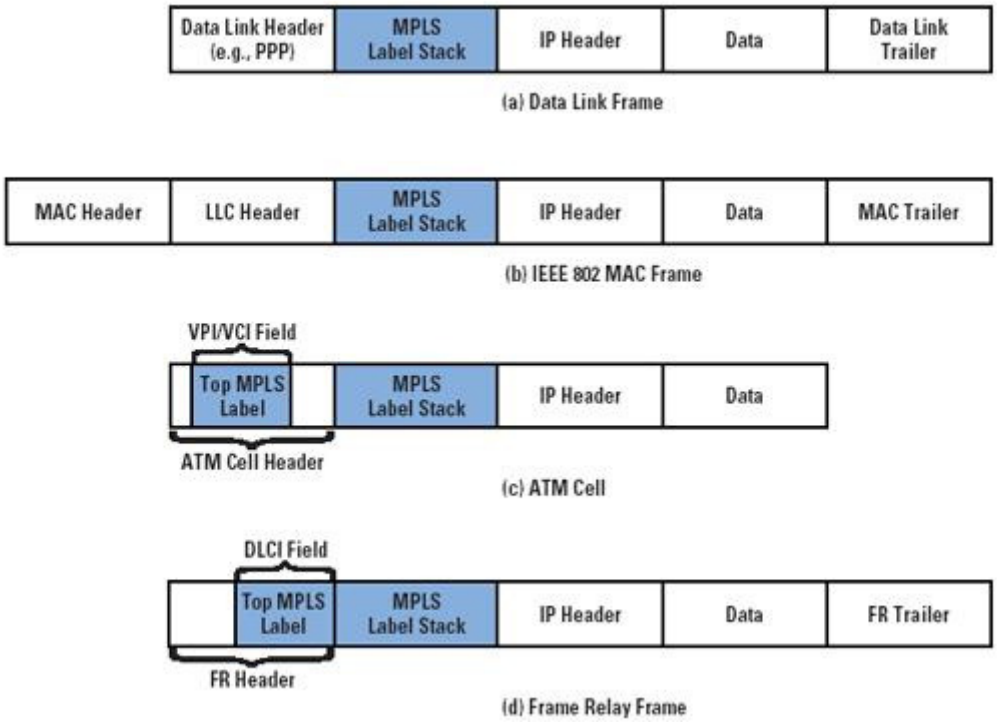
A 32 bites címke 24-edik bitje a bottom of stack jelzi, hogy az adott címke a verem legalsó címkéje-e. Ha értéke egy, akkor a címke az utolsó a veremben (a verem alja). A címke stackelést a 3032-es RFC írja le.

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

A címke stackelést MPLS VPN és traffic engineering, továbbá több önálló rendszeren (Autonom System - AS) történő forgalmazás estén alkalmazzák.

Az alábbi ábrán jól látható, hogy a beágyazás (enkapszuláció) során a címkeverem a hálózati rétegbeli csomagegység fejrésze és az adatkapcsolati rétegbeli keret fejrésze közé ágyazódik

be. Az is látható, hogy ATM és Frame Relay esetén kihasználható a keretek fejrészének sajátosságai.



12. ábra

2.9 Címke stackelés ATM és Frame Relay esetén

A cellakapcsolt ATM esetén a címkét a VPI/VCI (Virtual Path Identifier / Virtual Channel Identifier) érték adja, így a verem alja bit tárolása nem lenne lehetséges. Ezért az összes címke 32 bites formátumban a csomag elejére rakodik, és a csomag részeként továbbítódik. De továbbra is a legfelső MPLS címke adja a VPI/VCI értékét. Így az edge LSR-ek észlelik, hogy több címkét is kapott a csomag.

ATM alapú MPLS esetén az MPLS hálózat belsejében lévő LSR-ek ATM kapcsolók, melyek a cellakapcsolást változatlan módon hajtják végre, tehát csak továbbító sík (data plane) funkciókat látnak el. Ezért szükség van egy külső vezérlő sík komponensre, amely a vezérlő sík információk terjesztését biztosítja. Ezt a komponenst Label Switch Contoller-nek (LSC) nevezik. Bizonyos ATM LSR-ek képesek ellátni a vezérlő sík funkciókat is, így azoknál nincs szükség külső komponensre.

A csomagkapcsolt Frame Relay esetében is hasonlóan történik a címkevermek beágyazódása, a különbség csak annyi, hogy itt a legfelső címke a Frame Relay fejrész DLCI (Data-Link Connection Identifier) mezőjébe is bekerül.

Ez azért lehetséges, mert a DLCI és VPI/VCI értéknek is csupán helyi jelentése van, mint az MPLS címkéknek. Viszont sem az ATM sem a Frame Relay nem kezeli az ugrásértéket, azért a bemeneti LSR-nek kell csökkentenie ezt az értéket. Mivel a pontos érték sokszor nem áll rendelkezésére, ezért ilyenkor az ugrásszámot egyel csökkenti, így a teljes ATM vagy Frame Relay hálózatot csupán egyetlen hopnak érzékeli.

2.10 Penultimate Hop Popping

Az egress edge LSR-ek két lépést végeznek el minden csomag esetén. Először megvizsgálják a legfelső címkét és eltávolítják azt, aztán ha ez volt a legfelső MPLS címke, akkor az IP forgalomirányítási táblája alapján hoz döntést, ha pedig a címke verem tartalmazott további címkét, akkor ezt a címkét is megvizsgálja, kicseréli és továbbítja a csomagot.
A Penultimate Hop Popping módszer segítségével az egress edge LSR-ek feladata egyszerűsíthető, azáltal, hogy az LSP-ben az edge LSR-t megelőző LSR eltávolítja a legfelső címkét. Ezt természetesen az edge LSR-nek kell kérnie LDP-n keresztül az implicit-null címke segítségével. Ekkor az utolsó előtti LSR címkekapcsolási táblájában a kimenő címke értéke implicit null lesz, így az LSR tudja, hogy a címkét el kell távolítania.
Ez a módszer ATM és Frame Relay esetén nem alkalmazható, mivel a VPI/VCI illetve DLCI értékek nem hagyhatók el.

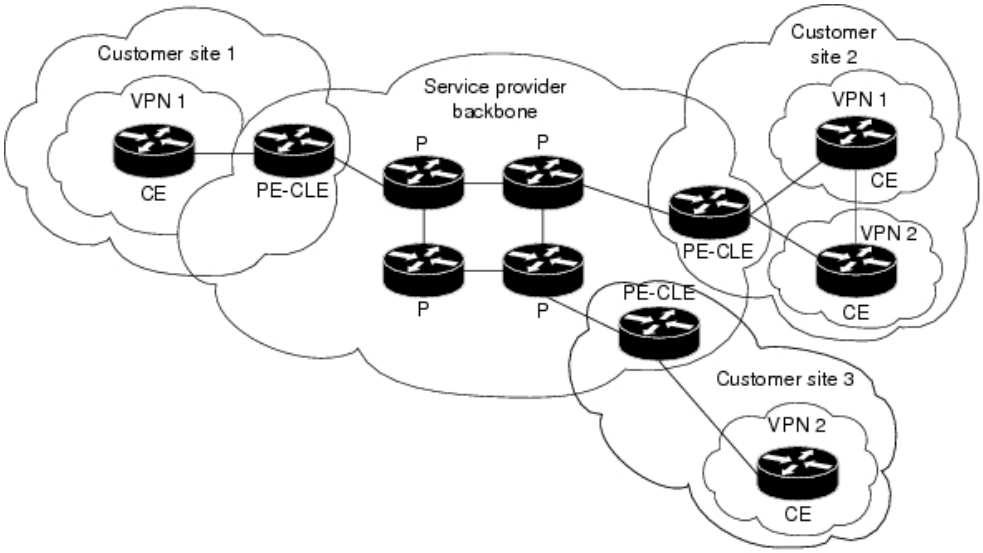
3. MPLS alapú virtuális magánhálózatok

3.1 VPN fogalma, megjelenése

A virtuális magánhálózat (Virtual Private Network, röviden: VPN) egy logikailag összetartozó, egységesen menedzselt hálózat, mely telephelyeinek összekötését egy általános célú csomagorientált átviteli hálózaton valósítja meg. A technológia lényege, hogy több, egymástól fizikailag különböző helyen lévő hálózatot kapcsolunk össze. Itt a különböző helyen érthetünk akár a világ két távoli pontján lévő hálózatot is.

A mindennapok során gyakran találkozhatunk a VPN-ekkel kapcsolatos fogalmakkal, olyan új termékekkel vagy szolgáltatásokkal, melyek VPN támogatottsággal rendelkeznek. Akár gondolhatnánk azt is, hogy a VPN koncepció valamifajta forradalmian új technológiai áttörés. Erről azonban szó sincs, a fogalom több mint 15 éves múltra tekint vissza a szolgáltatók piacán.

Az új technológiák ugyanakkor nagyobb biztonságot, valamint skálázhatóságot tesznek lehetővé, mindezt egyre hatékonyabb implementációs megoldásokkal. Nem meglepő tehát, hogy a VPN szolgáltatások iránti igény volt az egyik fő oka az MPLS technológiák elterjedésének és alkalmazásának a szolgáltatói és a nagyvállalati hálózatokban.



13. ábra

Az MPLS-en alapuló virtuális magánhálózatok nagy áttörést jelentettek a piacon, ugyanis ezzel sikerült egy igen költséghatékony, jól skálázható megoldást találni. Az MPLS technológia részletezése előtt azonban először tekintsük át a közönséges VPN terminológiát, a szolgáltatásait, a különböző felhasználási területeket és a leggyakrabban előforduló topológiákat.

3.2 VPN-ek kialakítása, osztályozása

A szolgáltatói hálózatokban az új technológiák megjelenésével, valamit napjaink új igényei miatt a VPN fogalom egyre összetettebbé vált. A különböző vállalatok olykor ellentmondásos fogalmakat vezettek be, ezzel még bonyolultabbá téve az egyébként sem mindig egyértelmű koncepciókat. Jelenleg a világ virtuális magánhálózataiban számos technológiát és topológiát alkalmaznak. Ahhoz, hogy hatékonyan meg tudjuk különböztetni a különböző VPN-eket, bevezetjük a következő osztályozást:

- Üzleti probléma, amelyet a VPN segítségével meg akarunk oldani. Ezek a problémák a következő kategóriákba tartozhatnak:
    - Intranet: egyazon vállalaton belüli hálózati kommunikáció
    - Extranet: vállalatok közötti kommunikáció megteremtése
    - VPDN (Virtual Private Dialup Network): mobil felhasználók hálózati hozzáféréseinek megteremtése

  A fenti három megoldás általában megtalálható a legtöbb topológiában és technológiai megoldásban, melyeket a szolgáltatók nyújtanak.

- Az OSI modell melyik rétegében történik a szolgáltató és az ügyfél közötti topológiára vonatkozó információk váltása. Itt a következő osztályozásról beszélhetünk:
    - Overlay model: a szolgáltató point-point (vagy multipoint) kapcsolatokat biztosít az ügyfél egyes telephelyei között.
    - Peer Model: a szolgáltató és az ügyfél 3. rétegbeli routing információkat vált egymással.
- A VPN szolgáltatáshoz használt 2. vagy 3. rétegbeli technológia. (mint például: X.25, ATM, Frame Relay, IP)

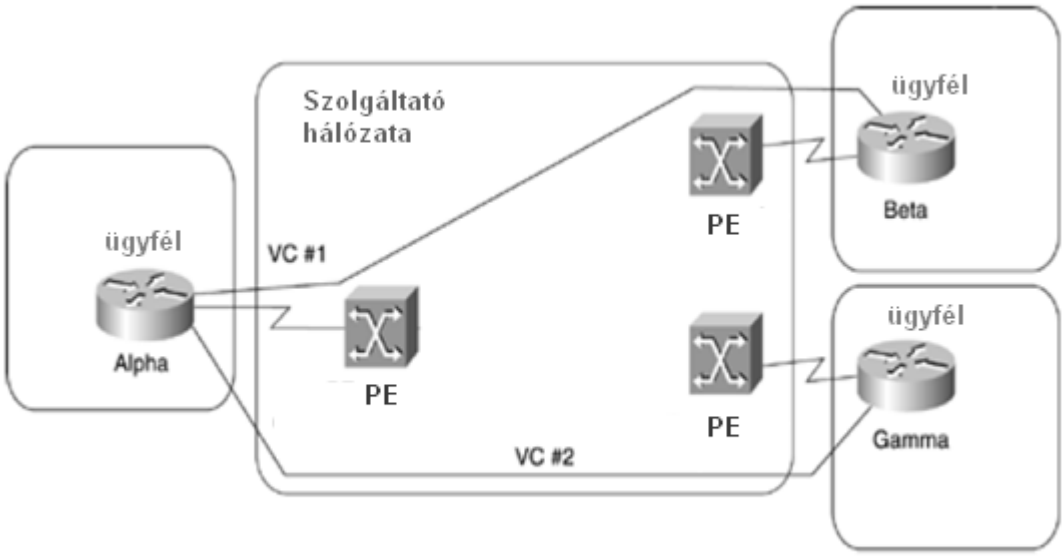
- A hálózat topológiája: az egyszerűbb Hub-and-Spoke topológiától kezdődően a teljesen összekapcsolt, valamint a többszintű hierarchikus nagyobb méretű hálózatok.

3.3 VPN Modellek

3.3.1 Overlay VPN Modell

Az Overlay modell talán a legkönnyebben megérthető, ugyanis igen jól elkülöníthetők a szolgáltató és ügyfél szerepkörei.

- Szolgáltató: emulált bérelt vonalak egy halmazát nyújtja, melyeket virtuális áramköröknek (VC, Virtual Circuit) nevezünk. Ezeknek egyik fajtája, amikor a virtuális áramkör tartósan fennáll: PVC (Permanent Virtual Circuit). A másik típusban az áramkör csak szükség esetén épül ki: SVC (Switched Virtual Circuit).



14. ábra

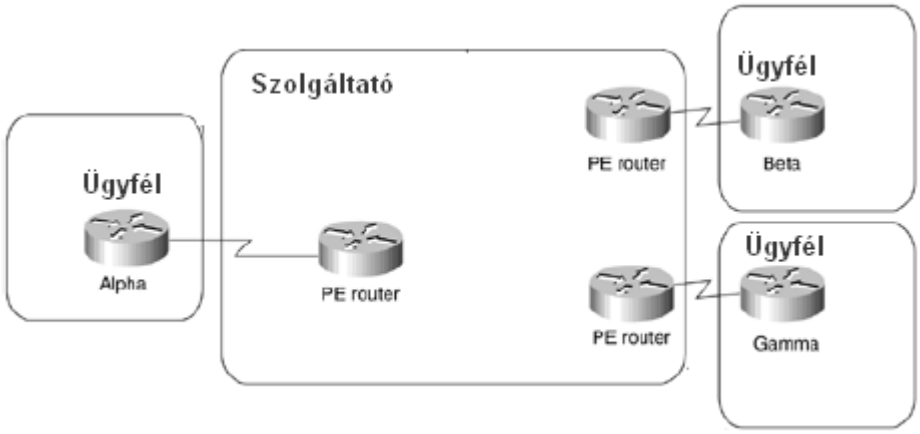
- Az ügyfél: a CPE (Customer Premises Equipment, Alpha, Beta és Gamma) routerek között létesít kapcsolatot virtuális áramkörökön keresztül, melyeket a szolgáltató biztosít számára. A routing protokollal kapcsolatos kommunikáció mindig az ügyfelek eszközei között zajlik, mindeközben a szolgáltatónak egyáltalán nincsen információja a hálózat belső felépítéséről.

A QoS-el szembeni elvárások az Overlay modellben általában egy bizonyos VC esetében megadott garantált sávszélességgel (Commited Information Rate, CIR) és a maximum sávszélességgel (Peak Information Rate, PIR) fogalmazhatóak meg.
Az Overlay VPN hálózatokat számos 2. rétegbeli WAN technológiával lehetséges implementálni, mint például: X.25, Frame Relay, ATM vagy SMDS.
Habár azt mondhatjuk, hogy ennek a modellnek viszonlag könnyű az implementációja, emellett néhány hátrányt mindenképpen meg kell említenünk:

- Jól használható abban az esetben, ha nem hibatűrő konfigurációkkal van dolgunk, kevés központi helyszínnel és egyéb telephelyekkel. Azonban nagyon bonyolulttá válik a helyzet egy sok kapcsolattal rendelkező hálózat esetén.
- A virtuális áramkörök sávszélességének helyes meghatározásához részletes ismeretekre van szükségünk az adatforgalmat illetően, amelyek általában nem állnak rendelkezésünkre.
- Az implementációt illetően a költségek gyakorlatilag egyenes arányban növekednek a létrehozott pont-pont kapcsolatok számával, nem a pedig a hálózatba kapcsolt telephelyek számával.

3.3.2 Peer-to-peer modell

A peer-to-peer modell 2000-es év környékén került bemutatásra, méghozzá olyan megfontolásból, hogy az Overlay modell hátrányait lehetőleg kiküszöböljék.



15. ábra

Ebben a modellben a PE (Provider Edge) eszközök szerepét már routerek töltik be, melyek a CPE routerekkel váltanak routing információkat. A CPE forgalomirányítókat két csoportba sorolhatjuk, ezek pedig a menedzselt és a nem menedzselt routerek. A peer-to-peer modell ugyanis lehetővé teszi szolgáltatói oldalról az ügyfél CPE forgalomirányítóinak a menedzselhetőségét.

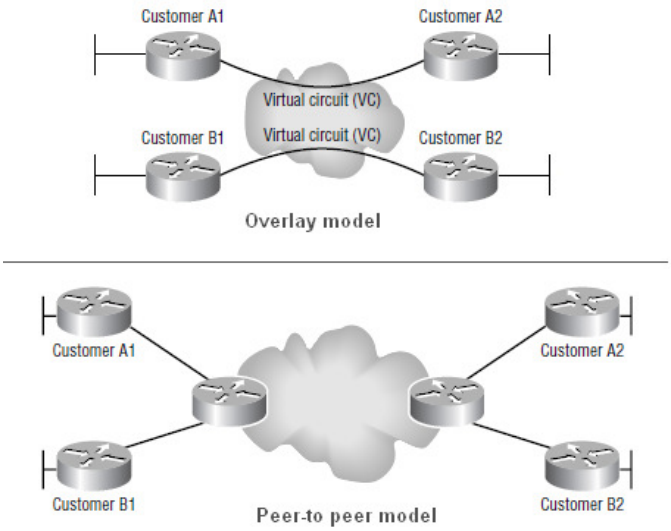
Az alábbiakban felsoroljuk a tradicionális Overlay modellhez képest az imént tárgyalt peer-to-peer modell előnyeit:

- Az ügyfél szemszögéből nézve a routing igencsak leegyszerűsödik, köszönhető mindez annak, hogy az ügyfél router (CPE) csak egy PE routerrel vált routing információkat. Az előző modellben a szomszédok száma igencsak megnövekedhetett.
- Az útvonalválasztás az ügyfél egyes telephelyi között mindig megfelelő, mivel a szolgáltató tisztában van az ügyfél hálózatának a topológiájával, így gondoskodni tud a helyes routing-ról.

- Az egyes helyszínek kívánt sávszélességeinek meghatározása jóval egyszerűbb, ugyanis elegendő döntenünk egyszerre csak egy adott site-hoz tartozó sávszélesség nagyságáról.

- Új telephely hozzáadása jelentősen leegyszerűsödik, ugyanis ebben az esetben a szolgáltatónak az új kapcsolat kiépítése után elegendő a megfelelő módosításokat megtennie a CPE routerhez kapcsolódó PE routeren. Ezzel szemben az Overlay modellben hasonló szituációban a szolgáltatónak virtuális áramkörök egész csoportját kellett beállítania, a szükséges telephelyekhez.

A következő ábra szemléletesen mutatja az előbbiekben tárgyat két modell közötti különbséget:



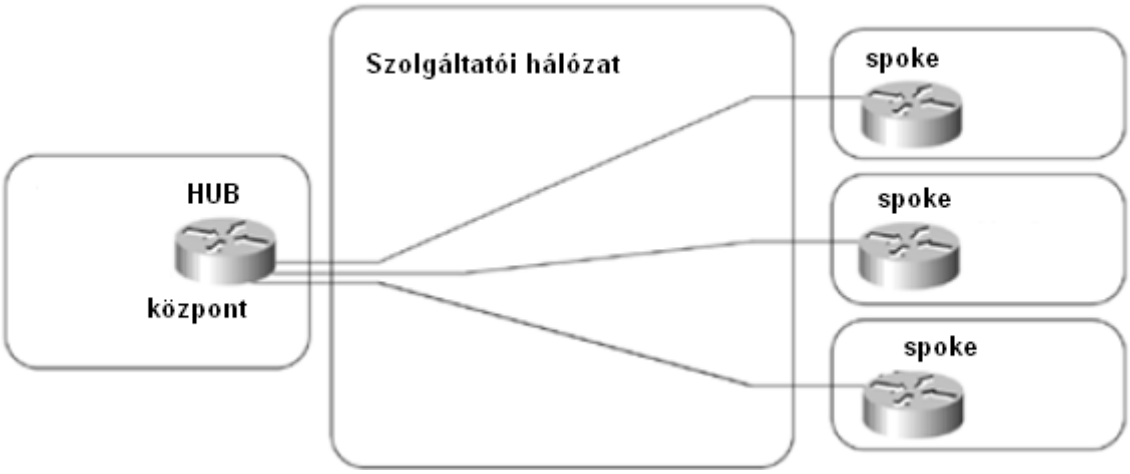
16. ábra

3.4 VPN Topológiák

Egy szervezet által használni kívánt topológiát a működésének mikéntje határozza meg, illetve, hogy melyik modell révén tudja a lehető leghatékonyabban végezni a tevékenységét. A valóságban az a legjellemzőbb, hogy nem egyedi topológiákkal találkozunk, hanem néhány jól ismert változatot alkalmaznak a különböző vállalatok, üzleti problémáiknak megfelelően. A következőkben tárgyalt VPN topológiák három fő kategóriába sorolhatók:

- Az Overlay VPN modell hatására kialakult topológiák, név szerint hub-and-spoke, részlegesen vagy teljesen összekapcsolt és hibrid topológiák.
- Extranet topológiák, melyek magában foglalják az any-to-any Extranet és a Central Services extranet topológiákat.
- Speciális topológiák, mintpéldául VPND gerinchálózat vagy központilag menedzselt topológiák.

3.4.1 Hub-and-spoke topológia



17. ábra

Ez az egyik legelterjedtebb, jelenleg is használatos topológia, amelynek a lényege, hogy több távoli iroda (spoke) kapcsolódik egy központi telephelyhez. A topológia köti meg, hogy a távoli telephelyek kommunikálhatnak-e egymással, azonban az esetek többségében ez lehetséges. Az adatforgalom nagy része a központtal bonyolódik le, a távoli irodák közötti

kommunikáció elhanyagolható. A topológiából adódik, hogy ilyenfajta kapcsolat csak a központon (hub) keresztül épülhet ki. Olyan vállalatok használják virtuális magánhálózataikban ezt a megoldást, amelyek felépítése szigorú hierarchiának felel meg. Például: kormányzatok, bankok vagy olyan nagy nemzetközi szervezetek, amelyek az egyes országokban csak kisebb irodákkal rendelkeznek. Megjegyezzük, hogy bizonyos esetekben bár egy másik megoldás jobban ki tudná szolgálni az ügyfél igényeit, anyagi vagy a felépítés egyszerűségét szem előtt tartó megfontolásból mégis ezt a topológiát preferálják.
Nem kérdéses, hogy a központi routernek (hub) kulcsfontosságú szerepe van az ily módon kialakított virtuális magánhálózatokban. Jogosan merülhet tehát fel az a kérdés, hogy miként kezelhető az a szituáció, amikor valamilyen okból kifolyólag működésképtelenné válik a hub. Ennek tipikus megoldása, hogy két routert alkalmaznak, amelyek a hub feladatát ellátják. Ez azonban szükségessé teszi azt, hogy minden egyes spoke sitenak kapcsolónia kell mindkét hub-hoz. Ilyenkor az egyik kapcsolat vagy backupként funkcionál, vagy mindkét kapcsolat használata esetén terheléselosztásra van lehetőség. Tovább bonyolíthatja a topológiát, ha valamilyen dial-in (pl. ISDN) backup megoldást szeretnénk a hálózatba implementálni. Ezt csak akkor nevezhetjük valóban biztonságos megoldásnak, ha hiba esetén a CPE eszközök az ISDN kapcsolatot az adott VPN-en kívül építik ki. A hálózat növekedésével többszintű hub-and-spoke topológiák jöhetnek létre.

3.4.2 Részlegesen vagy teljesen összekapcsolt topológia

Nem minden ügyfél tudja azonban megvalósítani a hálózatát az előbb tárgyalt topológia segítségével. Akkor fordulhat ez elő, ha a szervezet felépítése kevésbé hierarchikus struktúrájú vagy bizonyos szolgáltatások megkövetelik a különböző telephelyek közötti kommunikációt. Amennyiben minden egyes telephely között direkt kapcsolatot építünk ki, úgy teljesen összekapcsolt topológiáról beszélünk, néhány direkt kapcsolat esetén pedig részlegesen összekapcsolt topológiának nevezzük. A teljesen összekapcsolt (full meshed) topológia kivitelezése igen drága, ezért használata nem túl gyakori.

3.4.3 Hibrid topológia

Az olyan nagy hálózatokban, amelyek az Overlay VPN modellre épülnek, szokás kombinálni a hub-and-spoke és a részlegesen összekapcsolt topológiát. Ennek tervezése talán a moduláris hálózati felépítés alapján a legegyszerűbb. Először bontsuk szét a hálózatot hozzáférési, elosztási és központi részekre. Majd tervezzük meg külön-külön a központi és a hozzáférési réteget és a kettőt kapcsoljuk össze az elosztási rétegen oly módon, hogy a lehető legjobban elkülönítsük őket.

3.4.4 Extranet topológia

A topológiánál előtérbe kerülnek a biztonsági kérdések, ugyanis itt jellemzően különböző vállalatok között hozunk létre összeköttetéseket, amelyek valamilyen szempontból kapcsolatban állnak egymással. Általában igaz az, hogy minden hálózat a saját maga biztonságáért felelős. Azért ezt a megoldást preferálják az internettel szemben, mert itt lehetőség van különböző szolgáltatás minőségek (QoS) használatára.

3.5 Az MPLS és a VPN-ek kapcsolata

A hálózati szolgáltatók (Internet Service Providers - ISP) egyik legfőbb feladata virtuális magánhálózatok (VPN-ek) biztosítása ügyfelei számára, amelyek telephelyei földrajzilag egymástól távol esőek lehetnek.

Az MPLS kifejlesztése előtt az overlay VPN modell sokkal elterjedtebb volt, mint a peer-to-peer modell, de az MPLS esetében a peer-to-peer modell implementálása a célszerűbb. Az MPLS VPN nagy előnye, hogy az ügyfelek telephelyei között nem szükséges virtuális áramkörök beállítása, elegendő az ügyfél routerei és a szolgáltató határ routerei közötti kapcsolatok definiálása. Az egyes ügyfelek hálózatai ugyanazt a fizikai hálózatot használhatják, egymástól elkülönített logikai hálózatok formájában, így a különböző ügyfelek egymás hálózatait nem láthatják.

Az MPLS VPN-ek probléma nélkül tudják biztosítani ugyanazt a biztonsági szintet, mint a hagyományos VPN-ek esetében. Egymást átfedő címtartományok, intranetek, extranetek és a hub-and-spoke topológiák mind támogatottak az MPLS VPN-ben. Peer-to-peer VPN modell megvalósításával, az MPLS hálózat belsejének nem kell tudni a VPN-ek létezéséről, azt csupán az MPLS határán lévő eszközöknek kell kezelniük.

Az MPLS VPN a nagyvállalatok körében egyre népszerűbb, hiszen a bérelt vonalakhoz képest olcsóbbak, mégis biztonságosak és megbízhatóak. Mindemellett képes biztosítani a skálázhatóságot, és lehetővé tenni az ügyfél hálózatának kisebb részekre bontását, amely nagyvállalatoknál gyakran szükséges, hogy a különböző szervezeti egységek IT-infrastruktúráját elkülönítsék.

3.5.1 Az MPLS VPN Modell

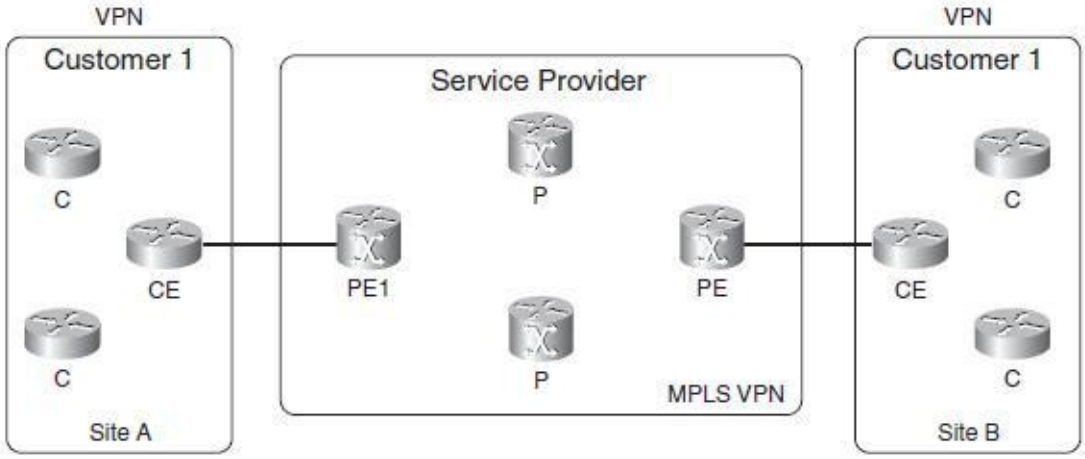
MPLS VPN esetén a routerek három típusát különböztetjük meg.

A **PE router** a Provider Edge router, az MPLS hálózat ingress/egress LSR routerei, amelyek ez ügyfél hálózattal közvetlenül kapcsolatban vannak. Ezen routerek címkézik fel a csomagokat és látják el a megfelelő VPN azonosítóval azokat, illetve távolítják el a címkét és továbbítják az ügyfél hálózatában a cél felé.

A **CE router** a Customer Edge router az ügyfél telephelyén található router, amely közvetlen layer 3-as kapcsolattal rendelkezik egy, esetleg több PE router felé. A CE routereknek nem

szükséges MPLS-t futtatniuk. A C routerek az ügyfél olyan routerei, amelyek nincsenek közvetlenül a PE-hez kapcsolva.

A **P routerek** a Provider routerei, az MPLS hálózat azon LSR-jei, amelyeknek nincs közvetlen kapcsolata az ügyfél CE routereivel, így az ügyfelek számára rejtve maradnak. Ezen routerek feladata a gyors címkekapcsolás.



18. ábra

A PE és CE routerek között továbbra is a hagyományos forgalomirányítási protokollok futnak, mint például RIP, OSPF, BGP.

3.5.2 Virtuális router a routerben

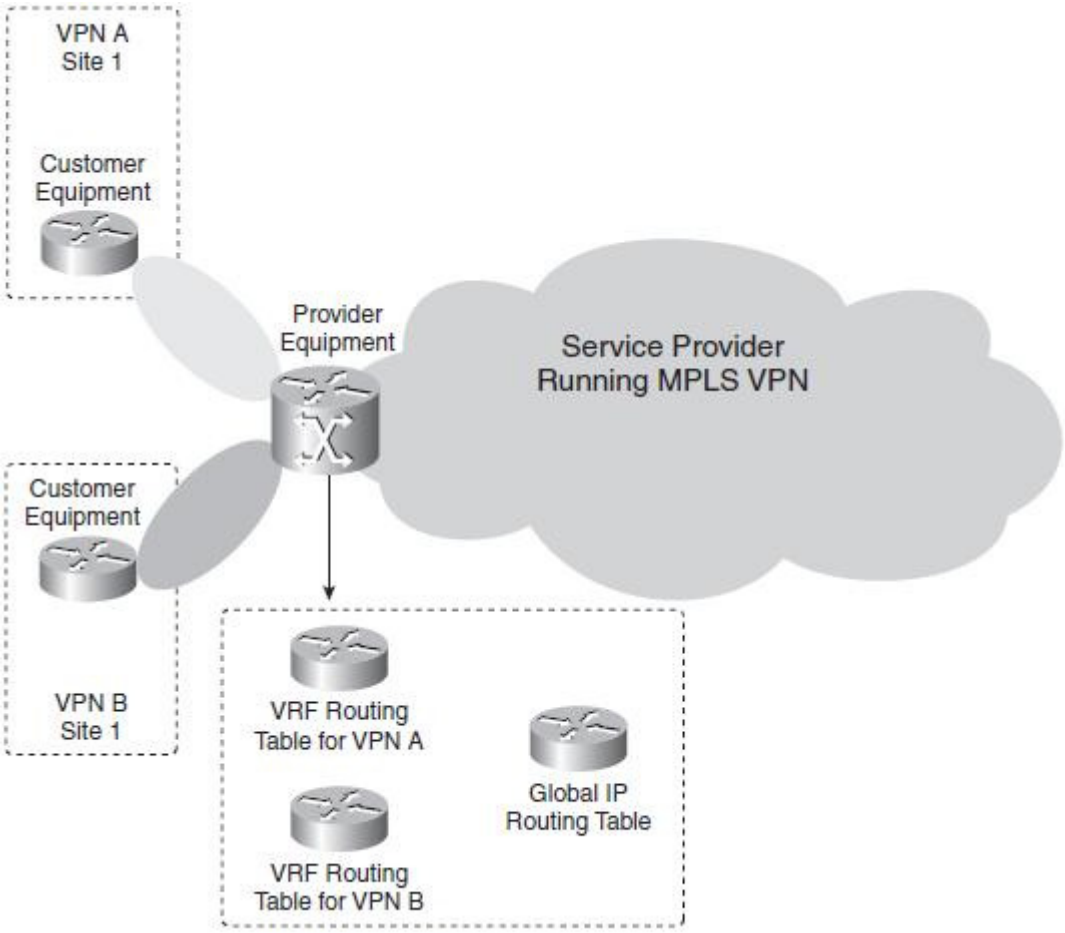
Az ügyfelek hálózataikban nyilvánvalóan nagyrészt privát IP címeket használnak. Ez azonban egyfajta limitációt jelent a peer-to-peer VPN implementációkra nézve, illetve számos egyéb bonyodalmat vet fel. Az MPLS/VPN technológia elegáns megoldást kínál ez előbb említett problémára. Minden VPN saját forgalomirányító és továbbító táblával rendelkezik a routeren, ebből adódóan a szolgáltató hálózatában lévő PE routerek több, különböző VPN-hez tartozó forgalomirányítási táblával rendelkeznek. Tartalmaznak még továbbá egy globális irányító táblát, melynek segítségével a szolgáltató hálózatában lévő többi routert érhetik el.

A virtuális router koncepció lehetővé teszi az ügyfelek számára, hogy globális vagy privát IP címeket használjanak. Azt azonban természetesen megköveteljük, hogy egy adott VPN-ben a privát IP címeknek egyedinek kell lenniük. Megfordítva: két különböző VPN tartalmazhat

megegyező IP címeket, kivéve azt az esetet, ha különböző VPN-ek közötti kommunikációra van szükség és azok ugyanazt a privát címtartományt használják.
Egy PE routerhez természetesen több CE router is kapcsolódhat, amelyek más-más ügyfelekhez tartozhatnak. Ezért ügyfelenként egyedi forgalomirányítási táblázatra van szükség, ami úgynevezett VRF routing táblák (Virtual Routing/Forwarding routing table) alkalmazásával valósítható meg. Az ügyfelek irányítótáblái egymástól el vannak különítve, habár ugyanahhoz a routerhez (PE) kapcsolódnak. Lényegében tehát ügyfél oldalról ezek a routerek dedikált routerekként jelennek meg, amelyek csak hozzá kapcsolódnak. A szolgáltatói oldalról nézve pedig a router szimulálja mindazokat a mechanizmusokat, melyek szükséges, hogy ezt a hatást keltsük az ügyfél irányába.
A VPN-eket mind a PE és mind a CE routerek interfészeihez is hozzá kell rendelni. Természetesen egy PE router több intefésze is tartozhat ugyanahhoz a VPN-hez.
Tehát minden PE routeren VPN-enként egy VRF routing táblázat található meg, továbbá egy globális forgalomirányítási tábla is. Ez a tábla tartalmazza az MPLS-felhő magjának forgalomirányítási információit, amelyek a PE routerek és a P routerek között cserélődnek.
Tehát a PE router a szolgáltató szemszögéből úgy néz ki, mintha azon több virtuális router futna, de az ügyfél számára azt a hatást kelti, mint egy hagyományos fizikai router.

3.5.3 Virtual Routing and Forwarding (VRF)

A VPN-ek létrehozása a PE routerek feladata, ezért a PE routerek minden egyes VPN-hez külön routing táblát, úgynevezett VRF-táblát (VPN Routing and Forwarding table) tartanak fent. Ezért mind a PE, mind a CE router interfészét, amelyek lehetnek logikai interfészek is, hozzá kell rendelni egy VRF-hez.
Az LSP-k felépítéséhez szükséges információk mellett tehát a VPN információk tárolása is címkék segítségével történik, ezt az MPLS VPN hálózatoknál a kétszintű label stack segítségével valósítják meg.



19. ábra

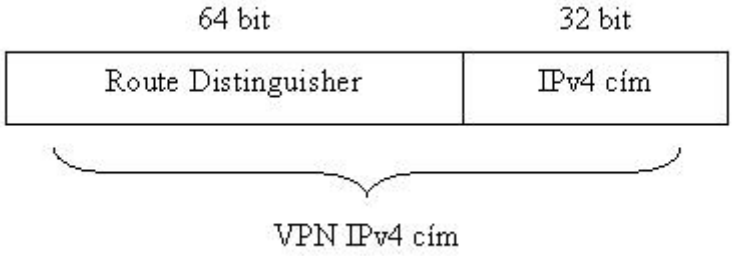
Az alsó címke a VPN-t azonosítja, ezért ez a címke csak a PE routerek számára hordoznak információt, a P routerek ezt nem veszik figyelembe. Az LDP címke, vagyis a verem tetején lévő címke pedig csak az MPLS belsejében történő továbbításra szolgál. Ezeket a címke információkat a P és PE routerek között LDP, RSVP-TE vagy CR-LDP cseréli ki, és ezeket az információkat a PE routerek a globális forgalomirányítási tábláikban tárolják.

3.5.4 Route Distinguisher

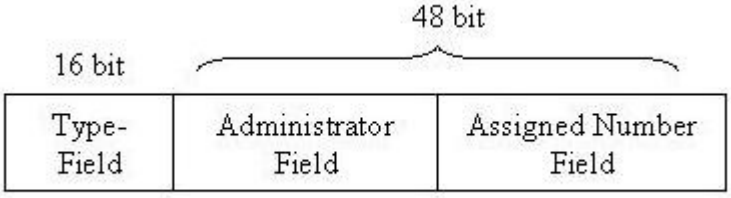
Több VPN esetén az egyes VPN-ekben előfordulhatnak ugyanazon privát IP-címtartományok. De a BGP, amely a PE routerek közötti routing információk cseréjért felelős csak egyértelmű, globális IP-címekkel tud dolgozni, tehát a különböző VPN-ek azonos IP-címeit nem tudja megkülönböztetni. A Multiprotocol BGP egy kiegészítése a BGP-nek, amely már az úgynevezett VPN-IPv4-es vagy VPNv4-es címeket is támogatja, amely lehetővé teszi a

különböző VPN-ekben egymást átfedő privát IP címtartományok használatát Természetesen egy adott VPN-en belül minden IP-címnek egyedinek kell lennie. A BGP 4-es verziójának multiprotokollos kiegészítését a 2283-as RFC írja le.

Egy VPN-IPv4-es cím 96 bit hosszú és két részből áll, egy 64 bites azonosítóból a Route Distinguisher-ből, és a 32 bites IPv4 címből. Tehát minden VPN kap egy 64 bites azonosítót, így az egymást átfedő privát IP címtartományok egyértelműen megkülönböztethetőek lesznek.



A Route Distinguisher egy 16 bites típus mezőből és a 48 bites értékből áll. RFC 4364.



A típus mezőnek jelenleg 3 értékét definiáltak a 0-t, az 1-t és a 2-t:

- Ha a típus mező értéke 0, akkor az administrator mező étéke 16 bites, az assigned number mezőé pedig 32 bites. Az administrator mező értéke az úgynevezett autónom rendszerazonosító (AS-Number - ASN), amely minden autonóm rendszer számára egyedi azonosító. Az assigned number pedig a szolgáltató által szabadon választott szám. Természetesen az adminisztrator mezőnek és az assigned mezőnek együttesen továbbra is biztosítania kell az egyediséget.
- Ha a típus mező értéke 1, akkor az administrator mező étéke 32 bites, az assigned number mezőé pedig 16 bites. Az administrator mező ekkor egy IP-címet tartalmaz, amely az IANA (Internet Assigned Numbers Authority) által kiosztott IP-cím. A publikus IP-címtartomány használata szigorúan tilos.

- Ha a típus mező értéke 2, akkor az administrator mező étéke szintén 32 bites, az assigned number mezőé pedig 16 bites. Az administrator mező értéke egy 32 bites autónom rendszerazonosító (BGP-AS4), amelyet szintén az IANA oszt ki.

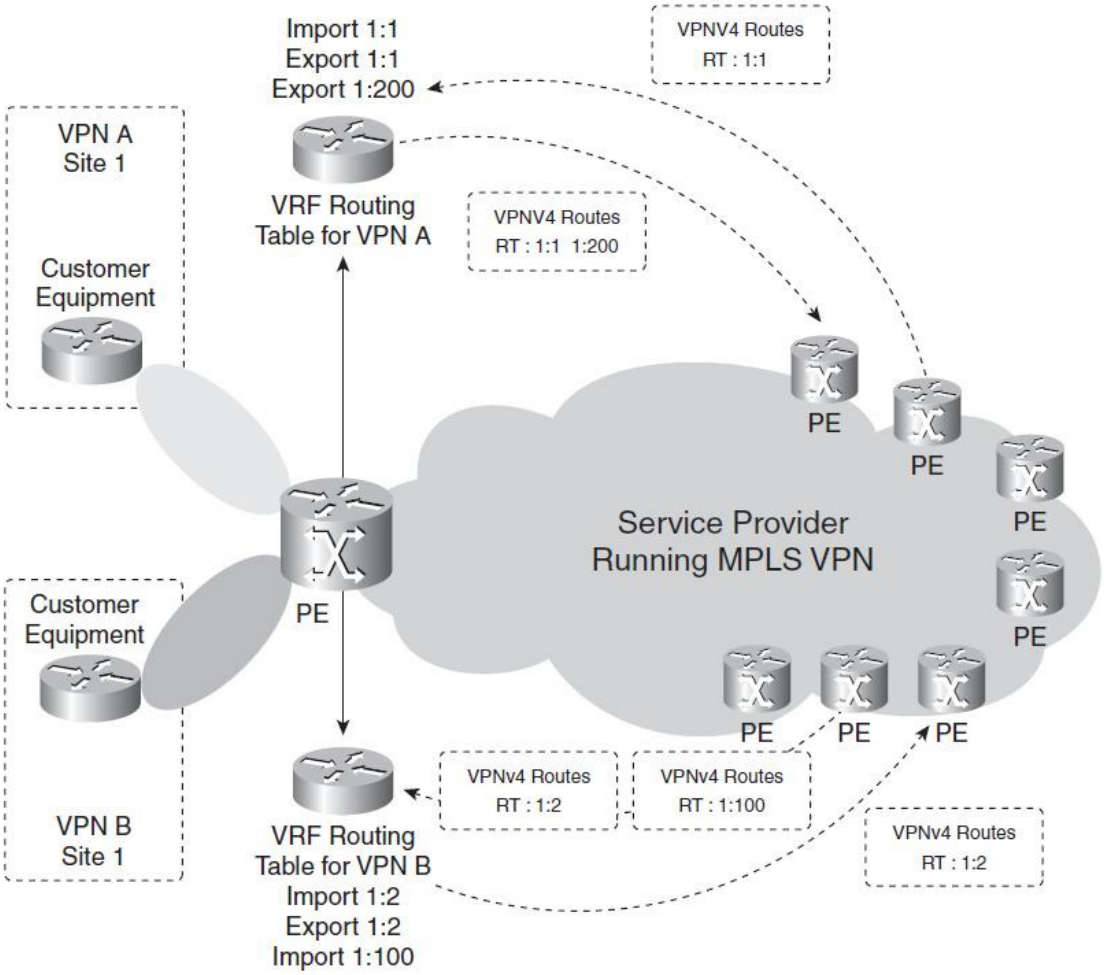
A VPN-IPv4 címeket csak a PE routerek közötti routing információk kicserélésére használja a MP-iBGP, a CE routerek nem használják a VPN-IPv4 címeket, csak a hagyományos 32 bites IPv4 címeket.
Tehát a Route Distinguisher egyértelművé teszi a privát IP-címeket és lehetővé teszi azonos IP-címek használatát különböző VPN-ekben.

3.5.5 Route Targets

A Route Distinguisher önmagában nem elegendő a VPN-routinginformációk kicseréléséhez, a MP-BGP-nek azt is ismernie kell, hogy a routinginformáció melyik VRF-táblához tartozik. Ezért minden VRF-táblához be kell jegyezni a route target értékét. A route target szintén 64 bit hosszú, és két funkciót lát el:

- Export-funkció: megad egy azonosítót, amely alatt a VRF routinginformációi a többi PE-nek egy MP-BGP-Update során továbbítva lesz.
- Import-funkció: meghatározza, hogy a többi PE routertől kapott VPN-routinginformációk közül melyeket importálja a VRF-táblába.

A PE routerek a közvetlenül csatlakoztatott CE routerek VPN-routinginformációit megtanulják, és ezeket MP-BGP segítségével kicserélik a többi PE routerrel. A CE routerek a PE routerek felé IPv4-címeket hirdetnek, ezeket a PE router ellátja a VRF-nek megfelelő Route Distinguisher-rel, így VPN-IPv4 címeket előállítva.



20. ábra

A PE routerek a következő routinginformációkat küldik el a többi PE router számára:

- A VPN-IPv4-címek
- A saját Loopback címűket, amely az útvonalak BGP-next hopjaként szolgál.
- A hozzárendelt címkét.
- A VRF-táblához hozzárendelt route target értéket vagy értékeket.

Amikor egy PE router VPN-routinginformációt kap, összehasonlítja annak route target értékét a saját VRF-tábláihoz tartozó route target értékekkel, és ha talál megfelelő VRF-táblát akkor beilleszti abba a megfelelő routinginformációkat.
A VPN-routinginformációjának cseréjével egy időben a címkekapcsolt útvonalakat (LSP-k) is definiálják a PE routerek.

A PE és P routerek az egymás közti szomszédságot az MPLS-en belül OSPF vagy IS-IS segítségével derítik fel. A szomszédok IP-címeit a globális forgalomirányító táblában tárolják.

3.5.5.1 Átfedés a VPN-ek között

Az MPLS VPN lehetővé teszi, hogy a VPN-ek között routing információkat cseréljünk, ezáltal közös erőforrások több VPN-ből is elérhetőek legyenek. De emellett biztosítja, hogy a VPN-ek továbbra se lássák egymást.

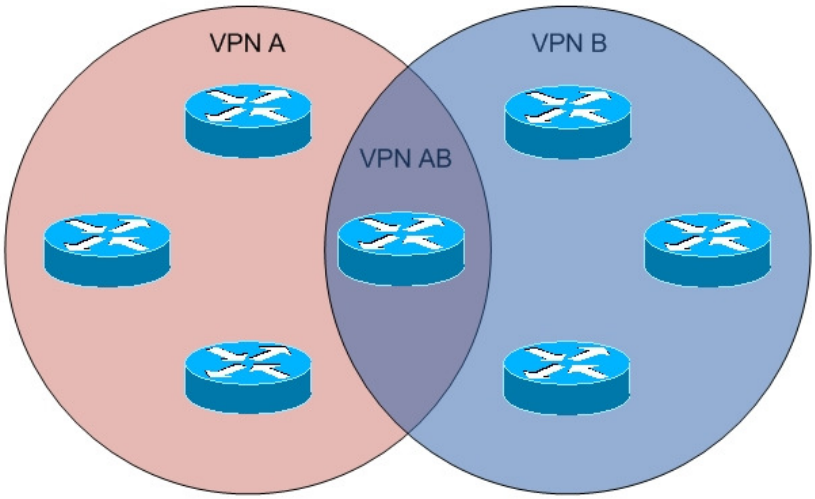
Gyakran előfordul, hogy egy vállalat a szervezeti egységeinek külön VPN-eket tart fent. De szükség lehet arra, hogy bizonyos rendszereket több VPN-ből is el lehessen érni. Tipikusan a központi menedzsment esetén van erre szükség.

Az egymást átfedő VPN-eket a route target-ek segítségével, illetve azok megfelelő exportálásával és importálásával lehet megvalósítani. Arra is lehetőség van, hogy az egyes telephelyek útvonalait külön-külön kezeljük, és több route target értékkel exportáljuk. Ezzel a VPN-ek közötti átfedések lényegében tetszőlegesen megadhatóak.

A következő két példa szemlélteti, a route targetek alkalmazási lehetőségeit.

Ez első példa azt mutatja be, hogyan valósítható meg, amikor egy erőforrást több VPN-ből is el kell érni, de a VPN-ek nem láthatják egymást.

Esetünkben VPN A-nak és VPN B-nek is hozzá kell férnie a VPN AB-vel jelölt telephely hálózatához.



21. ábra

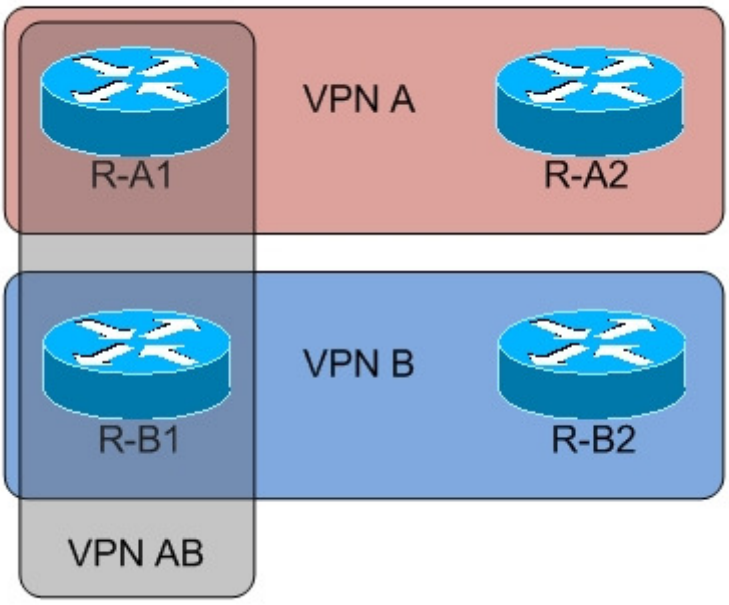
Ez úgy valósítható meg, hogy ha a VPN AB-ben mind a VPN A, mind a VPN B route target értékeit importáljuk és exportáljuk. Ekkor a VPN AB exportálja a forgalomirányítási információt VPN A-ba és VPN B-be is, így a VPN AB mind a két VPN-ből elérhető lesz, és a VPN AB importálja mind a két VPN útvonalait, hogy a VPN A és VPN B is elérhető legyen a VPN AB-ből. De a VPN A és a VPN B továbbra sem fér hozzá egymás erőforrásaihoz.

VPN A
Import route target: 100:1
Export route target: 100:1

VPN B
Import route target: 100:2
Export route target: 100:2

VPN AB
Import route target: 100:1, 100:2
Export route target: 100:1, 100:2

A második példa azt mutatja be, hogyan lehet megvalósítani azt, hogy két különböző VPN egy-egy része lássa egymást, a többi része pedig nem. Ahogy az ábrán is látható, a VPN A R-A1 nevű telephelye kapcsolatban van a VPN B R-B1 telephelyével is, nem csak a saját VPN-ükön belül lévő többi telephellyel. De az R-A1 nevű telephely az R-B1-en kívül nem fér hozzá a VPN B többi telephelyéhez, ugyanígy az R-B1 is csak az R-A1-el áll kapcsolatban, R-A2-vel már nem. Ezt a megoldást extranetnek is hívják.



22. ábra

R-A2:

Import route target: 100:1

Export route target: 100:1

R-B2:

Import route target: 100:2

Export route target: 100:2

R-A1:

Import route target: 100:1, 100:12

Export route target: 100:1, 100:12

R-B1:

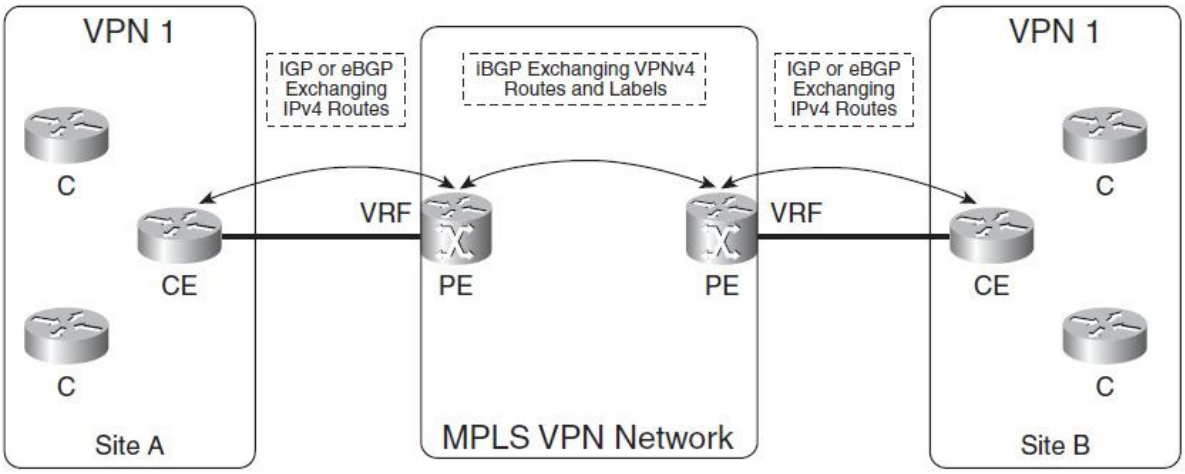
Import route target: 100:2, 100:12

Export route target: 100:2, 100:12

3.5.6 Az MPLS VPN és a BGP kapcsolata

A 2547-es RFC írja le, hogyan történik a VPN-ek forgalomirányítási információinak a cseréje BGP segítségével. Attól függően, hogy mely routerek között történik a VPN információk cseréje, három esetet különböztetünk meg:

- CE-PE routerek között a kommunikáció OSPF, RIPv2, eBGP vagy static route segítségével történik. A CE router IPv4 routing információkat küld a PE routernek, amely azokat elhelyezi a megfelelő VRF táblában.
- PE-PE között a VPN forgalomirányítási információk cseréjét a MP-iBGP (Multiprotocol - internal BGP) biztosítja. A PE routerek közötti globális forgalomirányítási táblák karbantarásához pedig a standard iBGP-t használja.
- PE-CE között szintén OSPF, RIPv2, eBGP vagy static route segítségével történik a kommunikáció. A PE router a BGP-ből a VRF-be importálja a VPN információkat, majd az IPv4 forgalomirányítási információkat a fent említett protokollok valamelyikének segítségével átadja a CE routernek.

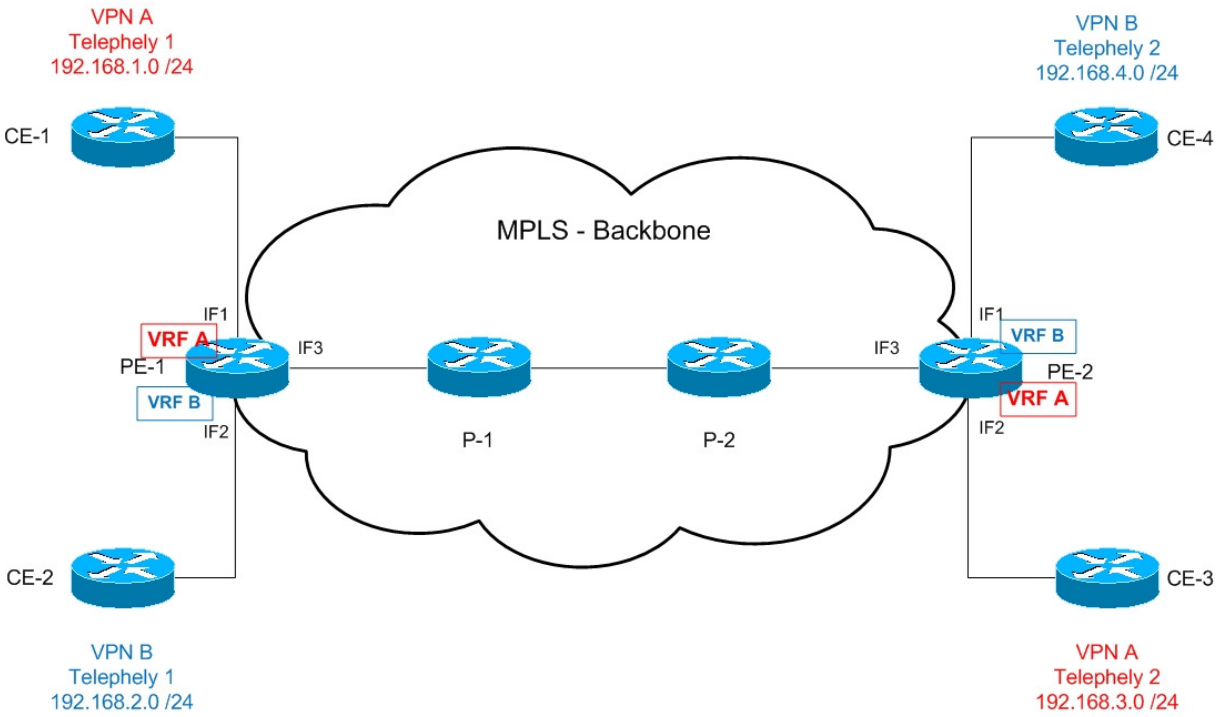


23. ábra.

A Provider routerek nem vesznek részt a VPN információk cseréjében, hiszen azok csak a felső MPLS címkét veszik figyelembe a továbbításhoz.

3.6 Példa az MPLS-VPN-re

A lenti ábra egy Internet Service Provider (szolgáltató) hálózatát mutatja be. Természetesen ez egy nagyon leegyszerűsített esett, a valóságban ennél sokkal összetettebb hálózatok vannak. Jelen esetben két ügyfél (customer) csatlakozik a hálózathoz, jelöljük őket VPN A és VPN B-vel. Mindkét VPN két telephelyből áll. A VPN A-hoz (az ábrán pirossal jelölve) a CE-1 és CE-3 nevű router tartozik. A VPN B-hez (az ábrán kék) pedig a CE-2 és CE-4 nevű routerek.



24. ábra

Mindkét PE routeren megtalálható egy-egy VRF-tábla mindkét VPN-hez.

A VRF-tábláknak minden esetben tartalmazniuk kell a következő információkat:

- VRF
- Route Distinguisher
- Route Target
- VPN-IPv4 címek

Ez a fenti példában a következőképpen néz ki:

	VPN A	VPN B
VRF	VRF-A	VRF-B
Route Distinguisher	450:11	450:22
Route Target	100:11	100:22

3.6.1 A forgalomirányítási információk kicserélése

3.6.1.1 A CE és PE routerek között

Amennyiben a PE és CE routerek között valamilyen forgalomirányítási protokollt alkalmaznak, akkor a PE-1 router megkapja a forgalomirányításhoz szükséges információkat a CE-1 és CE-2 nevű routerektől. A CE-1-től kapott IPv4 címeket ellátja egy MPLS-címkével és tárolja azokat a VRF-A nevű VRF-táblában. Ugyanezt elvégzi a CE-2 router esetén is.
Ekkor a PE-1 router VRF-táblái a következő VPN-forgalomirányítási információkat tartalmazza:

VRF-A a PE-1 routeren:

MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
1001	192.168.1.0 /24	Direkt	IF1	-

VRF-B a PE-1 routeren:

MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
2002	192.168.2.0 /24	Direkt	IF2	-

A forgalomirányítási információk cseréje hasonlóan történik a PE-2 illetve CE-3 és CE-4 esetén. Ekkor a VRF-táblák a következőképpen alakulnak:

VRF-A a PE-2 routeren:

MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
1003	192.168.3.0 /24	Direkt	IF2	-

VRF-B a PE-2 routeren:

MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
2004	192.168.4.0 /24	Direkt	IF1	-

3.6.1.2 A PE routerek között

Azokat a forgalomirányítási információkat, amelyeket a PE routerek a közvetlenül csatlakoztatott CE routerektől tanultak, MP-BGP segítségével kicseréli a többi PE routerrel.
De előtte a PE routerek a CE routerektől tanult 32 bites IPv4-címeket 96 bites VPN-IPv4-címekké alakítják, vagyis ellátják a VRF-táblához tartozó Route Distinguisher értékkel. Ezen kívül minden bejegyzéshez társítja az export route target értéket is.
Az itt használt címkék a VPN-ek azonosítását szolgáló címkék, tehát a kételemű címke verem alján elhelyezkedő címkék. Amíg a csomag áthalad az MPLS felhőn, ezek a címkék nem változnak, mert a P routerek a verem tetején lévő LSP-t azonosító címkét dolgozzák fel, és cserélik le.

A PE-1 router a következő információkat küldi el a PE-2-nek:

PE-1	VPN-IPv4	Címke	BGP Next hop	Route Target
VRF-A	10:1:192.168.1.0/24	1001	PE-1	100:11
VRF-B	20:2:192.168.2.0/24	2002	PE-1	100:22

A PE-2 pedig a következő információkat küldi el a PE-1-nek:

PE-2	VPN-IPv4	Címke	BGP Next hop	Route Target
VRF-A	10:1:192.168.3.0/24	1003	PE-2	100:11
VRF-B	20:2:192.168.4.0/24	2004	PE-2	100:22

Amikor egy PE router egy másik PE routertől VPN-IPv4 forgalomirányítási információt kap, akkor összehasonlítja annak Route Target értékét a saját VRF-tábláinak Route Target értékeivel (import route target), és ha egyezőt talál, akkor a VPN-IPv4-címet visszaalakítja IPv4-be, és tárolja a route targethez tartozó VRF-táblájában.

Miután a PE-1 és PE-2 kicserélte az információkat, a VRF-tábláik a következőképpen néznek ki:

PE-1	MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
VRF-A	1001	192.168.1.0 /24	Direkt	IF1	-
VRF-A	-	192.168.3.0/24	PE-2	-	1003
VRF-B	2002	192.168.2.0 /24	Direkt	IF2	-
VRF-B	-	192.168.4.0 /24	PE-2	-	2004

illetve,

PE-2	MPLS-címke BE	IPv4-cím	BGP Next hop	Kimenő interfész	MPLS-címke KI
VRF-A	1003	192.168.3.0 /24	Direkt	IF2	-
VRF-A	-	192.168.1.0/24	PE-1	-	1001
VRF-B	2004	192.168.4.0 /24	Direkt	IF1	-
VRF-B	-	192.168.2.0 /24	PE-1	-	2002

3.6.1.3 A PE és CE routerek között

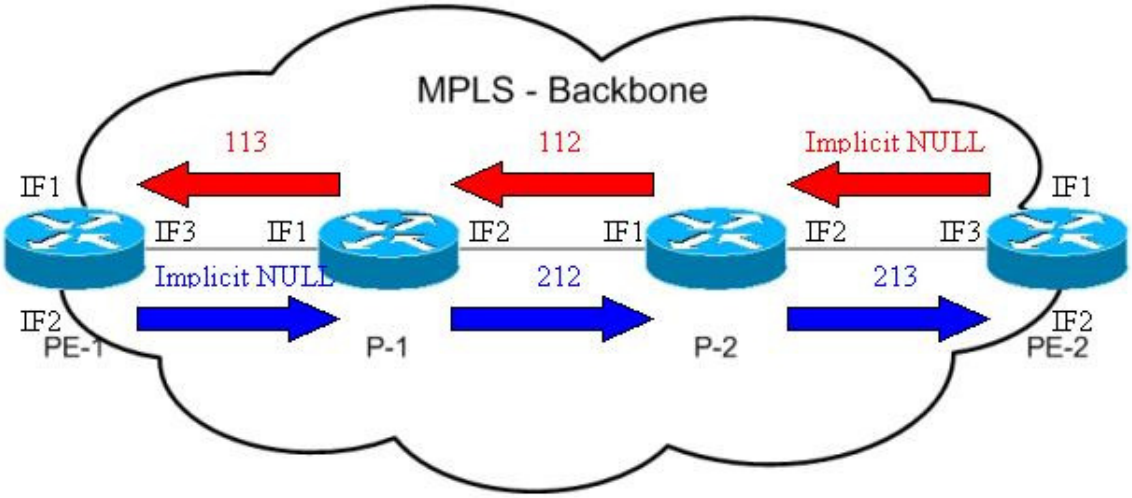
Miután a PE routerek kicserélték egymás között az információkat, értesítik a CE routereket is. Például a PE-1 router informálja a CE-1 routert, hogy a 192.168.3.0 /24-es hálózat rajta keresztül elérhető. Ugyanígy a CE-2-t is informálja, hogy a VPN-B-ben mely további hálózatok elérhetők rajta keresztül. Továbbá a PE-2 is informálja a CE-3 és CE-4 routereket.

3.6.1.4 Az LSP-k felépülése az MPLS-VPN hálózatban

A PE routerek a VPN-routing információk kicserélésével egyidejűleg a címke kapcsolt útvonalakat (Label Switched Path-LSP) is definiálják. Ez ugyanúgy történik, mint az MPLS-hálózatokban, a VPN-ektől függetlenül.

A PE és P routerek az egymás közti szomszédságot az MPLS-en belül OSPF vagy IS-IS segítségével derítik fel. A szomszédok IP-címeit a globális forgalomirányító táblában tárolják, elkülönítve az egyes VPN-ekhez tartozó VRF-táblától.

A példában két darab LSP kerül meghatározásra, az egyik a PE-1-től a PE-2-ig, a másik pedig a PE-2-től a PE-1-ig.



25. ábra

A PE-1-től a PE-2-ig tartó LSP meghatározásánál az első címkét a PE-2 határozza meg, és egy címke elosztó protokoll segítségével, amilyen például az LDP, értesíti a P-2 routert, hogy minden cél, ami rajta keresztül elérhető mely címkét kapja. De a korábban már bemutatott Penultimate Hop Popping módszer alkalmazásával a PE-2 router az implicit NULL címkét küldi a P-2 routernek, amelyből a P-2 tudni fogja, hogy ha a PE-2 routernek továbbít adatot, akkor a felső címkét el kell távolítania.
Ezután a P-2 értesíti a P-1-et (112-es címke), és a P-1 informálja a PE-1-et (113-as címke). Hasonlóan történik a címkék meghatározása a P-2-től P-1-ig tartó LSP esetén is.

A P routereknek csak a kételemű címkeverem felső, azaz az LSP-t meghatározó címkéje alapján kell továbbítási döntést hoznia, így a címke-megfeleltetési táblázatuk is egyszerűbb lesz.
A példánkban a P-1 és P-2 routerek táblázata a következőképpen alakul. A táblázatokban látható, hogy amikor egy P router az LSP utolsó tagjának, vagyis egy PE routernek továbbít, akkor a kimenő címke értéke helyett a POP érték szerepel. Ez azt jelenti a P router számára, hogy a címkét el kell távolítania a címke verem tetejéről.

P-1 esetén:

Bemenő interfész	Bemenő címke	Kimenő Interfész	Kimenő címke
IF1	113	IF2	112
IF2	212	IF1	POP

P-2 esetén:

Bemenő interfész	Bemenő címke	Kimenő Interfész	Kimenő címke
IF1	112	IF2	POP
IF2	213	IF1	212

A PE router feladata ennél összetettebb, hiszen két címkét is kezelniük kell. Az alsó címke a fentebb bemutatott VRF táblának megfelelően kerül meghatározásra. A felső, azaz az LSP-t azonosító címkéket pedig a globális routing táblák alapján kezelik a PE-k. A példánkban ez a következőképpen néz ki.

PE-1 esetén:

Bemenő címke	Next hop	Kimenő Interfész	Kimenő címke
-	PE-2	IF3	113

PE-2 esetén:

Bemenő címke	Next hop	Kimenő Interfész	Kimenő címke
-	PE-1	IF3	213

Természetesen ezek a táblázatok több PE router esetén több sort tartalmaznának, de ezen a példán is látható, hogy miként kezelik a címkéket a PE routerek az MPLS-felhő határán.

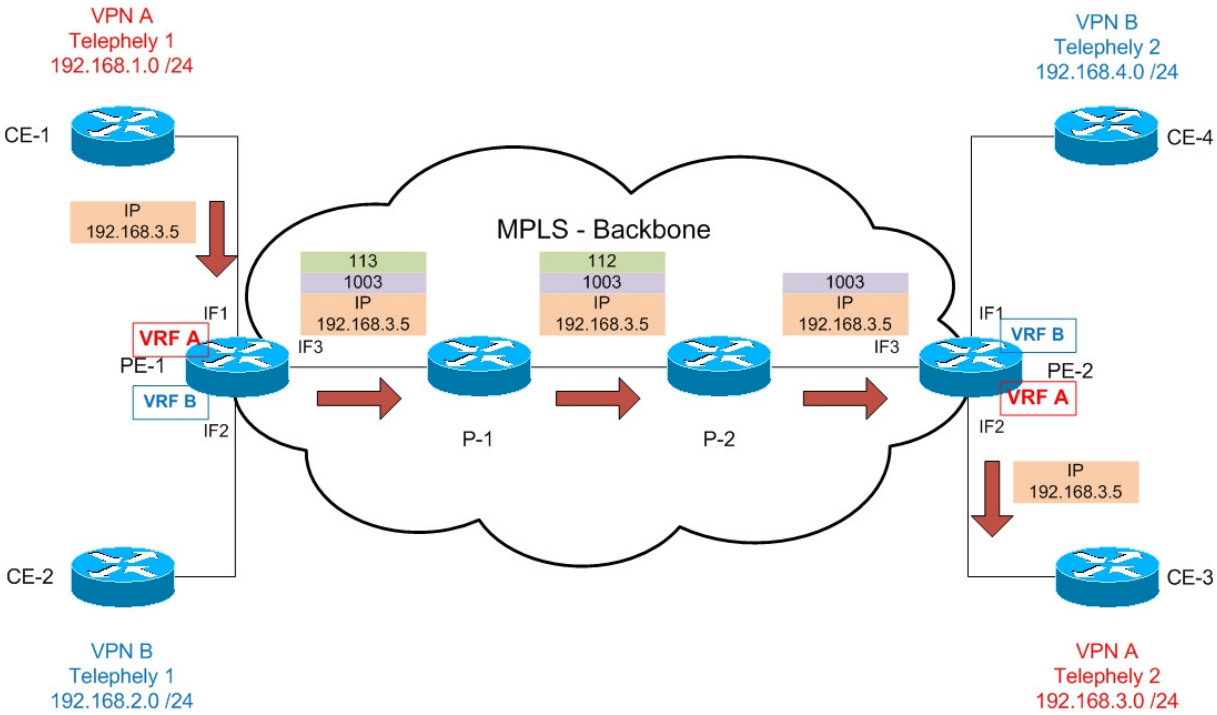
Ha a PE router valamelyik CE routertől kap egy csomagot, akkor először a fentebb leírt módszerrel ráhelyezi a VPN címkét, aztán pedig az LSP-t meghatározó címkét is, az alapján, hogy melyik másik PE router felé kell a csomagot továbbítani.

Ha a PE router egy P routertől kap csomagot, akkor tudja, hogy a legfelső címkét már a P router eltávolította, hiszen erre ő kérte az implicit Null címkével. Ezért esetünkben csak a VPN címkével kell foglalkoznia.

3.6. 2. Adatfolyam

A következő ábra azt mutatja be, miként jut el egy csomag egyik telephelyről a másikra az MPLS-felhőn keresztül. A VPN A 1-es telephelyéről indul a csomag, amelynek cél IP-címe a 192.168.3.5, amely a VPN A 2-es telephelyén található.

- A CE-1 és a PE-1 között egy szabványos IP-csomag továbbítódik.
- Ezt a csomagot a PE-1 router ellátja a két megfelelő címkével. Az alsó a VPN-t azonosítja, a felső az LSP-t.
- A P-1 router csak a felső címke vizsgálatát, cseréjét végzi el, és továbbítja a csomagot a P-2 routernek.
- P-2 router is megvizsgálja a felső címkét, és a táblázatában látja, hogy a Penultimate Hop Popping módszert alkalmazva el kell távolítania a felső címkét. A csomagot egy címkével továbbítja a PE-2 routernek.
- A PE-2 router megvizsgálja a címkét, eltávolítja azt, és továbbítja az IP-csomagot aCE-3 routernek.



26.ábra

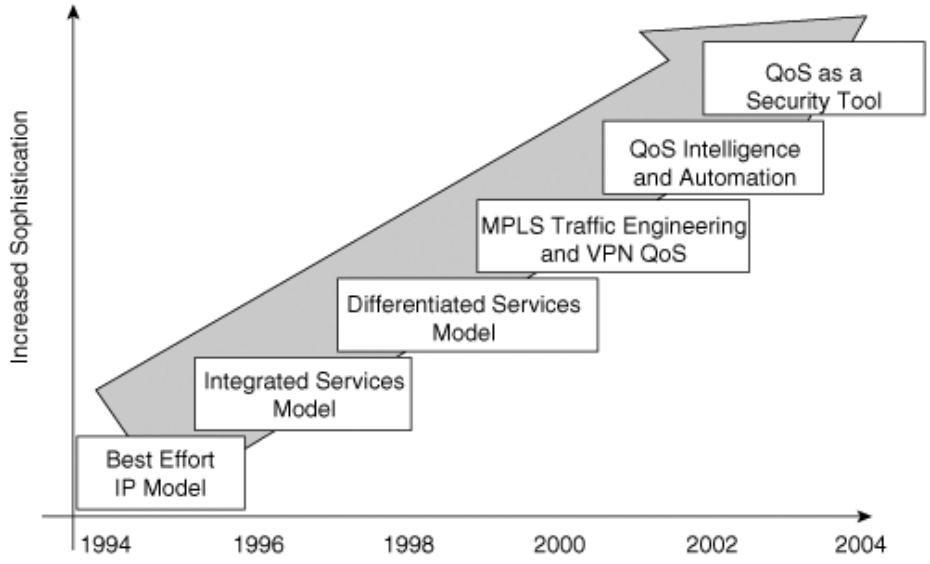
4. A szolgáltatás minősége

4.1 A QoS bemutatása

Ebben a fejezetben a szolgáltatás minőségéhez kapcsolódó témaköröket tárgyaljuk, melyek a szakirodalomban QoS (Quality of Service) összefoglaló néven kerülnek elő. Már a 90-es évek elején megjelent az igény arra vonatkozóan, hogy valamilyen módon priorizáljuk a hálózati adatforgalmat. Az IP alapú hálózatok rohamos mértékű növekedésével egyre több helyen merült fel ez a probléma, jellemzően az üzleti szférában. Számos olyan kulcsfontosságú szolgáltatás létezik, melyeknek működniük kell olyan esetben is, amikor a hálózat túlterhelt állapotban van. Mindez jelentősen felértékelődött abban az időben, amikor a technika fejlődése lehetővé tette az IP alapon történő hangátvitelt. Napjaink multimédiás alkalmazásai (pl. video konferencia) speciális minőségű adatátvitelt igényelnek. Az összes felhasználó megfelelően működő és megbízható hálózatot szeretne az alkalmazásaihoz. Problémát jelenthet, hogy a valósidejű, nagy sávszélességet igénylő szoftverek ne foglalják le teljes mértékben a hálózatot. A megfelelő igények kielégítésére jött létre a QoS.

4.2 A QoS fejlődése

Az 1990-es évek közepén szinte az összes IP alapú hálózatot a „Best effort”, azaz a legjobb szándékon alapuló csomagtovábbítás jellemezte.



27. ábra

Ez alatt az idő alatt egyes vállalatoknál illetve szolgáltatóknál sajátos és jellemzően bonyolult megoldások születtek arra, hogy a különböző alkalmazások által generált hálózati forgalmak más és más szolgáltatási szintet kapjanak.
Felismerte az IETF is, hogy az új alkalmazásokhoz nem megfelelőek az eddigi minőségi mutatók. Ennek eredményeként két lehetséges megoldással foglalkoztak ezután.

- Adatfolyam alapú QoS: az egyes erőforrásoknak adatfolyamok szerint történik a lefoglalása. Így itt a minőségi garanciát az adatfolyamok kapják, ebből következően az adatfolyamok állapotát nyilván kell tartani a hálózatban.
- Nem adatfolyam alapú Qos: az egyes adatcsomagok megjelölésre kerülnek és a továbbiakban a csomagok feldolgozása, valamint továbbítása az előre meghatározott jelölési osztályok szerint történik.

Az 1990-es évek végére a QoS módszerek egyre kifinomultabbá váltak és alkalmazásra kerültek olyan fejlett hálózati technológiákba mintpéldául az MPLS és a VPN.

A legújabb trend a QoS-t illetően a minél egyszerűbb implementálás és az automatizáció. A cél a gyorsan és hatékonyan alkalmazható „intelligens” megoldások használata az IP alapú hálózatok esetében. Számos beállítási lehetőség áll rendelkezésünkre a QoS konfigurálásához. Hozzáértő szakemberek segítségével testreszabott megoldások implementálhatóak, ugyanakkor ez esetenként igen bonyolult konfigurációkat eredményez. Sok esetben azonban erre nincsen szükség, illetve nem áll rendelkezésre megfelelő erőforrás az összetett, egyedi elképzelések kivitelezéséhez. Általánosságban kijelenthetjük, hogy a QoS nagyméretekben jól standardizálható.

4.3 QoS modellek

Egy alkalmazás bármikor és bármilyen mennyiségben küldhet adatot. A hálózat megbízhatósági, késleltetési és teljesítmény garancia nélkül továbbítja a PDU-kat. Ezt best effort modellnek nevezzük és ebben az esetben nem beszélünk szolgáltatási minőségekről.

4.3.1 Intserv (Integrated Services)

A QoS standardizálására ez volt az első próbálkozás, mely az 1994-ben megjelent 1633-as RFC-ben van megfogalmazva. A módszer lényege, hogy adatfolyamon alapszik és az egyes kapcsolatokhoz szükséges erőforrásokat próbálja meg lefoglalni. Az alkalmazás jelzi elvárásait a hálózatnak, majd válaszra vár. Amennyiben ez a válasz pozitív, azaz kielégíthető a kért sávszélesség, akkor elindulhat az adatátvitel a megadott minőségi szinten és lefoglalásra kerülnek a megfelelő erőforrások. A lefoglalás érvényben marad mindaddig, amíg az alkalmazás nem jelzi a végét vagy a továbbiakban a hálózat képtelen a megfelelő erőforrásokat biztosítani.
A módszer alapvető célja a real-time (valós idejű) alkalmazások interneten való támogatása volt, garantálva a szolgáltatást. Az Intserv elmélete illeszkedik például az ATM technológiához.
Az IETF-nél kidolgozták az RSVP (Resource Reservation Protocol) protokollt. Ez egy szimplex protokoll, ahol az erőforrás lefoglalását külön kell mindkét irányban elvégezni, amelyet legtöbbször a küldő kezdeményez, azonban a tényleges foglalás a fogadótól visszafelé haladva történik. A forrás és a cél között levő útvonalon az összes hálózati elemnek támogatnia kell az RSVP-t, különben a kapcsolat nem épülhet fel és hibajelzést kapunk. A működésből adódóan a foglalásokat időnként meg kell ismételni. Az erőforrások automatikus felszabadítása biztosított, ha egy adott idő után nem érkezik megerősítés.
A fentiekből következik, hogy mivel alkalmazásonként tarjuk nyilván a különböző állapotokat, ezért ez viszonylag nagy terhet jelent a hálózatnak. A skálázhatósága nem megfelelő ezért a szolgáltatók körében nem terjedt el.

4.3.2 Diffserv (Differentiated Services)

Az IETF második próbálkozása a QoS standardizálására a Diffserv modell, melynek leírása az 1998-ban megjelent RFC 2475-ben található. A szolgáltatás minőségének meghatározását más alapokra helyezi, a hálózat állapotát nem kapcsolatonként tartjuk nyilván. A módszer lényege, hogy a csomag fejrészében elhelyezünk egy bizonyos jelzést (amely lehet IPP - IP Precedence, vagy az utódja DSCP - Differentiated Services Code Point), ezt követően a csomagok kezelése ezen jelzések alapján történik.

Az alapgondolat az, hogy a hálózatban csak néhány, speciális kiszolgálási osztályt különböztetünk meg. A hasonló igényű hálózati forgalmat azonos osztályba soroljuk, melyekhez erőforrások allokálhatók. Az osztályokba (FEC – Forwarding Equivalence Class) történő besorolás a hálózatba belépéskor történik, majd ezt követően a forgalom további továbbítási sorokba kerül. Az osztályok a továbbítás alapján PHB (Per Hop Behaviour – Ugrásonkénti Viselkedésmód) csoportokba kerülnek, egy PHB-ba több FEC is tartozhat. A diffserv tehát az egyes csomagok „viselkedési módját" befolyásolja a hálózaton történő továbbítás során. Szolgáltatói hálózatokban jól alkalmazható, a továbbiakban ezzel a megoldással foglalkozunk részletesebben.

4.4 QoS a gyakorlatban

A fejezetben a QoS-t a gyakorlati oldaláról közelítjük meg napjaink elvárásait szem előtt tartva. Ebbe beletartozik a standardizáció, a hatékony megoldások használata. Szolgáltatói környezetben a QoS-t az IETF Diffserv modellje alapján implementáljuk.

4.4.1 QoS jelölési sémák

4.4.1.1 Alapelv

Az IPv4 header 13 mezőt tartalmaz, melyek közül a Type of Service (TOS) hordozza a QoS információt.

+	Bits 0 – 3	4 – 7	8 – 15	16 – 18	19 – 31
0	Version	Header Length	**Type of Service (TOS)**	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live (TTL)		Protocol	Header Checksum	
96	Source IP Address				
128	Destination IP Address				
160	Options (optional)				
160/192+	Data				

A fenti táblázatban látszik, hogy a TOS a 3. mező az IPv4 csomag fejrészében. Konfigurációtól függően használhatunk IP Precedence illetve DSCP értékeket a QoS beállításához, ennek megfelelően változtatjuk a TOS mezőben a biteket.

4.4.1.2 IP Precedence

Bit 0	1	2	3	4	5	6	7
IP Precedence			0 = Normal Delay 1 = Low Delay	0 = Normal Throughput 1 = High Throughput	0 = Normal Reliability 1 = High Reliability	reserved	
P2	P1	P0	T2	T1	T0	CU1	CU2

A 791-es RFC-ben a fenti táblázat szerint alakul a TOS mező 8 bitje. Az IPP értéket tehát a 0-3. bit határozza meg.

4.4.1.3 DSCP

Bit 0	1	2	3	4	5	6	7
IP DSCP						Explicit Congestion Notification	
Osztály választó			Eldobási valószínűség			ECN	
DS5	DS4	DS3	DS2	DS1	DS0	ECN1	ECN0

A 2474-es RFC-ben a fenti táblázatban látható módon határozták meg a TOS mező 8 bitjét, melyből az első 6 bit vonatkozik a DSCP értékre. Ez a 6 bit két részre bontható, az első része (DS5-DS2) megadja az elsőbbségi sorrendet (a csomag prioritását), ezzel definiál egy szolgáltatási osztályt (CoS – Class of Service). A DSCP érték második része a csomageldobási sorrendet határozza meg.

A következő minőségi osztályokat különböztethetjük meg:

1. Best Effort PHB (alapértelmezett)
2. Class Selector PHB (csoportválasztó)
3. Assured Forwarding PHB (AF, Biztosított továbbítás)
4. Expedited Forwarding PHB (EF, Akadálytalan továbbítás)

Az egyes osztályok esetén különbözőek az elvárásaink a következőeket illetően:

-sávszélesség

-késleltetés

-ingadozás

-csomagvesztés

A DSCP AF értékek:

	Osztály 1 **AF1x**	**Osztály 2** **AF2x**	**Osztály 3** **AF3x**	**Osztály 4** **AF4x**
Alacsony eldobási prio. AFx1	**001**010 DSCP 10 AF11	**010**010 DSCP 18 AF21	**011**010 DSCP 26 AF31	**100**010 DSCP 34 AF41
Közepes eldobási prio. AFx2	**001**100 DSCP 12 AF12	**010**100 DSCP 20 AF22	**011**100 DSCP 28 AF32	**100**100 DSCP 36 AF42
Magas eldobási prio. AFx3	**001**110 DSCP 14 AF13	**010**110 DSCP 22 AF23	**011**110 DSCP 30 AF33	**100**110 DSCP 38 AF43

Először a magas eldobási szinthez tartozó csomagok kerülnek eldobásra, mielőtt a közepes majd az alacsony szinthez tartozó csomagokat dobnánk el. Ez egyfajta finomhangolásra ad lehetőséget egy adott osztályon belül. A fenti DSCP AF értékek a 2597-es RFC-ben találhatóak meg.

DSCP EF (Expedited Forwarding)

Az „express továbbítás” (RFC 2598) esetén az ezen osztályban lévő csomagok garantált sávszélességet kapnak az egyéb hálózati forgalomtól függetlenül. Mivel a csomagvesztés, a késleltetés és a késleltetés ingadozása a hálózatban fellépő torlódások következménye, ezért amennyiben az osztályba sorolt csomagok nem tapasztalnak sorbaállást az átvitel során, ezért egy bérelt vonalhoz hasonló szolgáltatást képes nyújtani. Azonban mindez csak akkor működik, ha a közbenső csomópontok mindegyikén az ezen osztályhoz tartozó beérkező forgalom kisebb, mint az ugyanezen csomópontban ugyanezen osztályhoz tartozó kimenő

sávszélesség. A végpontok számára ez a szolgáltatás pont-pont kapcsolatnak látszik, azaz olyan, mint egy "virtuális bérelt vonal".

A VoIP adatforgalmat DSCP EF értékkel jelölik meg. A szakértők különféle véleményen vannak, azonban átlagosan azt javasolják, hogy a Voice átvitelre leválasztott sávszélesség ne legyen több a teljes sávszélesség 30%-ánál.

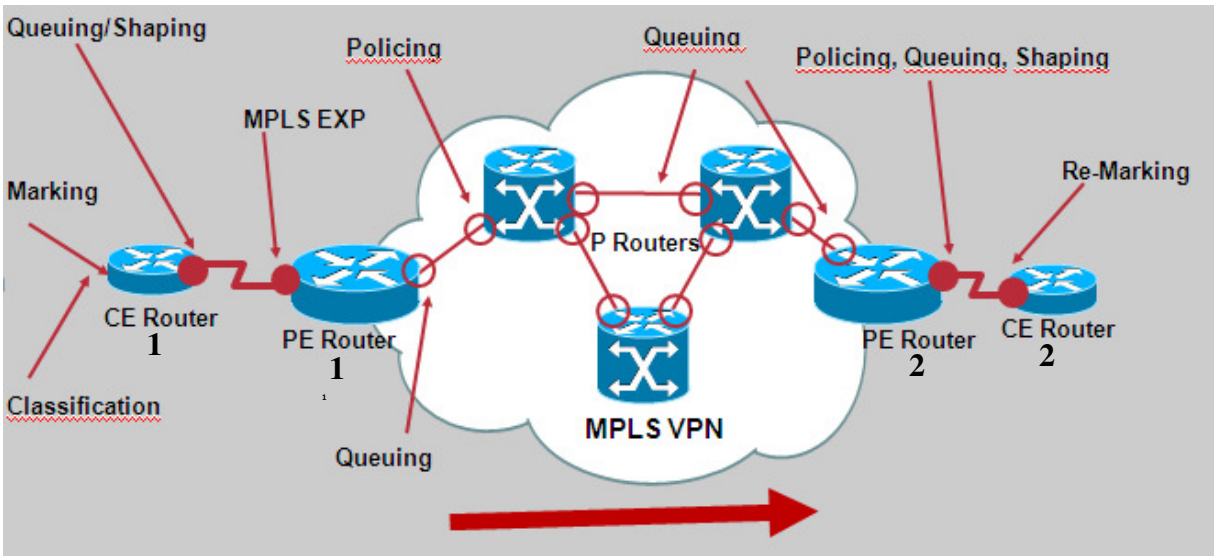
Példa az osztályok kialakítására:

		Optimalizációs paraméterek				
Szolgáltatási osztály	**Tipikus alkalmazás**	**Sávszél.**	**Késlelt.**	**Ingadoz.**	**Csom. ve.**	**DSCP**
Általános	E-mail, FTP, HTTP	X	-	-	-	BE
Alkalmazás	ERP, Telnet	X	-	-	X	AF21
Valós idejű	Interaktív adatforg.	X	X	-	X	AF41
Hang	Voice over IP	X	X	X	X	EF
Hálózat vezérlés	Hálózat Management	X	-	-	X	CS6

A hálózat menedzseléséhez azért alkalmaznak külön osztályt, mert a hálózati eszközök elérését biztosítani kell olyan esetekben is, amikor a hálózat extrém módon túlterheltté válik.

4.4.2 QoS alkalmazása

Tekintsük a következő ábrát, melyen a nyíl az adatforgalom irányát jelöli.



28. ábra

- **Marking** (Csomagok megjelölése): az 1-es CE router-be kétféleképpen érkezhetnek meg az IP csomagok. Lehetséges, hogy a LAN oldalról az egyes IP csomagok már el vannak látva a megfelelő DSCP értékkel. (Például: egy VoIP telefon képes lehet arra, hogy az általa generált IP csomagoknak a TOS mezőjét DSCP EF-re állítsa.) Amennyiben ez nem történik meg, úgy nekünk kell gondoskodnunk a csomagok megjelöléséről. Ezt megtehetjük például ACL-ek segítségével.

- **Classification** (Osztályba sorolás): amikor az 1-es CE routerbe IP csomagok érkeznek, gondoskodni kell róla, hogy a megtörténjen a csomagoknak a különböző osztályokba való besorolása valamilyen előre definiált szabályrendszer szerint. Ezután alkalmazható a DiffServ PHB. Az adatforgalom a CE routert elhagyva, a PE routerbe lép a DSCP értéknek megfelelő osztályban. Ezt követően az IP forgalom az MPLS felhő irányába halad tovább, tehát az osztályozásra vonatkozó információt az MPLS címkében kell letárolni, erre az MPLS EXP bitek adnak lehetőséget.

Egy lehetőség az MPLS EXP bitek megfeleltetésére:

	DSCP Edge	MPLS EXP Core
Szolgáltatási osztály		MPLS Core
Általános	BE	0
Alkalmazás	AF41	2
Valós idejű	AF21	7
Hang	EF	5
Hálózat vezérlés	CS6	6

A forgalom az MPLS hálózaton az MPLS EXP bit vizsgálatával történik, majd az újabb PE router elérése után az EXP bitek leválasztásra kerülnek és a DSCP érték alapján folytatódik tovább az adatforgalom továbbítása. Jellemzően mindig a CE forgalomirányítókon kerülnek beállításra a megfelelő DSCP értékek.

A kimenő interfészeken a következő módszerek használatosak:

- **Shaping**: az adott interfészhez a kívánt VPN CAR (Committed Access Rate) hozzárendelését végzi. Ez az a sávszélesség, amiért a felhasználó a szolgáltatónak fizet. Nem összekeverendő a policing-al, ugyanis a shaping egy buffert használ a

hálózati forgalom kisimítására, hogy a megadott sávszélesség átvitele megoldható legyen. Kissé visszatartva kiegyenlíti illetve eldobja a többletforgalmat.

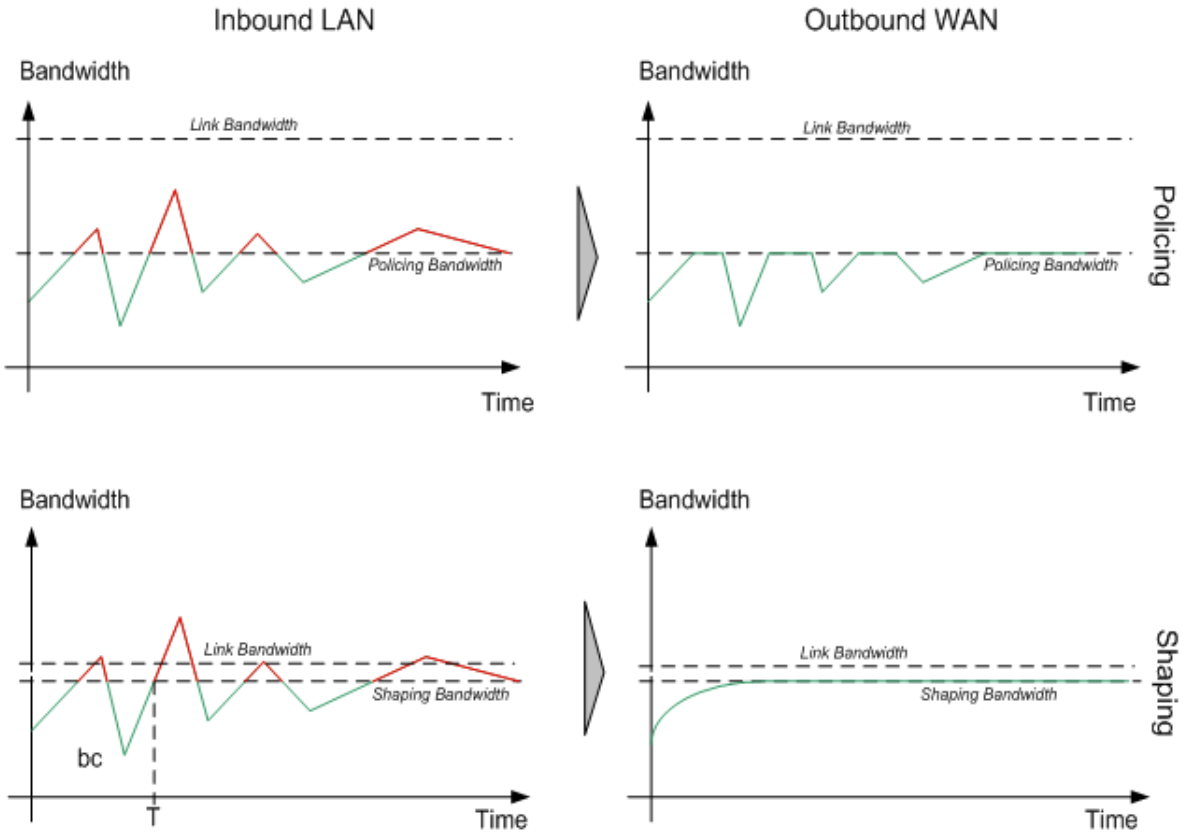
- **Scheduling** (ütemezés): akkor kerül elő, amikor különböző osztályokban lévő csomagokat kell továbbítani a megfelelő szolgáltatási osztálynak megfelelően, mely az adott interfész QoS profiljában van definiálva. Meghatározza, hogy az egyes csomagok hogyan hagyják el az eszközt, melyiket kerüljön továbbításra leghamarabb. Akkor is megtörténik az ütemezés, ha nem tapasztalható torlódás.

- **Queuing** (sorbarendezés): abban az esetben, ha egy hálózati eszköznél az adatforgalom szempontjából a bemenő interfész nagyobb átviteli sebességet tesz lehetővé, mint a kimenő interfész, torlódás, szűk keresztmetszet keletkezhet. Az eszközök pufferekkel rendelkeznek, hogy átmenetileg tárolni, majd ezt követően ütemezni tudják ezeket a feltorlódott csomagokat is. A queuing tehát csak abban az esetben funkcionál, ha torlódás jelentkezik valamelyik interfészen. A queuing és a scheduling egymást kiegészítő, összefüggő folyamatok.

  A scheduling és a queuing konkrét működésére többfajta módszer is ismeretes.

- **Policing**: a beállított sávszélesség túllépése esetén eldobja a csomagokat. Bizonyos esetekben szükségünk van erre, például valós idejű, VoIP adatátvitel esetén. Ezt a típusú forgalmat korlátozni kell egy policing módszerrel, mielőtt üzemezésre kerülne sor.

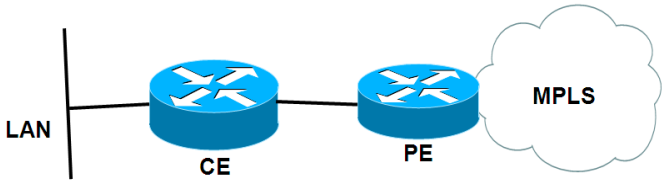
A következő ábrán a shaping és a policing közötti különbséget láthatjuk:



29. ábra

4.4.3 Példák

A következőkben néhány egyszerűen megvalósítható, ugyanakkor hatékony és napjainkban is használt módszert tárgyalunk a QoS konfigurálását illetően.



30. ábra

4.4.3.1 Classification és marking

Amennyiben a CE routerbe beérkező IP csomagok nincsenek megjelölve DSCP értékekkel, ACL-ek segítségével tudjuk meghatározni az osztályokba történő besorolást.

1. *ACL-ek létrehozása.*

```
ip access-list extended cos-map-acl-fe-0.0-voice
  …

ip access-list extended cos-map-acl-fe-0.0-business
  …
```

2. *Class-map-ek létrehozása.*

```
class-map match-any cos-map-fe-0.0-voice
  match access-group name cos-map-acl-fe-0.0-voice

class-map match-all cos-map-fe-0.0-business
  match access-group name cos-map-acl-fe-0.0-business
```

3. *Service-policy-k létrehozása.*

```
policy-map input-policy-fe-0.0
```

```
 class cos-map-fe-0.0-voice
  set ip dscp ef

 class cos-map-fe-0.0-business
  set ip dscp af21

 class class-default
  set ip dscp default
```

4. Policy alkalmazása a CE routeren a LAN interfészre.

```
interface fastethernet0/0
  service-policy input input-policy-fe-0.0
```

Manapság a leggyakrabban használt marking „eszköz” a fenti példában látható osztály alapú marking illetve osztály alapú policing. Miután a service policy-t input irányban alkamaztuk a LAN interfészre (fastethernet0/0), vizsgáljuk meg, mi történik a gyakorlatban.

Beérkezik egy csomag a LAN interfészen, az itt alkalmazott policy az `input-policy-fe-0.0` Ezután a policy-map –ben található class-okat vizsgálja meg a router, mindaddig, amíg egyezést nem talál. A `class class-default` osztályra legvégül minden csomag illeszkedik, amennyiben ez az előtte felsoroltak közül (`cos-map-fe-0.0-voice, cos-map-fe-0.0-business`) egyikre sem sikerült, ekkor az alapértelmezett DSCP érték kerül beállításra.

A policy-map-en belüli egyes class-okhoz tartozik egy-egy class-map. Ez a class-map pedig egy nevesített ACL-t tartalmaz. Így ha az adott ACL-re illeszkedik az IP csomag, akkor a hozzá tartozó osztály alapján kerül beállításra a csomag header-ben a TOS mező.

Tehát ha például ha a `cos-map-acl-fe-0.0-business` elnevezésű ACL a következőképp néz ki:

```
ip access-list extended cos-map-acl-fe-0.0-business
  permit ip any any
```

Mivel ez az ACL az összes IP csomagra illeszkedik, ezért így az összes csomag például DSCP AF21 értékkel lesz megjelölve.

Korábban már megemlítettük, hogy bizonyos esetekben a megfelelő csomagok már el vannak látva az általunk ismert DSCP (vagy akár IP Precedence) értékekkel. A fenti konfigurációs példa ebben az esetben mindössze annyiban módosul, hogy nem használunk ACL-eket, a class-map részekbe pedig az ACL-re való illeszkedés helyett magát a DSCP értéket vizsgáljuk:

```
class-map match-any cos-map-fe-0.0-voice
  match ip dscp ef
class-map match-all cos-map-fe-0.0-business
  match ip dscp af21

policy-map input-policy-fe-0.0
   class cos-map-fe-0.0-voice
   class cos-map-fe-0.0-business
   class class-default
     set ip dscp default

interface fastethernet0/0
  service-policy input input-policy-fe-0.0
```

4.4.3.2 Policing, shaping, queuing

A következő példában a CE router kimenő interfészére koncentrálunk, és egy példán keresztül bemutatjuk a shaping, policing, queing fogalmak egy lehetséges alkalmazási módját.

```
class-map match-any cos-map-dscp-vo-1
  match ip dscp ef
class-map match-any cos-map-dscp-nm-1
  match ip dscp cs6
```

```
class-map match-any cos-map-dscp-vs-1
  match ip dscp cs3 af31
!
policy-map cos-po2-ge-0.1
  class cos-map-dscp-vo-1
    priority 676 6760
    police 676000 conform-action transmit exceed-action drop
  class cos-map-dscp-vs-1
    bandwidth 34
    queue-limit 20
  class cos-map-dscp-nm-1
    bandwidth 138
    queue-limit 24
  class class-default
    bandwidth 1997
    queue-limit 334
!
policy-map cos-po1-ge-0.1
  class class-default
    shape average 3072000
    service-policy cos-po2-ge-0.1
!
interface GigabitEthernet 0/1
  service-policy output cos-po1-ge-0.1
!
```

A CE router PE felé menő interfésze a GigabitEthernet0/1, erre alkalmazzuk kimenő irányban az előre definiált policy-t. A shape average parancs jelentőségéről már volt szó az előzőekben, ennek a szemléltetéséhez a konfigurációban úgynevezett hierarchikus policy-kat hozunk létre, majd a shape avergage értéket a szülő policy alá állítjuk be.

Az egyes osztályokhoz a sávszélességeken kívül még egy ún. queue-limit paramétert is alkalmaztunk az osztályra a fenti példában. Mivel a queue-k mérete nem végtelen,

megtelhetnek és túlcsordulhatnak. Amikor egy queue tele van, több csomag nem kerül bele, eldobásra kerül. Azért, hogy ez ne forduljon elő, alkalmazhatjuk a queue-limit parancsot, ezzel megnövelve a puffer méretét.

Hierarchikus QoS Policy és Shaping

```
WAN interfész
       Service-policy out parent (szülő)

policy-map parent (szülő)
  class class-default
    shape average cir
    service-policy child (gyerek)

policy-map child (gyerek)
    class VO
    class VS
    class EC
```

Amikor a LAN irányából érkező IP csomagok elérik a CE router WAN interfészét, már be van állítva a header TOS mezője a megfelelő értékre. Most a feladatunk az, hogy a DSCP értékek szerint „szétválogatva" minden csomag a neki megfelelő osztályban, ill. queue-ban folytassa útját a PE router felé.

Ahhoz, hogy a csomagok a nekik megfelelő osztályokba kerüljenek a match parancsot használjuk a class-map alatt az összes csomagra.

A priority parancs alacsony késleltetésű queuing-ot valósít meg az erre érzékeny adatforgalmak számára, mintpéldául VoIP. Minimális garantált sávszélességként nevezhetjük. A police parancs a maximális adatátviteli sebességet állítja be, amelyet a VoIP osztályban alkalmazunk szintén.

5. Hibatűrő ügyfélhálózatok megvalósítása: HSRP

A fejezet az RFC2281-es szabványdokumentum alapján készült. Mivel ez a protokoll önmagában is egy Cisco specifikus protokoll a fejezet leírása során a Cisco által használt fogalmak, megvalósítások is jobban előtérbe kerülnek.

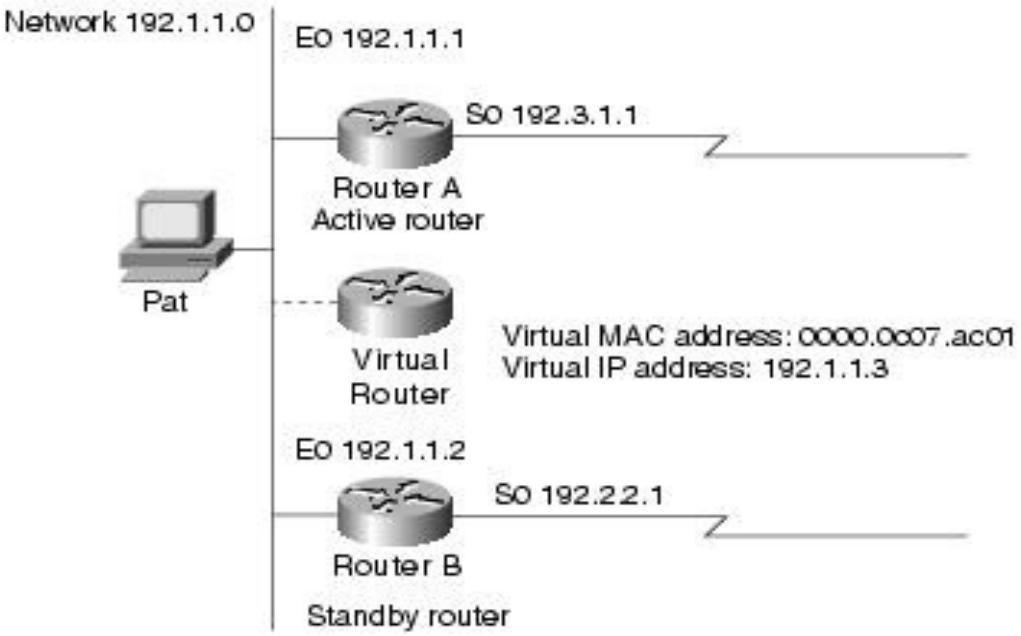
5.1 HSRP fogalma

A HSRP (Hot Standby Routing Protocol) hibatűrő IP hálózatokat építésére szolgáló protokoll, mely a Cisco által került implementálásra. A HSRP azon hostok esetében, melyek nem képesek megtanulni dinamikusan az alapértelmezett átjáró címét, megoldást nyújt az átjáró meghibásodásának kiküszöbölésére. A protokoll alkalmazható többszörös hozzáférésű (multi-access), multicast vagy broadcast képes LAN-okra.
A HSRP működése során forgalomirányítók egy csoportja létrehoz egy virtuális útválasztót (átjárót), a helyi hálózaton lévő hostoknak, ezt a csoportot HSRP vagy Standby csoportnak nevezzük. Egy adott Standby csoportba tetszőleges számú forgalomirányító tartozhat, melyek közül egy aktív és egy készenléti útválasztó kerül kiválasztásra. Az aktív lesz a felelős a csomagtovábbításért. Az aktív forgalomirányító meghibásodása esetén a készenléti átveszi a helyét és az aktív forgalomirányító csomagtovábbító szerepét. Bármennyi útválasztó is van a csoportban, csak az aktív továbbítja a csomagokat.
A választási procedúra lejátszódása során, számos HSRP csomag kering a HSRP csoportban szereplő forgalomirányítók között, de a folyamat végeztével csak az aktív és a készenléti útválasztók küldenek egymásnak HSRP csomagokat. Ezzel a hálózati forgalom jelentősen csökken.
Egy LAN-on lelet több HSRP csoport is. Ebben az esetben minden csoport emulál egy virtuális útválasztót, amelyhez hozzárendel egy fizikai MAC és egy IP címet.
Definíciók:

- Aktív útválasztó: Az a forgalomirányító, amely aktuálisan továbbítja a csomagokat a virtuális forgalomirányítónak
- Készenléti (Standby) útválasztó: elsődleges backup forgalomirányító. Az aktív útválasztó meghibásodása esetén ez a forgalomirányító lesz az aktív csomagtovábbító eszköz.

- Standby csoport: Útválasztók egy csoportja, amely létrehoz egy virtuális átjárót a helyi hálózaton lévő hostok számára. Alapértelmezetten a 0–ás Standby csoport konfigurálható, de tetszőleges csoportszámot is megadhatunk a konfigurációs utasításokban.
- Hello time: Az adott HSRP csoportban lévő forgalomirányítók rendszeres időközönként úgynevezett Hello üzeneteket küldenek egymásnak. Ez egyfajta „életjelként" fogható fel az eszközök hibátlan működésére nézve. A hello time a sikeresen fogadott HSRP Hello üzenetek érkezése között eltelt időintervallum.
- Hold time: a Hello üzenetek nyugtázása között eltelt időintervallum, amely után a küldő útválasztót hibásnak tekinti. Azaz ennyi időtartamot vár az útválasztó egy hello üzenet megérkezésére.

Nézzünk egy példát:



31. ábra

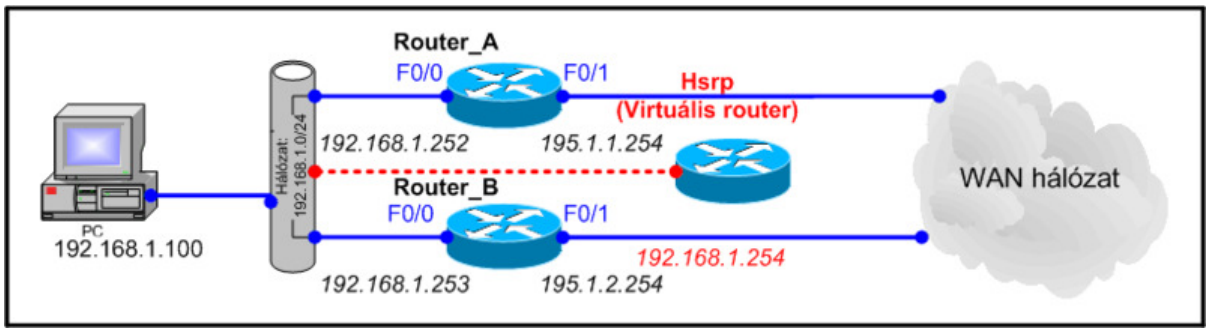
HSRP nélkül a Pat nevű számítógép alapértelmezett átjárója, legyen a Router A. Amennyiben ez az útválasztó meghibásodik, a munkaállomás elveszti a távoli hálózatok felé vezető kapcsolatát. Ha hibatűrő megoldás alkalmazása mellett döntünk és HSRP-t alkalmazunk a Pat nevű számítógépen az átjáró a virtuális IP cím lesz. Ezt a címet a két útválasztó virtuális útválasztóként (az ábrán Virtual Router) emulálja. Alapértelmezésben a csomagok az aktívként működő Router A felé mennek. Amennyiben ez meghibásodik, a Router B aktívvá válik és ő továbbítja a csomagokat. A munkaállomáson ez a folyamat észrevehetetlen.

Ebben az esetben az alap konfigurációs lépések a következők: Elsőként a **standby ip <IP cím>** paranccsal engedélyezzük a HSRP-t az adott interfészen és egyben definiáljuk a virtuális átjáró címét is. Ezt a címet a HSRP csoport minden tagjában meg kell adnunk.

Ahhoz, hogy egy útválasztó aktív legyen vagy a későbbiekben aktívvá válhasson, ha a prioritása nagyobb a csoportban szereplő többi forgalomirányítóval szemben, ki kell adnunk a **standby preempt** utasítást.

A forgalomirányítók alapértelmezett prioritása 100, melyet a **standby priority** utasítással bírálhatunk felül. Egy csoporton belül a legnagyobb prioritású lesz az aktív forgalomirányító.

Lehetőség van a HSRP csomagok hitelesítésére is a **standby authentication <szöveg>** paranccsal. Hitelesítés alkalmazása esetén a hitelesítő szöveg bekerül a HSRP multicast csomagokba ezért a csoport minden tagjában alkalmazni kell.



32. ábra

Router_A HSRP konfigurációja:

Interface Fastethernet 0/0

Ip address 192.168.1.252 255.255.255.0

Standby ip 192.168.1.254

Standby preempt

Standby priority 110

Router_B HSRP konfigurációja:

Interface Fastethernet 0/0

Ip address 192.168.1.253 255.255.255.0

Standby ip 192.168.1.254

Standby preempt

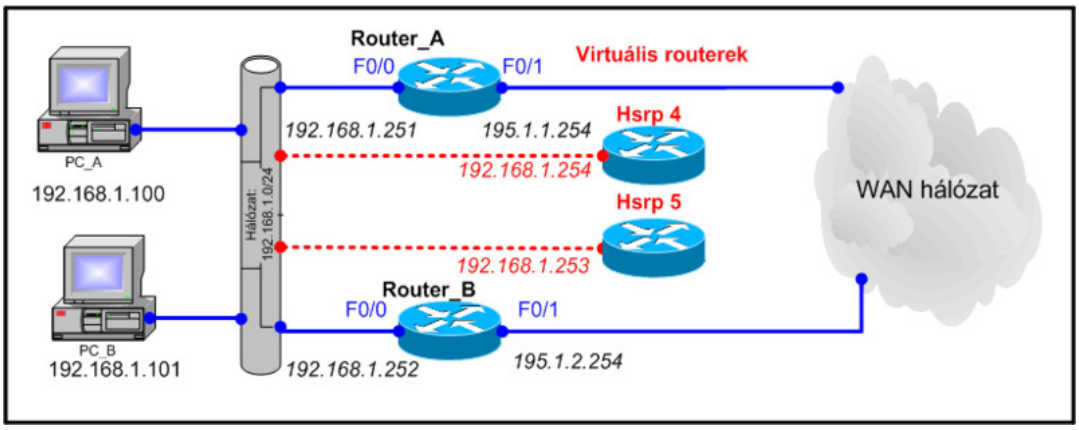
PC konfiguráció:

IP cím: 192.168.1.100

Alhálózati maszk: 255.255.255.0

Alapértelmezett átjáró: 192.168.1.254

Lehetőség van a HSRP terheléselosztásos konfigurálására is. Ebben az esetben két HSRP csoportot hozunk létre fordított szerepkörrel (prioritások szabályzásával) mindkét forgalomirányító esetében.



33. ábra

Router_A HSRP konfigurációja:

Interface Fastethernet 0/0

Ip address 192.168.1.251 255.255.255.0

Standby 4 ip 192.168.1.254

Standby 4 preempt

Standby 4 priority 110

Standby 5 ip 192.168.1.253

Standby 5 preempt

Router_B HSRP konfigurációja:

Interface Fastethernet 0/0

Ip address 192.168.1.252 255.255.255.0

Standby 4 ip 192.168.1.254

Standby 4 preempt

Standby 5 ip 192.168.1.253

Standby 5 priority 110

Standby 5 preempt

PC_A konfiguráció:

IP cím: 192.168.1.100

Alhálózati maszk: 255.255.255.0

Alapértelmezett átjáró: 192.168.1.254

PC_B konfiguráció:

IP cím: 192.168.1.101

Alhálózati maszk: 255.255.255.0

Alapértelmezett átjáró: 192.168.1.253

A 4-es HSRP csoport aktív tagja a Router_A és tartalék a Router_B az 5-ös csoportban pedig épp fordítva. A Router_A kiesése esetén a 4-es csoportban a Router_B átveszi az aktív router szerepét és az 5-ösben ugyanúgy aktív marad. A Router_B kiesése esetén a 4-es csoportban a Router_A aktív marad, és az 5-ösben válik aktívvá. A gépek terhelésmegosztás alapú konfiguráláshoz a gépek egyik részét az egyik virtuális útválasztóhoz, másik felét pedig a másik virtuális forgalomirányítóhoz kell hozzárendelni.

5.2 HSRP protokoll

5.2.1 Csomagformátum

A HSRP protokoll az UDP 1985 ös portot használja a kommunikációra és a 224.0.0.2-es csoportos címet 1-es TTL értékkel. A forgalomirányítók a saját IP címüket használják a csomagok forrás címében nem a virtuális címet. Ennek célja, hogy a HSRP forgalomirányítók azonosítani tudják egymást.

HSRP csomag:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Verzió	Opcode	Állapot	Hellotime
Holdtime	Prioritás	Csoport	Foglalt
Hitelesítési adat -			
Virtuális IP cím			

Verzió: 8 bit. HSRP verzió száma.

Opcode: 8 bit. Az üzenet típusát írja le. Lehetséges értékei:

Opcode	Leírás
0	Hello: Az útválasztó működik és képes aktív/készenléti állapotba kerülni
1	Coup: Az útválasztó aktív állapotba kerülésének kezdeményezése
2	Resign: Az útválasztó aktív állapotának megszűnésére irányuló üzenet

Állapot: 8 bit. A HSRP csoport minden útválasztója felépít egy állapot gépet. Ez a mező írja le a küldő útválasztó aktuális állapotát. (*4.2.3 HSRP állapotok*)

Állapot	Leírás
0	Initial: kezdő állapot, jelzi, hogy a HSRP nem fut. Ez az állapot konfigurálás vagy az interface első bekapcsolása után jön létre.
1	Learn: Az útválasztó még nem határozta meg a virtuális IP címet és még nem kapott Hello üzenetet az aktív útválasztótól. Ebben az állapotban az útválasztó üzenetet vár az aktív útválasztótól.
2	Listen: Az útválasztó ismeri a virtuális IP címet, de még sem aktív sem készenléti szerepben nincs. Hallgatja a hello üzeneteket.
4	Speak: Az útválasztó periodikusan küld hello üzeneteket és aktívan részt vesz az aktív/készenléti választáson. Az útválasztó nem léphet ebbe az állapotba, amíg nem ismeri a virtuális ip címet.
8	Standby: Az útválasztó a készenléti útválasztó és időszakosan küld hello üzenetet.
16	Active: Az útválasztó az aktív útválasztó és periodikusan küld hello üzenetet. Ebben az állapotban lévő forgalomirányító továbbítja a csomagokat

Hellotime: 8 bit.

Hello üzeneteknél értelmezett. Definiálja a forgalomirányítók által küldött Hello üzenetek között eltelt időtartamot. Az idő másodpercben értendő. A hellotime alapértelmezett értéke 3 másodperc. Ez az érték felüldefiniálható illetve konfigurálás hiányában a Hello üzenetekből tanulható az aktív forgalomirányító által. Az a forgalomirányító, amely Hello üzenetet küldd az általa használt Hellotime értéket köteles a csomag Hellotime mezőjébe beszúrni.

Holdtime: 8 bit.

Szintén csak Hello üzenetek kapcsán értelmezhető és másodpercben értendő. Azt az időtartamot határozza meg, ameddig az aktuálisan kapott Hello üzenet érvényesnek tekinthető. Ugyanúgy konfigurálható és tanulható, mint a Hellotime időköz. Értéke általában a hellotime háromszorosa, de mindig nagyobbnak kell lennie annál. Alapértelmezetten 10 másodperc.

Aktív állapotban lévő forgalomirányító sosem tanulhat hellotime vagy holdtime értékeket más forgalomirányítóktól, de kézzel konfigurálható. Tehát vagy az előzőleg aktív

forgalomirányítótól tanult értéket vagy a kézzel konfigurált értéket használja és helyezi el a megfelelő mezőbe.

Prioritás: 8 bit
A mezőt az aktív és a készenléti útválasztó kiválasztásához használatos. Amennyiben két útválasztó versenyez, akkor a magasabb prioritású nyer. Azonos prioritás esetén a magasabb IP című nyer.

Csoport: 8 bit
A mező azonosítja a HSRP csoportot. Token Ring esetén a 0-2 közti értékek, egyéb esetben a 0-255 közti értékek lehetségesek.

Foglalt: 8 bit

Hitelesítési adat: 8 bit
A mező 8 karakteres cleartext jelszót tartalmaz. Amennyiben nincs hitelesítési adat beállítva az ajánlott alapértelmezett érték a következő: 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00.
Virtuális IP cím: 32 bit
A csoport virtuális IP címe. Amennyiben a virtuális IP cím nincs beállítva, az útválasztó megtanulhatja az aktív útválasztó Hello üzeneteiből.

5.2.2 Időzítők

Minden útválasztó 3 időzítőt tart fenn:

- Active timer: Az aktív útválasztó monitorozására szolgál. Újraindul minden esetben amikor authentikált hello üzenet jelenik meg az aktív forgalomirányítón.
- Standby timer: Készenléti útválasztó monitorozása. Újraindul minden esetben amikor authentikált hello üzenet jelenig meg a készenléti forgalomirányítón.
- Hello Timer: Ez az időzítő a Hellotime periódusban meghatározott időközönként jár le. Ha a forgalomirányító Speak, Standby vagy Active állapotban van a Hello timer lejártáig generál Hello üzeneteket.

6. Backup megoldások

A VPN-ek esetében az egyik legfontosabb az adatátvitel biztonságának megteremtése, miközben a nyilvános hálózaton halad keresztül. Természetesen meg kell akadályoznunk az illetéktelen hozzáférést, valamint a napjainkban folyamatosan növekvő biztonsági igényeknek is meg kell felelnünk.

A TCP/IP nem ad lehetőséget a csomagok titkosítására és azonosítására.

6.1 IPSec (Internet Protocol Security)

6.1.1 Az IPSec elméleti háttere

Az IPSec egy protokollcsalád, melyet az IETF fejlesztett ki abból a célból, hogy az IP alapú csomagkapcsolt hálózatokban biztonságosan lehessen különféle szolgáltatásokat használni. A legismertebb ilyen hálózat az Internet. Két távoli telephely közötti kommunikáció során az IPSec lehetővé teszi az információ sértetlenségét a titkosítás és hitelesítés mellett.

Egy vállalat számára jelentős költségmegtakarítást jelenthet egy bérelt vonalas kapcsolathoz képest. Az IPSec az egyik legszélesebb körben elterjedt VPN megoldás, melynek használatakor lehetőség nyílik az IP csomagokban a ToS bitek változtatására is, így a megfelelő szolgáltatás minőség megadása is elvégezhető a kommunikáció során.

Biztosítva van továbbá, hogy a későbbiekben újabb algoritmusokat alkalmazzunk, ha a jelenlegi módszereink már elavulnak. Nagy előnye, hogy további biztonsági eljárásokkal lehet kiegészíteni és mivel szabványokon alapul, a különböző IPSec alapú megoldások kompatibilisek egymással.

6.1.2 Az IPSec működése

A kódolási rendszerben valamilyen módon megoldást kell találni a kulcsok cseréjére. Az IPSec esetében ezt az IKE (Internet Key Exchange) algoritmus oldja meg. Kezeli és elosztja a kulcsokat, továbbá beállítja az SA-t (Security Association), azaz a kapcsolat paramétereit.

Az IKE működésének két fázisa van:

1. A két végpont hitelesíti egymást

- Magunknak állítjuk be a kulcsokat a VPN végpontjaiban, tulajdonképpen itt az IKE nincs használva.
- PSK (Pre Shared Key) használatakor is be kell állítanunk manuálisa egy kulcsot, ezt csak az elsődleges hitelesítéshez használják a felek. Ezt követően új kulcsok generálására kerül sor, melyeket megosztanak egymással a résztvevők és időnként meg is újítják őket. A kezdő kulcs nyilvánosságra kerülése esetén a rendszer összes pontján le kell cserélni a kulcsokat.
- Tanúsítványokat kapnak a résztvevő felek, ezzel hitelesítik magukat.

2. IPSec működésének paramétereit egyezteti a felek közt

- átvitel módja transport vagy tunnel
- kódolási módszerek, protokollok
- VPN kapcsolat vége (idő- vagy forgalomkorlát)

Mindezeket kapcsolatonként az SA-ban tárolja.

A kulcscserén kívül a forgalom védelmét is biztosítja az IPSec, erre alkalmas az AH (Authentication Header) protokoll. Egy hash függvény segítségével lenyomat készül a csomagról, majd miután a csomag célba ért, a lenyomat újbóli elkészítésével eldönthető, hogy sértetlen maradt-e az átvitt adat. Az AH utódja az ESP (Encapsulaing Security Protocoll) protokoll, mely az előbbieken túl titkosításra is képes a DES, 3DES ill. AES algoritmusok segítségével.

Transzport módban az AH/ESP fejléc az IP csomag eredeti fejléce mögé kerül be.

Tunnel módban teljesen új IP csomag jön létre, amelynek új fejléce van, ezt követi az AH/ESP fejléc, majd az eredeti IP csomag. Ezáltal lehetőség van arra, hogy például a routerek IPSec proxy funkciót lássanak el, ami azt jelenti, hogy a hostok helyett ők végzik el a titkosítást, illetve a dekódolást. A kliens gépeken nem szükséges IPSec-hez kapcsolódó semmilyen feldolgozás, csak az IPSec átjáró elérését kell biztosítani. A támadó nem tudja,

hogy hova lettek címezve a csomagok, mindössze azt ismeri, hogy mely két átjáró között haladt át.
Az IPSec IOS támogatottsága a Cisco routerek széles köréhez megtalálható.

6.1.3 Példa IPSec konfiguráció

Konfigurációs lépések:

- Transform-set konfigurálása: az alkalmazott biztonsági protokollokat és algoritmusokat írja le.
- Crypto ACL definiálása: itt adható meg, hogy milyen hálózati forgalom legyen titkosítva.
- Crypto map konfiguráció: egymáshoz rendelhetőek a IPSec beállítások, például: társ végpont megadása, transform set hozzárendelés, ACL társítás, kulcsgenerálás módja.
- Crypto map hozzárendelése az interfészhez

```
crypto ipsec transform-set tset esp-aes esp-sha-hmac
crypto map VPN1 10 ipsec-isakmp
 set peer 199.1.1.9
 set transform-set tset
 match address 101
 set pfs group5
access-list 101 permit ip 113.0.0.0 0.255.255.255 213.0.0.0
0.255.255.255

interface FastEthernet0/0
 ip address 199.1.1.10 255.255.255.252
 crypto map VPN1          <--- alkalmazás az interfészre
```

6.2 ISDN Backup

Az ISDN-t (Integrált Szolgáltatású Digitális Hálózat / Integrated Services Digital Network) kis sávszélesség igények kielégítésére tervezték, azzal a céllal, hogy egy digitális hálózatot hozzanak létre a meglévő telefonos hálózaton. Az ISDN-t nem állandó összeköttetésnek terveztek, hanem úgy, hogy a kommunikáció előtt felépít egy WAN összeköttetést, annak befejeztével pedig lebontja azt. Ezzel költséghatékony megoldás, mivel nem teljes idejű szolgáltatás, csak az adatforgalom idejére jön létre a kapcsolat.
Ezen tulajdonságainak hála, kiválóan alkalmazható backup megoldásként.
A fővonal esetleges kiesése esetén, felépül az ISDN kapcsolat, amely ha kisebb teljesítményű is, de a fővonal helyreállításáig képes az adatforgalmat korlátozottan biztosítani.
Ez optimális megoldás lehet, hiszen az ISDN vonalnak csak a fővonal kiesése alatt lesz költségvonzata, ezzel elkerülve azt, hogy egy backup megoldásként működő állandó összeköttetést tartsunk fent.
Az ISDN csatornák szabványosított típusai: [Tannenbaum]

- A – 4kHz-es analóg telefoncsatorna
- B – 64kbit/s-os hangcsatorna, amely adatátviteli célokra is alkalmas
- C – 8kbit/s-os vagy 16kbit/s-os digitális csatorna
- D – 16kbit/s-os vagy 64kbit/s-os digitális csatorna sávon kívüli jelzésre
- E – 64kbit/s-os digitális csatorna belső ISDN jelzésre
- H – 384kbit/s-os, 1536kbit/s-os vagy 1920kbit/s-os digitális csatorna

Ezeket a csatornákat nem lehet tetszés szerint felhasználni, eddig három kombinációt szabványosítottak:

- Basic Rate Interface (alapsebesség): 2B + 1D, tehát 2*64kbit/s + 16kbit/s jelzésre
- Primary Rate Interface (primer sebesség): 23B + 1D (Egyesült Államok és Japán – 1,544Mbit/s) vagy 30B + 1D (Európa – 2,048Mbit/s), ahol a D csatorna 64kbit/s-os
- Hibrid: 1A + 1C

Kis sávszélesség igény esetén az alapsebességű ISDN, nagyobb igény esetén a primer sebességű alkalmazható backup megoldásként.

Tehát amikor az állandó összeköttetés a PE és a CE router között valamilyen okból megszakad, akkor a CE router ISDN összeköttetést épít fel egy ISDN képes PE routerrel, így a CE router továbbra is az MPLS hálózathoz kapcsolódik.

Az adatkapcsolat létrehozásánál leggyakrabban PPP beágyazást (Point-to-Point Protocol) használnak, melynek nagy előnye, hogy a hitelesítésre is megfelelő választ ad. A PPP két beépített biztonsági szolgáltatást nyújt, mégpedig a PAP (Password Authentication Protol) jelszó hitelesítő protokollt, és CHAP (Challange Handshake Authentication Protocol) kérdés-felelet hitelesítési protokollt. A CHAP erősebb hitelesítés, amely háromfázisú kézfogás módszerét alkalmazza, és a jelszavakat nem egyszerű szövegként továbbítja.

Amennyiben több B csatorna áll rendelkezésre, (lehetőség van több BRI interfész használatára is) akkor a multilink PPP képes a sávszélesség igények kezelésére, újabb B csatorna felépítésére vagy lebontására.

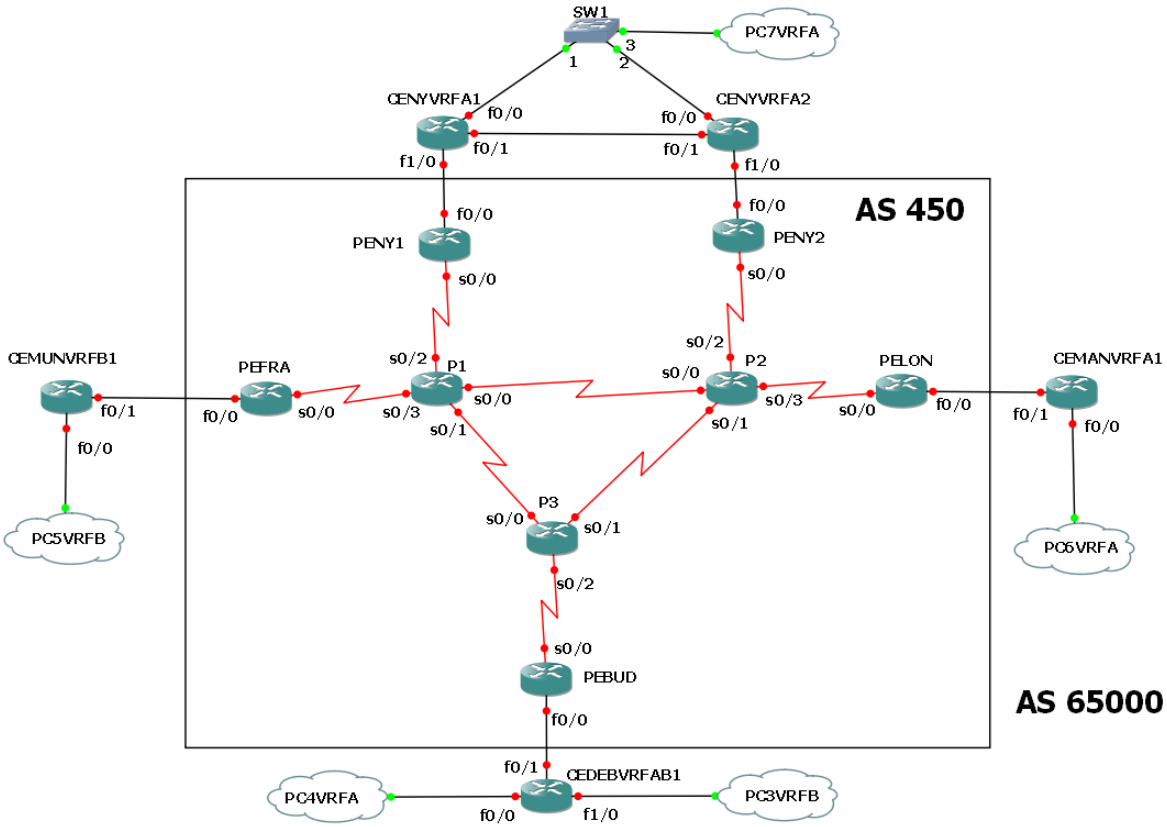
7. Reprezentatív modell bemutatása

A dolgozatunkban számos - napjaink szolgáltatói által használt - technológiát, módszert ismertettünk. Szükségét éreztük ezen módszerek lehető legszélesebb módon történő gyakorlati bemutatását a rendelkezésünkre álló eszközök segítségével. Mivel fizikai eszközök nem álltak a rendelkezésünkre, ezért egy szimulációs környezetet használtunk, amelyet a későbbiekben mutatunk be.

Az MPLS topológiát két ügyféllel szemléltetjük, az első virtuális magánhálózatát VPNA, a másodikat pedig VPNB jelöli. A topológia megtervezésénél egy napjainkban használatos hálózat modellezését tartottuk szem előtt. Az ügyfelek stratégiai partnerek és rendelkeznek egy közös telephellyel Magyarországon Debrecenben, ahol mindkét virtuális magánhálózat megtalálható. Továbbá a VPNA ügyfél telephellyel rendelkezik az Amerikai Egyesült Államokban, New Yorkban és Nagy-Britanniában, Manchesterben. A VPNB-nek még egy központja van Németországban, Münchenben. A telephelyek egymástól földrajzilag távol helyezkednek el, melyet az eszközök elnevezéseivel szemléltetünk. Az MPLS hálózat határán PE forgalomirányítók vannak, amelyek lehetővé teszik az ügyfelek kapcsolódását a szolgáltatói hálózathoz. A PE routerek nagyobb földrajzi egységenként találhatók: Londonban, Frankfurtban, New Yorkban és Budapesten. A későbbi bővítés lehetőségét is biztosítjuk, mivel a PE szolgáltatói eszközökhöz további ügyfél routerek kapcsolhatók. A VPNA ügyfél New Yorkban nagyobb biztonsági szintet követel meg, ezért redundáns topológiát alakítottunk ki, amely egy esetlegesen bekövetkező hiba esetén automatikusan a backup eszközre kapcsol át, biztosítva a zavartalan működést. A VPNB két telephelye között egyes üzleti alkalmazások magasabb prioritási szintet követelnek meg az egyéb hálózati forgalommal szemben, ezért különböző szolgáltatásminőségi szinteket határoztunk meg.

A modell megfelelően példázza napjaink MPLS alapú szolgáltatói hálózatainak működését, amelyet nagy méretekben is hatékonyan alkalmaznak.

7.1 Hálózati topológia, eszköz specifikációk



34. ábra

A hálózat gerincét a P1, P2, P3 forgalomirányítók alkotják, melyek soros vonalakkal vannak összekötve.

A P (Provider – szolgáltatói) routerekhez kapcsolódnak a következő PE (Provider Edge – szolgáltatói határ) routerek szintén soros linkkel:

- PEFRA: PE router Frankfurtban
- PENY1: PE router New Yorkban, amely a VPNA ügyfél fővonalát biztosítja
- PENY2: PE router New Yorkban, amely a VPNA ügyfél tartalék vonalát biztosítja
- PELON: PE router Londonban
- PEBUD: PE router Budapesten

A PE routerekhez FastEthernet vonallal kapcsolódnak az alábbi CE (Customer Edge – ügyfél határ) forgalomirányítók:

- CEMUNVRFB1: CE router Münchenben, mely a VPNB ügyfél kapcsolódását biztosítja
- CENYVRFA1: CE router New Yorkban, mely a VPNA ügyfél fővonalát biztosítja
- CENYVRFA2: CE router New Yorkban, mely a VPNA ügyfél tartalék vonalát biztosítja
- CEMANVRFA1: CE router Manchesterben, a VPNA ügyfél vonalához
- CEDEBVRFAB1: CE router Debrecenben, mely a már említett stratégiai megfontolás miatt mindkét ügyfél kapcsolatát ellátja

A CE forgalomirányítókhoz kapcsolódó hálózati végberendezések:

- PC3VRFB: Debrecenben található VPNB-beli ügyfélszámítógép
- PC4VRFA: Debrecenben található VPNA-beli ügyfélszámítógép
- PC5VRFB: Münchenben található VPNB-beli ügyfélszámítógép
- PC6VRFA: Manchesterben található VPNA-beli ügyfélszámítógép
- PC7VRFA: New Yorkban található VPNA-beli ügyfélszámítógép
- SW1: A New Yorkban található helyi hálózati eszköz

Működési mechanizmusukat a későbbiekben részletesen tárgyaljuk.

A BGP protokoll tárgyalásánál említett autonóm rendszer (AS) fogalmát mutatják az ábrán jelölt „AS450” és „AS65000” azonosítók. A szolgáltató eszközei, azaz a P és PE routerek alkotják az AS450-et. Az ügyfél oldali eszközök az AS65000-be tartoznak.

A hálózat modellezéséhez a következő eszközöket és szoftvereket (IOS – Internetwork Operating System, Hálózati operációs rendszer) verziókat használtuk fel:
P és PE routerek esetében Cisco 3725-ös eszközöket használtunk, míg ügyfél oldalon Cisco 2621-es forgalomirányítók kerültek alkalmazásra. A felhasznált IOS verziók P és PE oldalon C3725-SPSERVICESK9-M , CE oldalon C2600-IS-M. Ezeket a show version parancs kimenetekben láthatjuk:

PE_FRA#show version

Cisco IOS Software, 3700 Software (C3725-SPSERVICESK9-M), Version 12.3(11)T5, RELEASE SOFTWARE (fc1)

…

PE_FRA uptime is 23 hour, 14 minutes

…

Cisco 3725 (R7000) processor (revision 0.1) with 124928K/6144K bytes of memory.

R7000 CPU at 240MHz, Implementation 39, Rev 2.1, 256KB L2, 512KB L3 Cache

2 FastEthernet interfaces

6 Serial(sync/async) interfaces

DRAM configuration is 64 bits wide with parity enabled.

55K bytes of NVRAM.

16384K bytes of ATA System CompactFlash (Read/Write)

Configuration register is 0x2102

ce_mun_vrfb-1#sh ver

Cisco Internetwork Operating System Software

IOS (tm) C2600 Software (C2600-IS-M), Version 12.3(25), RELEASE SOFTWARE (fc1)

…

ce_mun_vrfb-1 uptime is 21 hour, 15 minutes

…

cisco 2621 (MPC860) processor (revision 0x202) with 59392K/6144K bytes of memory.

2 FastEthernet/IEEE 802.3 interface(s)

128K bytes of non-volatile configuration memory.

8192K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102

7.2 GNS3 (Graphical Network Simulator) környezet

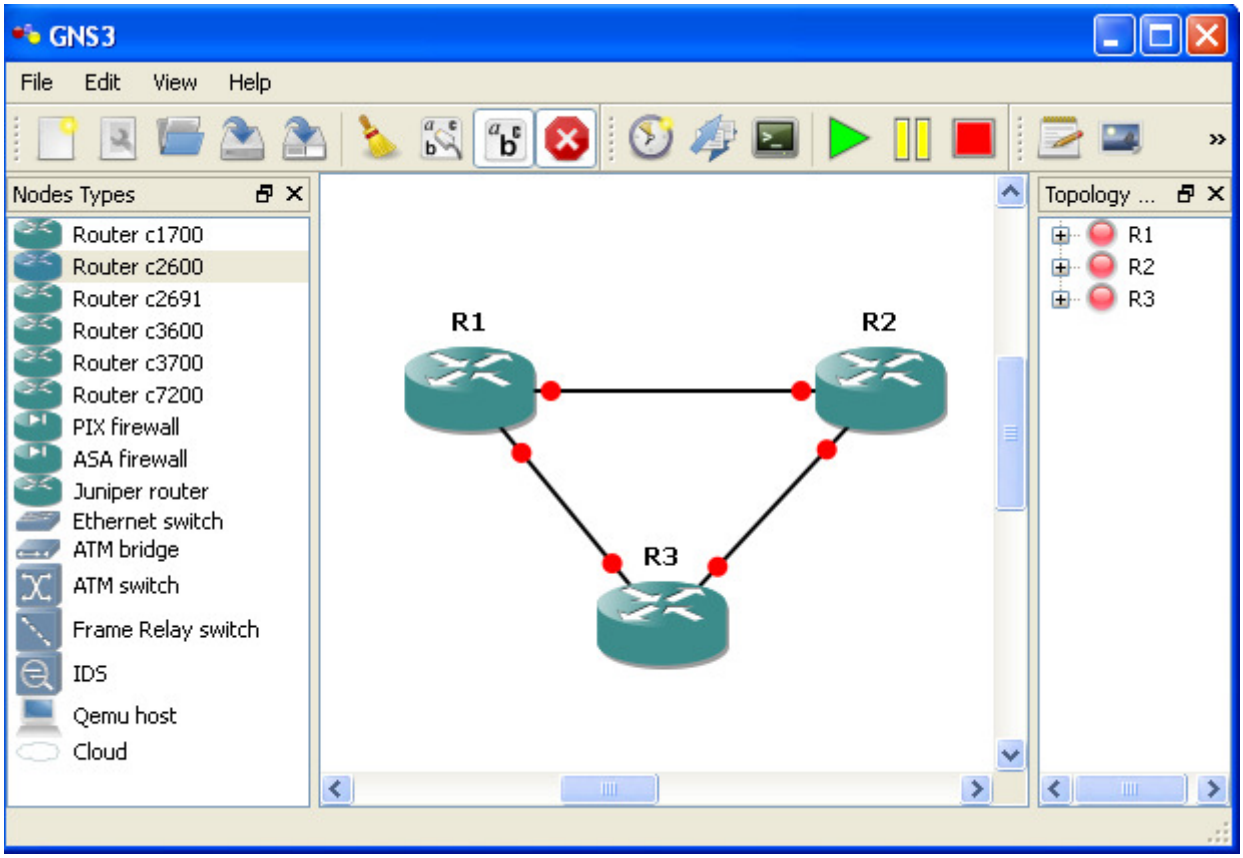
Az alapvető hálózati megoldások modellezésére alkalmas Packet Tracer program eszközkészlete a dolgozatban tárgyalt témák megvalósításához nem nyújt támogatást. Ezért jóval komplexebb lehetőségeket nyújtó környezetet kellett keresnünk. Így esett választásunk a GNS3 programra, mely a következő részekből tevődik össze:

- Dynamips: Cisco IOS emulálását teszi lehetővé
- Dynagen: Parancssoros kezelőfelületet biztosít az eszközök konfigurálásához
- Qemu: PC- k szimulálásához szükséges összetevő, melyhez szintén külön OS szükséges

A Dynagen helyett a Putty terminálszolgáltatási programot integráltuk a környezetbe. A Qemu újabb erőforrás igénye miatt a végeszközök emulálását eltérő módon oldottuk meg.

A GNS3 teljesen ingyenes nyílt forráskódú alkalmazás, mely a http://gns3.net internetes oldalról tölthető le többféle platformra. Munkánk során a jelenleg elérhető legfrissebb, 0.7-es verziót használtuk.

Felülete átlátható, kezelése nem túl bonyolult, de futásához nagy erőforrásokra van szükség, mely a forgalomirányítók számának növekedésével drasztikusan növekszik. Ezért korlátozott számú útválasztóval dolgoztunk. A komplett konfiguráció megvalósítása után, a tesztelés során csak az adott funkciók szempontjából lényeges forgalomirányítókat futtattuk. Így külön kezeltük a VPNA és VPNB ügyfél kapcsolatait.

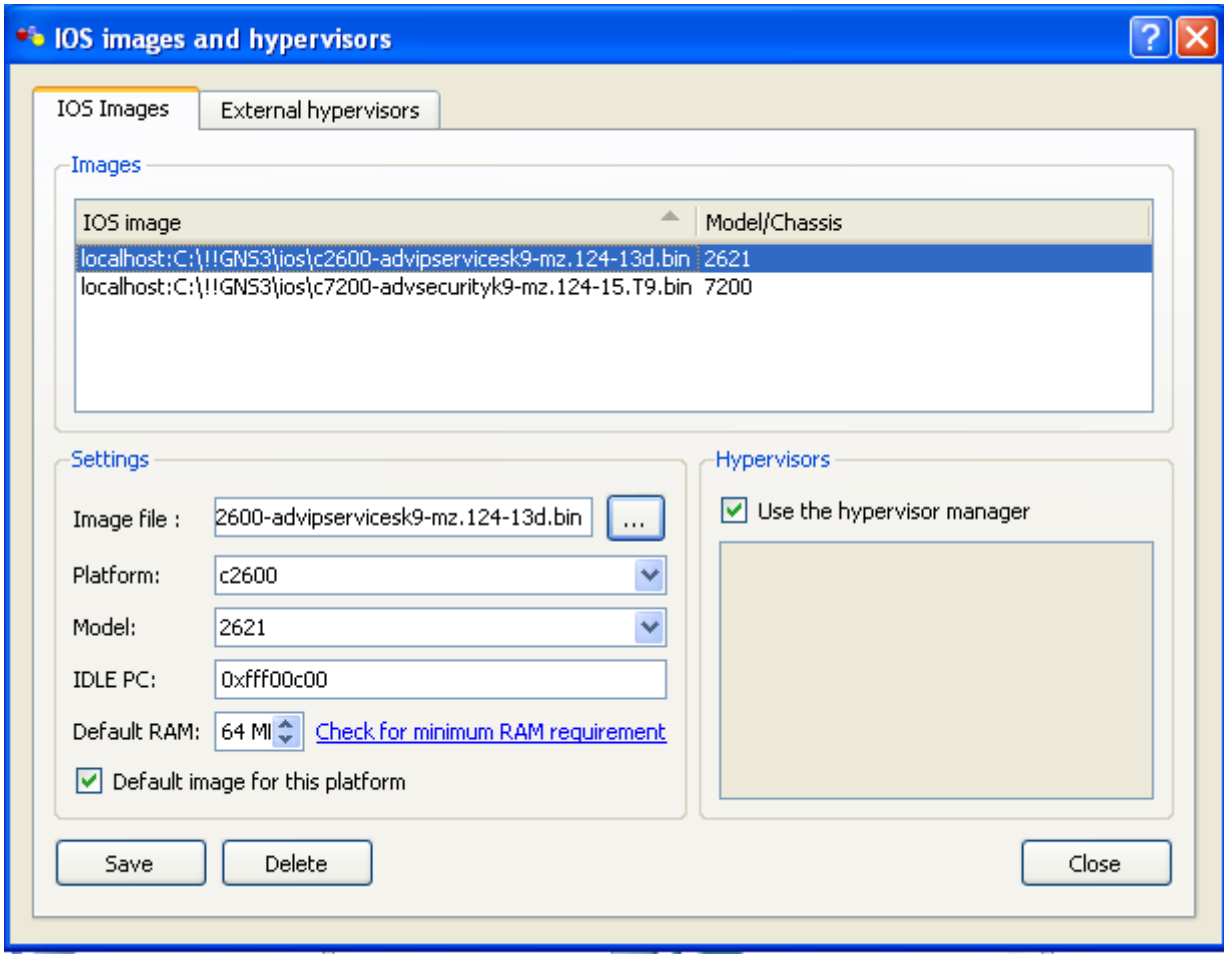


35. ábra

A 35. ábrán a GNS kezelőfelülete látható, amely 5 részre bontható:

- Menüsor: program kezelési és eszköz beállítási parancsok gyűjteménye
- Eszköztár: tervezéshez és futtatáshoz szükséges funkciók, például: kábelezés konfigurációk importálása és exportálása
- Csomópont típusok: hálózati eszközök kiválasztását lehetővé tevő felület
- Tervező felület: a topológia szerkesztését és elrendezését végezhetjük el
- Topológia adatbázis: a topológiában szereplő eszközök felsorolása

A 36. ábrán az IOS-ek kezelése látható. Egyedileg konfigurálható a kívánt router típus és IOS verzió az eszközökre vonatkozó beállításokkal együtt (alapértelmezett RAM).



36. ábra

Szintén ezen az ábrán látható az IDLE PC érték, melynek szerepe kulcsfontosságú. Egy adott IOS futtatása esetén a számítógép paramétereinek függvényében törekednünk kell egy alkalmas érték megválasztására az optimális teljesítmény elérése érdekében. A modell működésének bemutatása a GNS3 ezen képessége nélkül lehetetlen lett volna.

A hostok szimulálását a Virtual PC Simulator segítségével végeztük. A program kilenc számítógép egyidejű futtatását teszi lehetővé egyedi port számok generálásával, melyet a GNS3-beli felhő működésénél használtunk fel. Így képesek voltunk végponttól végpontig terjedő ügyfélkapcsolatok tesztelésére a *ping* parancs segítségével.

7.3 Hálózat működésének szemléltetése

A tervezési fázis után megvalósítottuk a topológiát a GNS3 környezetben. Ez magában foglalta:

- A routerek típusok és a megfelelő IOS verziók kiválasztás
- A routerek topológiának megfelelő elhelyezése
- A szükséges fizikai kapcsolatok kialakítása az eszközök között (kábelezés)
- Eszközök elindítása
- A kisebb erőforrás igény érdekében az IDLE PC érték kiválasztása

Ezek után nekiláttunk az eszközök konfigurálásának a fentebb ismertetett követelményeknek megfelelően. Az első fázisban az IP-címek kiosztása történt meg.

Az OSPF és BGP protokollok használatához minden forgalomirányító esetén egy-egy routerID szükséges, amely menedzsment célokat is ellát. Ehhez minden eszközön egy úgynevezett logikai Loopback interfészt konfiguráltunk, melynek legfontosabb jellemzője, hogy mindig „up” státuszban van. A Loopback IP-címeket a következő táblázat foglalja össze:

Router neve	**Loopback interfész IP-címe**
P1	10.100.0.1/32
P2	10.100.0.2/32
P3	10.100.0.3/32
PEFRA	10.200.0.1/32
PENY1	10.200.0.2/32
PENY2	10.200.0.3/32
PELON	10.200.0.4/32
PEBUD	10.200.0.5/32
CEDEBVRFAB1	10.250.0.1/32
CEMUNVRFB1	10.250.0.2/32
CEMANVRFA1	10.250.0.3/32
CENYVRFA1	10.250.0.4/32
CENYVRFA2	10.250.0.5/32

Ezek után az eszközök fizikai interfészeinek az IP-címeit állítottuk be az alábbiak szerint:

Provider routerek közötti kapcsolatok:

Routerek nevei	**Subnet**	**Interface 1**	**Interface 2**
P1 – P2	10.0.0.0/30	10.0.0.1	10.0.0.2
P2 – P3	10.0.0.4/30	10.0.0.5	10.0.0.6
P1 – P3	10.0.0.8/30	10.0.0.9	10.0.0.10

Provider és Provider Edge routerek közötti kapcsolatok:

Routerek nevei	**Subnet**	**Interface 1**	**Interface 2**
P1 – PEFRA	10.5.0.0/30	10.5.0.1	10.5.0.2
P1 – PENY1	10.6.0.0/30	10.6.0.1	10.6.0.2
P2 – PENY2	10.6.0.4/30	10.6.0.5	10.6.0.6
P2 – PELON	10.7.0.0/30	10.7.0.1	10.7.0.2
P3 – PEBUD	10.8.0.0/30	10.8.0.1	10.8.0.2

Provider Edge és Customer Edge routerek közötti kapcsolatok:

Routerek nevei	**Subnet**	**Interface 1**	**Interface 2**
PEFRA – CEMUNVRFB-1	10.5.0.4/30	10.5.0.5	10.5.0.6
PENY1 – CENYVRFA-1	10.6.0.8/30	10.6.0.9	10.6.0.10
PENY2 – CENYVRFA-2	10.6.0.12/30	10.6.0.13	10.6.0.14
PELON – CEMANVRFA-1	10.7.0.4/30	10.7.0.5	10.7.0.6
PEBUD – CEDEBVRFAB-1	10.8.0.4/30	10.8.0.5	10.8.0.6

A helyi hálózatok a következő C osztályú IP-címeket kapták a VPNA ügyfél esetén:

Routernév	**Interfész IP-címe**	
CE_NY_VRFA-1	192.168.1.2	HSRP:
CE_NY_VRFA-2	192.168.1.3	192.168.1.1
CE_MAN_VRFA-1	192.168.2.1	
CE_DEB_VRFAB-1	192.168.4.1	

VPNB ügyfél esetén:

Routernév	**Interfész IP-címe**
CE_MUN_VRFB-1	192.168.3.1
CE_DEB_VRFAB-1	192.168.5.1

A VPNA ügyfél New Yorkban található telephelyénél, mint már írtuk két router található, melyek között HSRP-t alkalmazunk. Tehát mind két routernek van egy fizikai IP-címe és egy közös virtuális IP-címük, esetünkben a 192.168.1.1. A debreceni telephely pedig mindkét táblázatban szerepel, egy-egy IP-címmel.

Itt említenénk meg, hogy a bemutatott show parancsok egy része némileg hiányos kimeneteket mutat. Ennek oka, hogy a teljes konfiguráció implementálása után erőforrás problémákba ütköztünk így kénytelen voltunk a VPNA és VPNB -hez kötődő hálózati eszközöket külön működtetni. Így a szolgáltatás minőségének működése esetén a gerinchálózati forgalomirányítókon kívül a PEFRA, CE_MUN_VRFB-1 és a PEBUD, CE-DEB-VRFAB-1 útválasztók üzemeltek. Minden más eset bemutatásánál a VPNA -hoz kötődő eszközök működtek, tehát a PEFRA és CE-MUN-VRFB-1 útválasztókon kívül az összes eszköz.

Az eszközök közötti pont-pont kapcsolatok beállítása önmagában még nem elegendő ahhoz, hogy azok az elvárásoknak megfelelően kommunikáljanak. A távoli routerek közötti útvonalak létrehozásához forgalomirányító protokoll beállítása szükséges. A Provider (P) illetve a Provider és Provider Edge (PE) routerek között az OSPF (Open Shortest Path First – legrövidebb út protokoll) kapcsolatállapot alapú forgalomirányító protokollt használtuk. Az OSPF adatbázist tart fent a hálózati topológiáról, és ez alapján határozza meg a legrövidebb utat két csomópont között.

Az OSPF-et a következőképpen konfiguráltuk a P1 router esetében:

```
router ospf 1
 router-id 10.100.0.1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.8 0.0.0.3 area 0
 network 10.5.0.0 0.0.0.3 area 0
 network 10.6.0.0 0.0.0.3 area 0
 network 10.100.0.1 0.0.0.0 area 0
```

A P2 és P3 routereken az OSPF konfigurálása szintén a megfelelő szomszédos alhálózatok hirdetésével történik.

A PE routerek esetén csak egy alhálózatot hirdetünk, mégpedig a P és a PE közötti kapcsolatot, ami a PEFRA router esetén a következőképpen alakul:

```
router ospf 1
 router-id 10.200.0.1
 log-adjacency-changes
 network 10.5.0.0 0.0.0.3 area 0
 network 10.200.0.1 0.0.0.0 area 0
```

Minden P és PE router a 0-s területazonosítót használja, mivel esetünkben nincs szükség további területazonosítók használatára. Továbbá minden router hirdeti a saját Loopback IP-címét is.
Miután az OSPF felépítette a topológia adatbázist, az eszközök közötti kommunikáció már biztosított. Az OSPF által létrehozott szomszédsági viszonyokat a következő parancs segítségével ellenőrizhetjük:

```
P2#sh ip ospf neighbor
Neighbor ID   Pri   State   Dead Time   Address     Interface
10.200.0.4    0     FULL/-  00:00:33    10.7.0.2    Serial0/3
10.200.0.3    0     FULL/-  00:00:34    10.6.0.6    Serial0/2
10.100.0.3    0     FULL/-  00:00:32    10.0.0.6    Serial0/1
10.100.0.1    0     FULL/-  00:00:39    10.0.0.1    Serial0/0
```

Például a P2 router forgalomirányító táblája a következőképpen néz ki:

```
P2#sh ip route
…
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
O       10.8.0.0/30 [110/128] via 10.0.0.6, 00:49:25, Serial0/1
O       10.0.0.8/30 [110/128] via 10.0.0.6, 00:49:25, Serial0/1
                    [110/128] via 10.0.0.1, 00:49:25, Serial0/0
C       10.6.0.4/30 is directly connected, Serial0/2
C       10.0.0.0/30 is directly connected, Serial0/0
O       10.6.0.0/30 [110/128] via 10.0.0.1, 00:49:25, Serial0/0
C       10.7.0.0/30 is directly connected, Serial0/3
```

```
C       10.0.0.4/30 is directly connected, Serial0/1
C       10.100.0.2/32 is directly connected, Loopback0
O       10.100.0.3/32 [110/65] via 10.0.0.6, 00:49:25, Serial0/1
O       10.100.0.1/32 [110/65] via 10.0.0.1, 00:49:25, Serial0/0
O       10.200.0.2/32 [110/129] via 10.0.0.1, 00:49:25, Serial0/0
O       10.200.0.3/32 [110/65] via 10.6.0.6, 00:49:29, Serial0/2
O       10.200.0.4/32 [110/65] via 10.7.0.2, 00:49:29, Serial0/3
O       10.200.0.5/32 [110/129] via 10.0.0.6, 00:49:29, Serial0/1
```

Az „O”-val jelölt forgalomirányító tábla bejegyzések az OSPF által megtanult hálózatokat jelöli, míg a „C”-vel jelölt sorok a közvetlenül kapcsolódó linkeket jelentik.

Az MPLS működésének szemléltetéséhez a gerinchálózati P forgalomirányítók címkekapcsolást alkalmaznak a hálózati csomagok gyors továbbítása érdekében. Mint fentebb említettük a forgalomirányítási információk terjesztését OSPF protokoll biztosítja, melyet az IS-IS hálózati protokoll mellett gerinchálózati környezetben használnak. Eddigi tanulmányaink során az OSPF protokollt alaposan megismertük, ezért választásunk erre a kapcsolatállapot alapú forgalomirányítási protokollra esett.

Az MPLS konfigurálásához legelőször engedélyezni kell a Cisco Express Forwarding protokollt

Konfigurációs módban az *„ip cef”* parancs használatával, amely gyorsabb csomagtovábbítást tesz lehetővé. Ezután az MPSL címkék továbbítását biztosító protokollt kell megadni a következő parancs kiadásával:

Router(config)#mpls label-distribution [ldp/tdp/both]

Választhatunk a Label Distribution Protocol és a Tag Distribution Protocol közül, de együttes alkalmazásuk is lehetséges. Mi az LDP mellett döntöttünk, mivel az gyártó független és fentebb ismertettük működését. Továbbá minden olyan interfészen, amely MPLS adatforgalmat fog biztosítani, ki kell adni a *„tag-switching ip”* parancsot. Ezen parancsok segítségével az MPLS alapkonfigurációját elvégeztük.

Az MPLS alapú gerinchálózati útválasztók LDP címke beágyazás alapján továbbítják a csomagokat, illetve ezen protokoll mentén építik fel MPLS szomszédsági viszonyaikat.

Ennek ellenőrzését láthatjuk a következő show kimenetben a P1 gerinchálózati router –re vonatkozóan:

```
P1#sh mpls ldp neighbor
    Peer LDP Ident: 10.100.0.2:0; Local LDP Ident 10.100.0.1:0
        TCP connection: 10.100.0.2.26082 - 10.100.0.1.646
        State: Oper; Msgs sent/rcvd: 304/311; Downstream
        Up time: 00:27:30
        LDP discovery sources:
          Serial0/0, Src IP addr: 10.0.0.2
        Addresses bound to peer LDP Ident:
          10.0.0.2       10.100.0.2      10.0.0.5        10.6.0.5
          10.7.0.1
    Peer LDP Ident: 10.100.0.3:0; Local LDP Ident 10.100.0.1:0
        TCP connection: 10.100.0.3.50635 - 10.100.0.1.646
        State: Oper; Msgs sent/rcvd: 117/69; Downstream
        Up time: 00:16:23
        LDP discovery sources:
          Serial0/1, Src IP addr: 10.0.0.10
        Addresses bound to peer LDP Ident:
          10.0.0.10       10.100.0.3     10.0.0.6        10.8.0.1
    Peer LDP Ident: 10.200.0.2:0; Local LDP Ident 10.100.0.1:0
        TCP connection: 10.200.0.2.64271 - 10.100.0.1.646
        State: Oper; Msgs sent/rcvd: 29/29; Downstream
        Up time: 00:08:03
        LDP discovery sources:
          Serial0/2, Src IP addr: 10.6.0.2
        Addresses bound to peer LDP Ident:
          10.200.0.2      10.6.0.2
```

A címkék hozzárendelődésének ellenőrzésére pedig a következő show parancsot használhatjuk a forgalomirányítókon, ahol jól látható a címke megfeleltetési táblázat, amely tartalmazza a bemenő és kimenő címkéket, a kimenő interfészeket és a kapcsolatok típusát:

```
P1#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     10.200.0.2/32    1801       Se0/2      point2point
18     Pop tag     10.6.0.4/30      0          Se0/0      point2point
19     18          10.200.0.4/32    460        Se0/0      point2point
20     21          10.200.0.3/32    469        Se0/0      point2point
21     Pop tag     10.0.0.4/30      0          Se0/1      point2point
       Pop tag     10.0.0.4/30      0          Se0/0      point2point
22     Pop tag     10.7.0.0/30      0          Se0/0      point2point
23     Pop tag     10.100.0.2/32    0          Se0/0      point2point
25     Pop tag     10.8.0.0/30      0          Se0/1      point2point
26     Pop tag     10.100.0.3/32    0          Se0/1      point2point
27     21          10.200.0.5/32    1436       Se0/1      point2point
```

Ahhoz, hogy a hálózat alapvető forgalomirányítása és működése teljes legyen, még ki kell térnünk a PE forgalomirányítók, illetve a PE-CE routerek között konfigurált BGP protokoll működésére. Itt beszélhetünk közvetlen szomszédságban és távoli szomszédságban lévő BGP partnerekről. Fontos megjegyeznünk, hogy a távoli szomszédságban álló routerek globális routing alapján kommunikálnak. Tehát forgalomirányítási információjukat VRF –ektől függetlenül terjesztik és kezelik. A közvetlen szomszédok az adott ügyfélhez tartozóan VRF-en belül építik fel szomszédsági viszonyukat. Lássunk ezekre példát:
A magyarországi eszközök esetét fogjuk bemutatni, mert itt a már említett stratégiai okok miatt mindkét VRF jelen van mind a PE mind a CE routeren. Vegyük a budapesti PE routert. Ez az eszköz távoli szomszédságban áll az összes többi PE routerrel, globális módon. A frankfurti PE router az említett korlátozott erőforrásaink miatt kikapcsolt állapotban volt, ez látható a következő parancs kimenetében is.

```
PE_BUD#sh ip bgp summary
BGP router identifier 10.200.0.5, local AS number 450
BGP table version is 7, main routing table version 7
3 network entries using 351 bytes of memory
3 path entries using 144 bytes of memory
…
```

```
Neighbor        V    AS  MsgRcvd  MsgSent  TblVer    InQ  OutQ  Up/Down
State/PfxRcd
10.200.0.1   4  450       0       0       0     0    0   never       Active
10.200.0.2   4  450      12      15       7     0    0   00:17:59        1
10.200.0.3   4  450      11      13       7     0    0   00:16:00        1
10.200.0.4   4  450      13      16       7     0    0   00:18:00        1
```

Mind a VPNA-ra mind a VPNB-re vonatkozóan a budapesti PE router közvetlen szomszédságban áll a debreceni CE forgalomirányítóval.

```
PE_BUD#sh ip bgp vpnv4 vrf vpna summary
BGP router identifier 10.200.0.5, local AS number 450
BGP table version is 38, main routing table version 38
…
Neighbor  V  AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.8.0.6  4  65000   10      13     38   0    0 00:15:39          2

PE_BUD#sh ip bgp vpnv4 vrf vpnb summary
BGP router identifier 10.200.0.5, local AS number 450
BGP table version is 38, main routing table version 38
…
Neighbor  V  AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.8.0.10 4  65000   10      10     38   0    0 00:15:55          2
```

A PEBUD és CEDEBVRFAB-1 között a két VPN-hez szükséges szomszédsági viszonyok kiépítéséhez alinterfészeket konfiguráltunk. A többi eszköz szomszédsági viszonyai ezzel analóg módon értelmezhető.

A következő részben az USA-ban létrehozott backup megoldással ellátott telephelyet tárgyaljuk. Ezen belül is a HSRP protokoll működését és annak megmagyarázását, hogy a New York-i LAN elérésére miért is a PENY1 – CENYVRFA-1 útvonal lesz a preferált. Kezdjük az utóbbival. Mint már tudjuk a BGP útvonalvektor alapú forgalomirányítási protokoll, de mértékeket ugyanúgy kezel az ugyanazon célhoz vezető útvonalak jóságának eldöntésére. Ez AS -k között a lokális preferencia, míg AS belül a metrika, melyek értelmezése ellentétes. Míg lokális preferencia esetén a nagyobb, metrika esetén a kisebb érték a preferált. A BGP ezen tulajdonságát használtuk ki az elsődleges útvonal megjelölésére

a következőképpen. A CE routereken úgynevezett ROUTE-MAP –eket használtunk, melyek feladata sokrétű lehet. Jelen esetben a metrika állításáért felelős kimenő irányban. Ez gyakorlatilag megvalósítja, hogy az adott CE router az általa ismert útvonalakat a ROUTE-MAP -ben megadott metrikával hirdesse. Az adott ROUTE-MAP-et az adott CE router-en a BGP-ben megadott közvetlen szomszédra alkalmazzuk jelen esetben a PENY1-re és PENY2-re.

```
neighbor 10.6.0.9 route-map PRIMARY-PATH out
route-map PRIMARY-PATH permit 10
 set metric 50
…
neighbor 10.6.0.13 route-map BACKUP-PATH out
…
route-map BACKUP-PATH permit 10
 set metric 100
…
```

Tehát a CENYVRFA-1 forgalomirányító 50-es metrikával hirdeti ugyanazt a LAN-t amit a tartalék CE 100-as metrikával. Így az MPLS hálózatban a CENYVRFA-1 felé vezető útvonal lesz a preferált. Ezt könnyen tesztelhetjük a következő trace parancs segítségével, amellyel végponttól végpontig tesztelhetjük a csomag útvonalát:

```
VPCS 4 >trace 192.168.1.4
traceroute to 192.168.1.4, 64 hops max
 1   192.168.4.1   70.000 ms  105.000 ms  47.000 ms
 2   10.8.0.5   143.000 ms  145.000 ms  157.000 ms
 3   10.8.0.1   527.000 ms  514.000 ms  502.000 ms
 4   10.0.0.9   689.000 ms  359.000 ms  263.000 ms
 5   10.6.0.9   350.000 ms  299.000 ms  387.000 ms
 6   10.6.0.10   264.000 ms  400.000 ms  442.000 ms
 7   192.168.1.4   265.000 ms  409.000 ms  596.000 ms
```

Ellenőrzés véget megcseréltük a metrikákat, a tartalék routert 50-es metrikával az elsődleges routert pedig 100-as metrikával láttuk el. Ekkor a trace parancs kimenete az elvárásoknak megfelelően a következőképpen alakult:

```
VPCS 4 >trace 192.168.1.4
traceroute to 192.168.1.4, 64 hops max
 1   192.168.4.1   88.000 ms  67.000 ms  55.000 ms
 2   10.8.0.5   193.000 ms  196.000 ms  98.000 ms
 3   10.8.0.1   387.000 ms  733.000 ms  426.000 ms
 4   10.0.0.5   458.000 ms  360.000 ms  375.000 ms
 5   10.6.0.13   342.000 ms  261.000 ms  252.000 ms
 6   10.6.0.14   305.000 ms  395.000 ms  405.000 ms
 7   192.168.1.4   576.000 ms  513.000 ms  549.000 ms
```

A HSRP protokoll konfigurációját követően az azonos HSRP csoportban szereplő forgalomirányítók között megtörténik az aktív, illetve készenléti forgalomirányítók kiválasztása.
Az elsődleges router HSRP konfigurációja:

```
ce-ny-vrfa-1# show run interface FastEthernet0/0
interface FastEthernet0/0
 ip address 192.168.1.2 255.255.255.0
 standby 1 ip 192.168.1.1
 standby 1 preempt delay minimum 60
 standby 1 track FastEthernet1/0 20
```

A másodlagos router konfigurációja pedig:

```
ce-ny-vrfa-2# show run interface FastEthernet0/0
interface FastEthernet0/0
 ip address 192.168.1.3 255.255.255.0
 standby 1 ip 192.168.1.1
 standby 1 priority 90
 standby 1 preempt
```

A HSRP állapotokat a „*show standby brief”* parancs kiadásával ellenőrízhetjük:

```
ce_ny_vrfa-1#show standby brief
Interface   Grp Prio P State    Active     Standby         Virtual IP
Fa0/0       1   100  P Active   local      192.168.1.3    192.168.1.1

ce_ny_vrfa-2#show standby brief
Interface   Grp Prio P  State    Active       Standby     Virtual IP
Fa0/0       1   90   P  Standby  192.168.1.2  local       192.168.1.1
```

A LAN-ban elhelyezkedő állomásnak a virtuális átjáró címét adjuk meg, tehát a 192.168.1.1-es IP címet. Hibamentes esetben a forgalom az elsődleges CE routeren halad keresztül. A hibát a CE router LAN felé vezető interfészének lekapcsolásával szimuláljuk, melyet követően az eddig készenléti backup CE válik aktív alapértelmezett átjáróvá.

```
ce_ny_vrfa-1(config)#int fa0/0
ce_ny_vrfa-1(config-if)#shutdown        < -----interfész deaktiválása

*Mar   1 01:36:06.551: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1
state Active -> Init
*Mar   1 01:36:08.551: %LINK-5-CHANGED: Interface FastEthernet0/0,
changed state to administratively down
*Mar    1  01:36:09.551:  %LINEPROTO-5-UPDOWN:  Line  protocol  on
Interface FastEthernet0/0, changed state to down

ce_ny_vrfa-2#
*Mar   1 01:36:14.327: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1
state Standby -> Active
```

Ezt követően a forgalom a backup CE-n halad keresztül.

A gyakorlati működés szempontjából utolsóként bemutatandó példa a szolgáltatási szintek definiálása a VPNB ügyfél számára a müncheni és debreceni telephely között. A QoS (Quality of Service) konfiguráció működésének bemutatására a ping parancsot használjuk, amellyel 5 csomagot küldünk a müncheni LAN-ból a debreceni LAN-ba és figyeljük, miként utazik végig ez a csomag a hálózaton. Célunk, hogy a neki kialakított business osztályon belül

utazzon a csomag, mely a gerinchálózaton kritikus adatként kezelendő. A bemutatást csak az egyik irányban mutatjuk be, a fordított irány teljesen analóg módon történik. A QoS-t CLASS-MAP-ek és POLICY-MAP-ek segítségével valósítottuk meg, melyet a konfigurációkban tekinthetünk meg részletesen. Lényege, hogy a CLASS-MAP-ekben a különböző feltételeket vizsgáljuk és a megfelelő értékeket állítjuk be. A POLICY-MAP-ekben pedig ezen CLASS-MAP-okat felhasználva paramétereket adunk meg például a sávszélességre vonatkozóan. Korlátozhatjuk, hogy az egyes osztályok a teljes sávszélesség mekkora részét használhatják.

Ezeket a POLICY-MAP-eket interfészekre alkalmazzuk a megfelelő irányban, mellyel elérjük, hogy a csomag egy neki létrehozott elkülönített szolgáltatási szinten utazzon végig a hálózaton. Ennek lépései a München-Debrecen irányt nézve:

A müncheni CE router fa0/0 interfészén alkalmazunk egy bemenő policyt, mely hozzáférési listákat (ACL) felhasználva elkapja a LAN-ból érkező csomagokat. Esetünkben ez annyit jelentett, hogy a business osztály számára létrehoztunk egy ACL –t mely engedélyezi az ICMP protokollt tetszőleges forrás és célcím tekintetében. Ezért itt egyezés fog történni, bármilyen ping végrehajtása esetén. A CE router fa0/1 interfészén alkalmazunk egy kimenő policyt, mely ellátja a csomagokat a megfelelő DSCP értékekkel. A következő lépésben a frankfurti PE router fa0/0 interfészén alkalmazunk egy bemenő és egy kimenő policyt is. Bemenő esetben a DSCP értékeknek megfelelően beállítja az EXP bit értékét, hogy a csomagok a megfelelő QoS értékekkel ellátva tudjanak utazni az MPLS hálózatban. Kimenő irányban természetesen ennek fordítottja történik, tehát az EXP bitek eltávolítása és a megfelelő DSCP érétkek beállítása.

A gerinchálózati forgalomirányítókon csak két osztályt hozunk létre. Egy kritikus adatnak és egy valós idejű adatnak megfelelő osztályt. A class-default alapértelmezett osztály természetesen létezik, ha az adott csomag nem sorolható be egy általunk létrehozott osztályba sem. A business osztály forgalma kritikus adatnak minősül. A megfelelő policyt a backbone routerek megfelelő kimenő interfészein alkalmazzuk. P1 esetén a s0/1 interfész, míg P3 esetén s0/2 interfész. A budapesti PE router a PEFRA PE által elvégzett műveletek fordítottját hajtja végre. Az s0/0 interfészén bemenő irányban alkalmazott policy alapján leválogatja az EXP biteket és ellátja a csomagot a megfelelő DSCP értékkel. Lássuk, hogyan is történt ez a gyakorlatban:

A teendő tehát nem más, mint a müncheni LAN-ból érkező ping-et eljuttatni Debrecenbe a business szolgáltatási szinten keresztül.

```
VPCS 5 >ping 192.168.4.2
192.168.4.2 icmp_seq=1 0.0237ms
192.168.4.2 icmp_seq=2 0.0443ms
192.168.4.2 icmp_seq=3 0.0342ms
192.168.4.2 icmp_seq=4 0.0235ms
192.168.4.2 icmp_seq=5 0.0543ms
```

A ping végrehajtása után vizsgáljuk meg mit is láthatunk a megbeszélt interfészeken alkalmazott policyk –en belül.

```
ce_mun_vrfb-1#sh policy-map interface Fa0/0
…
  Service-policy input: cos-map-vpnb-fa-0.0
   Class-map: cos-map-vpnb-fa-0.0-vo (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: access-group name cos-map-vpnb-fa-0.0-vo
        0 packets, 0 bytes
        30 second rate 0 bps

    Class-map: cos-map-vpnb-fa-0.0-st (match-any)
…
    Class-map: cos-map-vpnb-fa-0.0-bu (match-any)
      33 packets, 4224 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: access-group name cos-map-vpnb-fa-0.0-bu
        33 packets, 4224 bytes
        30 second rate 0 bps
      QoS Set
        dscp af21
          Packets marked 33
Class-map: class-default (match-any)
…
```

A fenti kimenet alapján látszik, hogy a kívánt csomagok valóban illeszkedtek az előre definiált hozzáférési listára, majd ezután af21 DSCP értékkel lettek megjelölve és a business osztályba kerültek.

```
ce_mun_vrfb-1#sh policy-map interface Fa0/1
 …
  Service-policy output: cos-parent-vpnb-fa-0.1

    Class-map: class-default (match-any)
      Service-policy : cos-child-vpnb-fa-0.1
…
        Class-map: cos-map-dscp-bu-1 (match-any)
          28 packets, 2768 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip dscp af21 (18) af22 (20) af23 (22)
            28 packets, 2768 bytes
            5 minute rate 0 bps
          Queueing
            Output Queue: Conversation 137
            Bandwidth 3530 (kbps) Max Threshold 590 (packets)
            (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

        Class-map: cos-map-dscp-nm-1 (match-any)
…
        Class-map: class-default (match-any)
          …
```

A ce_mun_vrfb-1 router kimenő interfészén az ICMP csomagok már meg vannak jelölve az af21 DSCP értékkel, így azoknak csak a megfelelő osztályokba sorolásáról kell gondoskodnunk.

```
PE_FRA#sh policy-map interface fa0/0

  Service-policy input: cos-po-in-set-exp-dscp-1
```

```
    Class-map: cos-map-dscp-ec-1 (match-all)
…
    Class-map: cos-map-dscp-st-1 (match-all)
…
    Class-map: cos-map-dscp-bu-1 (match-all)
      28 packets, 2768 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af21  af22  af23
      QoS Set
        mpls experimental imposition 7
          Packets marked 28

    Class-map: cos-map-dscp-vo-1 (match-all)
…
    Class-map: cos-map-dscp-vs-1 (match-all)
…
    Class-map: cos-map-dscp-nm-1 (match-all)
      …

    Class-map: class-default (match-any)
…
```

A PE-FRA router fa0/0 interfészén megvizsgáljuk a csomagok milyen DSCP értékekkel vannak ellátva, és ennek megfelelően állítjuk be az MPLS EXP biteket, melyek segítségével a gerinchálózaton is kezelhetővé válnak a különböző csomagprioritások. Az ICMP csomagnak itt 7-es értékű EXP biteket állítunk be.

```
P1#sh policy-map interface Serial0/1
…
  Service-policy output: cos-CORE

    Class-map: cos-map-CRITICAL (match-all)
      28 packets, 2712 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
```

```
      Match: mpls experimental topmost 1  2  3  4  7
      Queueing
        Output Queue: Conversation 265
        Bandwidth 40 (%)
        Bandwidth 617 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: cos-map-REALTIME (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: mpls experimental topmost 5
      …

    Class-map: class-default (match-any)
      2019 packets, 144919 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

A P routereken mindössze az EXP bitek vizsgálatát végezzük el. Esetünkben P routerek között a hálózati forgalom három kategóriába tartozhat, nevezetesen: valós idejű, kritikus illetve minden egyéb. Az ICMP csomagok a **cos-map-CRITICAL** osztályra illeszkednek, mivel 7–es EXP bittel voltak megjelölve.

```
P3#sh policy-map interface Serial0/2
…
  Service-policy output: cos-CORE

    Class-map: cos-map-CRITICAL (match-all)
      28 packets, 2600 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: mpls experimental topmost 1  2  3  4  7
      …
    Class-map: cos-map-REALTIME (match-all)
…
```

```
    Class-map: class-default (match-any)
…

PE_BUD#sh policy-map interface s0/0

  Service-policy input: cos-po-in-set-dscp

    Class-map: cos-map-match-exp-5-vo (match-all)
…
    Class-map: cos-map-match-exp-0-ec (match-all)
      …
    Class-map: cos-map-match-exp-2-st (match-all)
      …

    Class-map: cos-map-match-exp-6-nm (match-all)
…
    Class-map: cos-map-match-exp-7-bu (match-all)
      28 packets, 2600 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: mpls experimental topmost 7
…
    Class-map: class-default (match-any)
…
```

A PEBUD forgalomirányító a P3 routertől megkapott EXP bittel megjelölt csomagjait a Serial0/0 interfészén fogadja, majd az f0/0 kimenő interfészén az EXP biteknek megfelelően gondoskodik a megfelelő DSCP értékek beállításáról, amely esetünkben az *af21*.

Jól látható, ahogy a ping adatforgalma végighalad az egyes hálózati eszközökön alkalmazott policyken keresztül. A kimenetekben szereplő egyéb adatforgalom az előző tesztelésekből maradt vissza.

Összefoglalás

Dolgozatunkban bemutattuk az MPLS alapú virtuális magánhálózatok által nyújtott lehetőségeket, és az ezen technológiával kapcsolatos szabványokat illetve módszereket.
Részleteztük a Határ átjáró protokollt (a BGP-t), a Multiprotocol Label Switching architektúra felépítését és működését, illetve a VPN-ek kialakításának előnyeit.
Továbbá ismertettük a napjaink hálózataival szemben elvárt követelményeket, mint a szolgáltatásminőségi szintek meghatározását, és a legelterjedtebb backup technológiákat.
Egy szimulációs környezetben megterveztünk és felépítettünk egy olyan szolgáltatói hálózatot, amely képes kielégíteni két vállalat üzleti igényeit. Hálózatunkat a jelenleg elfogadott standard-eknek megfelelően alakítottuk ki, így a későbbi bővítés lehetőségét is biztosítottuk. További telephelyek, illetve új ügyfelek hozzáadása sem követelné meg a hálózat teljes újratervezését, azok könnyen csatlakoztathatóak lennének az MPLS hálózatunk határán lévő PE routerek valamelyikéhez.
A szimulációs környezet jelentős erőforrásigénye sajnos nagymértékben korlátozta a lehetőségeinket, leginkább az egyidejűleg futtatható routerek számát illetően. Ezért igyekeztünk egy olyan topológiát kialakítani, amely megfelelő számú csomópontot tartalmaz ahhoz, hogy magvalósítsuk elképzeléseinket, de a rendelkezésünkre álló erőforrásokat ne lépje túl. Természetesen egy összetettebb topológia használatával további technológiákat is bemutathattunk volna, mint például a Route Reflectorok használata, a Traffic Engineering vagy a loadsharing, amely lehetővé teszi a terhelés elosztását, illetve egy olyan ügyfélhálózat ismertetése, amely hub-and-spoke topológiát valósít meg.
Ebből is látszik, hogy az MPLS VPN hálózatok rengeteg további lehetőséget kínálnak, melyek részletes tárgyalása ezen dolgozat keretein belül nem tértünk ki.
Célunk az volt, hogy témáinkkal átfogó betekintést nyújtsunk az olvasó számára a szolgáltatói hálózatok alapvető működésébe. Reméljük ezt sikerült elérni és alaposabb tanulmányozás után akár komplexebb ismeretek elsajátítására is lehetőséget ad munkánk.

Köszönetnyilvánítás

Ezúton szeretnénk megköszönni témavezetőnknek, Dr. Almási Bélának a dolgozat elkészítése során nyújtott szakmai segítségnyújtást, valamint építő jellegű kritikáit.

Köszönetünket fejezzük ki továbbá a T-Systems-nél dolgozó kollegáknak, akik tanácsaikkal, tapasztalatiak megosztásával járultak hozzá munkánk sikeréhez. Szeretnénk kiemelni közülük a TSS részleg kiváló szakemberét Szilágyi Ferencet, továbbá az IP Configuration team külföldi szakembereit, Andreas Drehert-t, Lars Poller-t, Michael Frey-t és Simon C. Hirsch-t, akik átfogó segítséget nyújtottak a témák megvilágításában.

Köszönettel tartozunk közvetlen főnökeinknek Thomas Tilp–nek és Michael Dazert-nek, hogy lehetőséget biztosítottak és biztosítanak arra, hogy munkánk végzése során újabb és újabb ismeretekkel gyarapodjunk.

Irodalomjegyzék

Könyvek:

[1] **Alwayn, Vivek:** Advanced MPLS design and implementation. Indianapolis: Cisco Press, 2001.

[2] **Bollapragada, Vijay; Khalid, Mohamed; Wainner, Scott**: IPsec VPN Design, Indianapolis: Cisco Press, 2005.

[3] **Carmouche, James Henry**: IPsec Virtual Private Network Fundamentals, Indianapolis: Cisco Press, 2006.

[4] **De Ghein, Luc:** MPLS fundamentals. Indianapolis: Cisco Press, 2007.

[5] **Lobo, Lancy; Lakshman, Umesh:** MPLS configuration on Cisco IOS software. Indianapolis: Cisco Press, 2006.

[6] **Pepelnjak, Ivan; Guichard, Jim:** MPLS and VPN Architectures. Indianapolis: Cisco Press, 2002.

[7] **Szigeti, Tim; Hattingh, Christina:** End-to-End QoS Network Design. Indianapolis: Cisco Press, 2005

[8] **Tanenbaum, Andrew S.:** Számítógép-hálózatok. Budapest: Panem, 2004.

[9] **Thomas, Stephen A.:** IP kapcsolás és útválasztás: RIP, OSPF, BGP, MPLS, CR-LDP, RSVP-TE. Budapest: Kiskapu, 2002.

[10] **Tomsu, Peter; Wieser, Gerhard:** MPLS-based VPNs : designing advanced virtual networks. Upper Saddle River: Prentice Hall, 2002.

RFC dokumentumok (http://www.rfc-editor.org/):

[11] **Andersson, L., Doolan, P.; Feldman, N.; etc.:** LDP Specification. 2001. RFC3036.

[12] **Awduche, D.; Berger, L.; Gan, D.; etc.:** RSVP-TE: Extensions to RSVP for LSP Tunnels. 2001. RFC3209.

[13] **Bates, T.; Chandra, R.; Katz, D.; Rekhter, Y.:** Multiprotocol Extensions for BGP-4. 1998. RFC2283.

[14] **Blake, S.; Black, D.; Carlson, M.; etc.:** An Architecture for Differentiated Services. 1998. RFC2475

[15] **Braden, R.; Clark, D.; Shenker, S**.: Integrated Services in the Internet Architecture: an Overview. 1994. RFC1633

[16] **Braden, R.; Zhang, L; Berson, S.; etc.:** Resource ReSerVation Protocol (RSVP). 1997. RFC2205

[17] **Harkins, D. and Carrel, D:** The Internet Key Exchange (IKE). 1998. RFC2409

[18] **Heinanen, J:** Multiprotocol Encapsulation over ATM Adaptation Layer 5. 1993. RFC1483.

[19] **Jacobson, V.; Nichols, K.; Poduri, K.**: An Expedited Forwarding PHB. 1999. RFC259

[20] **Kent, S.; Atkinson, R.**: IP Authentication Header (AH). 1998. RFC2402

[21] **Kent, S.; Atkinson, R.**: IP Encapsulating Security Payload (ESP). 1998. RFC2406

[22] **Kent, S**.: IP Encapsulating Security Payload (ESP). 2005. RFC 4303.

[23] **Laubach, M.:** Classical IP and ARP over ATM. 1994. RFC1577

[24] **Nichols, K.; Blake, S.; Baker, F.; Black, D.:** Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. 1998. RFC2474

[25] **Rosen, E.**: Removing a Restriction on the use of MPLS Explicit NULL. 2005. RFC4182.

[26] **Rosen, E.; Rekhter, Y.:** BGP/MPLS IP Virtual Private Networks (VPNs). 2006. RFC4364.

[27] **Rosen, E.; Rekhter, Y.:** BGP/MPLS VPNs. 1999. RFC2547.

[28] **Rosen, E.; Tappan, D., Fedorkow, G.; etc.:** MPLS Label Stack Encoding. 2001. RFC3032.

[29] **Rosen, E.;Viswanathan, A.; Callon, R.:** Multiprotocol Label Switching Architecture. 2001. RFC3031.

Függelék

Konfigurációk:

```
P1#sh running-config
version 12.3
...
hostname P1
...
ip subnet-zero
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
class-map match-all cos-map-CRITICAL
 match mpls experimental topmost 1  2  3  4  7
class-map match-all cos-map-REALTIME
 match mpls experimental topmost 5
policy-map cos-CORE
 class cos-map-CRITICAL
  bandwidth percent 40
 class cos-map-REALTIME
  bandwidth percent 20
interface Loopback0
 ip address 10.100.0.1 255.255.255.255
...
interface Serial0/0
 ip address 10.0.0.1 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
...
interface Serial0/1
 ip address 10.0.0.9 255.255.255.252
 service-policy output cos-CORE
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
interface Serial0/2
 ip address 10.6.0.1 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
interface Serial0/3
 ip address 10.5.0.1 255.255.255.252
 service-policy output cos-CORE
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
...
router ospf 1
```

```
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.100.0.1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.8 0.0.0.3 area 0
 network 10.5.0.0 0.0.0.3 area 0
 network 10.6.0.0 0.0.0.3 area 0
 network 10.100.0.1 0.0.0.0 area 0
ip classless
ip http server
...
```

P2# sh running-config

```
version 12.3
...
hostname P2
...
ip subnet-zero
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
interface Loopback0
 ip address 10.100.0.2 255.255.255.255
...
interface Serial0/0
 ip address 10.0.0.2 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
...
interface Serial0/1
 ip address 10.0.0.5 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
interface Serial0/2
 ip address 10.6.0.5 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
interface Serial0/3
 ip address 10.7.0.1 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 56000
...
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.100.0.2
```

```
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.4 0.0.0.3 area 0
 network 10.6.0.4 0.0.0.3 area 0
 network 10.7.0.0 0.0.0.3 area 0
 network 10.100.0.2 0.0.0.0 area 0
ip classless
ip http server
...
```

P3# sh running-config

```
version 12.3
...
hostname P3
...
ip subnet-zero
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
class-map match-all cos-map-CRITICAL
 match mpls experimental topmost 1  2  3  4  7
class-map match-all cos-map-REALTIME
 match mpls experimental topmost 5
policy-map cos-CORE
 class cos-map-CRITICAL
  bandwidth percent 40
 class cos-map-REALTIME
  bandwidth percent 20
interface Loopback0
 ip address 10.100.0.3 255.255.255.255
...
interface Serial0/0
 ip address 10.0.0.10 255.255.255.252
 service-policy output cos-CORE
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
...
interface Serial0/1
 ip address 10.0.0.6 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
interface Serial0/2
 ip address 10.8.0.1 255.255.255.252
 service-policy output cos-CORE
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
...
router ospf 1
```

```
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.100.0.3
 log-adjacency-changes
 network 10.0.0.4 0.0.0.3 area 0
 network 10.0.0.8 0.0.0.3 area 0
 network 10.8.0.0 0.0.0.3 area 0
 network 10.100.0.3 0.0.0.0 area 0
ip classless
ip http server
...
```

PE_FRA# sh running-config

```
version 12.3
…
hostname PE_FRA
no aaa new-model
ip cef
ip vrf vpnb
 rd 450:111
 route-target export 450:111
 route-target import 450:111
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
class-map match-all cos-map-match-exp-2-st
 match mpls experimental topmost 2
class-map match-all cos-map-match-exp-6-nm
 match mpls experimental topmost 6
class-map match-all cos-map-match-exp-0-ec
 match mpls experimental topmost 0
class-map match-all cos-map-match-exp-7-bu
 match mpls experimental topmost 7
class-map match-all cos-map-match-exp-5-vo
 match mpls experimental topmost 5
class-map match-all cos-map-dscp-bu-1
 match ip dscp af21  af22  af23
class-map match-all cos-map-dscp-vo-1
 match ip dscp ef
class-map match-all cos-map-dscp-ec-1
 match ip dscp default
class-map match-all cos-map-dscp-st-1
 match ip dscp af41  af42  af43
class-map match-all cos-map-dscp-vs-1
 match ip dscp cs3  af31
class-map match-all cos-map-dscp-nm-1
 match ip dscp cs6
policy-map cos-po-in-set-dscp
 class cos-map-match-exp-5-vo
  set ip dscp ef
 class cos-map-match-exp-0-ec
  set ip dscp default
```

```
 class cos-map-match-exp-2-st
  set ip dscp af41
 class cos-map-match-exp-6-nm
  set ip dscp cs6
 class cos-map-match-exp-7-bu
  set ip dscp af21
policy-map cos-po-in-set-exp-dscp-1
 class cos-map-dscp-ec-1
  set mpls experimental imposition 0
 class cos-map-dscp-st-1
  set mpls experimental imposition 2
 class cos-map-dscp-bu-1
  set mpls experimental imposition 7
 class cos-map-dscp-vo-1
  set mpls experimental imposition 5
 class cos-map-dscp-vs-1
  set mpls experimental imposition 7
 class cos-map-dscp-nm-1
  set mpls experimental imposition 6
 class class-default
  set mpls experimental imposition 0
policy-map cos-po2-vpnb-fe-0.0
 class cos-map-dscp-bu-1
  bandwidth 500
 class cos-map-dscp-nm-1
  bandwidth 280
  queue-limit 50
 class class-default
  bandwidth 300
  queue-limit 300

interface Loopback0
 ip address 10.200.0.1 255.255.255.255
interface FastEthernet0/0
 description TO ce_mun_vrfb-1
 bandwidth 6144
 ip vrf forwarding vpnb
 ip address 10.5.0.5 255.255.255.252
 max-reserved-bandwidth 100
 service-policy input cos-po-in-set-exp-dscp-1
 service-policy output cos-po2-vpnb-fe-0.0
 load-interval 30
 speed 100
 full-duplex
interface Serial0/0
 ip address 10.5.0.2 255.255.255.252
 service-policy input cos-po-in-set-dscp
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
```

```
 router-id 10.200.0.1
 log-adjacency-changes
 network 10.5.0.0 0.0.0.3 area 0
 network 10.200.0.1 0.0.0.0 area 0
router bgp 450
 no synchronization
 bgp router-id 10.200.0.1
 timers bgp 60 180
 bgp log-neighbor-changes
 neighbor 10.200.0.5 remote-as 450
 neighbor 10.200.0.5 description link-to_PE_BUD***
 neighbor 10.200.0.5 update-source Loopback0
 no auto-summary
 address-family vpnv4
 neighbor 10.200.0.5 activate
 neighbor 10.200.0.5 send-community extended
 exit-address-family
 address-family ipv4 vrf vpnb
 neighbor 10.5.0.6 remote-as 65000
 neighbor 10.5.0.6 description VPNB***
 neighbor 10.5.0.6 version 4
 neighbor 10.5.0.6 activate
 neighbor 10.5.0.6 as-override
 neighbor 10.5.0.6 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.5.0.4 mask 255.255.255.252
 exit-address-family
ip classless
ip http server
control-plane
end
```

PE_LON# sh running-config

```
version 12.3
…
hostname PE_LON
no aaa new-model
ip cef
ip vrf vpna
 rd 450:112
 route-target export 450:112
 route-target import 450:112
mpls label protocol ldp
mpls traffic-eng tunnels
interface Loopback0
 ip address 10.200.0.4 255.255.255.255
interface FastEthernet0/0
 description TO ce_man_vrfa-1
 ip vrf forwarding vpna
 ip address 10.7.0.5 255.255.255.252
 speed 100
```

```
 full-duplex
interface Serial0/0
 ip address 10.7.0.2 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.200.0.4
 log-adjacency-changes
 network 10.7.0.0 0.0.0.3 area 0
 network 10.200.0.4 0.0.0.0 area 0
router bgp 450
 no synchronization
 bgp router-id 10.200.0.4
 bgp log-neighbor-changes
 network 10.200.0.4 mask 255.255.255.255
 timers bgp 60 180
 neighbor 10.200.0.2 remote-as 450
 neighbor 10.200.0.2 description link-to_PE_NY1***
 neighbor 10.200.0.2 update-source Loopback0
 neighbor 10.200.0.3 remote-as 450
 neighbor 10.200.0.3 description link-to_PE_NY2***
 neighbor 10.200.0.3 update-source Loopback0
 neighbor 10.200.0.5 remote-as 450
 neighbor 10.200.0.5 description link-to_PE_BUD***
 neighbor 10.200.0.5 update-source Loopback0
 no auto-summary
 address-family vpnv4
 neighbor 10.200.0.2 activate
 neighbor 10.200.0.2 send-community extended
neighbor 10.200.0.3 activate
 neighbor 10.200.0.3 send-community extended
 neighbor 10.200.0.5 activate
 neighbor 10.200.0.5 send-community extended
 exit-address-family
 address-family ipv4 vrf vpna
 neighbor 10.7.0.6 remote-as 65000
 neighbor 10.7.0.6 description VPNA***
 neighbor 10.7.0.6 version 4
 neighbor 10.7.0.6 activate
 neighbor 10.7.0.6 as-override
 neighbor 10.7.0.6 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.7.0.4 mask 255.255.255.252
 exit-address-family
ip classless
ip http server
no ip http secure-server
control-plane
end
```

```
PE_NY1# show running-config
version 12.3
…
hostname PE_NY1
no aaa new-model
ip cef
ip vrf vpna
 rd 450:112
 route-target export 450:112
 route-target import 450:112
mpls label protocol ldp
mpls traffic-eng tunnels
interface Loopback0
 ip address 10.200.0.2 255.255.255.255
interface FastEthernet0/0
 description TO ce_ny_vrfa-1
 ip vrf forwarding vpna
 ip address 10.6.0.9 255.255.255.252
 speed 100
 full-duplex
interface Serial0/0
 ip address 10.6.0.2 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.200.0.2
 log-adjacency-changes
 network 10.6.0.0 0.0.0.3 area 0
 network 10.200.0.2 0.0.0.0 area 0
router bgp 450
 no synchronization
 bgp router-id 10.200.0.2
 bgp log-neighbor-changes
 network 10.200.0.2 mask 255.255.255.255
 timers bgp 60 180
 neighbor 10.200.0.3 remote-as 450
 neighbor 10.200.0.3 description link-to_PE_NY2***
 neighbor 10.200.0.3 update-source Loopback0
 neighbor 10.200.0.4 remote-as 450
 neighbor 10.200.0.4 description link-to_PE_LON***
 neighbor 10.200.0.4 update-source Loopback0
 neighbor 10.200.0.5 remote-as 450
 neighbor 10.200.0.5 description link-to_PE_BUD***
 neighbor 10.200.0.5 update-source Loopback0
 no auto-summary
 address-family vpnv4
 neighbor 10.200.0.3 activate
 neighbor 10.200.0.3 send-community extended
 neighbor 10.200.0.4 activate
```

```
 neighbor 10.200.0.4 send-community extended
 neighbor 10.200.0.5 activate
 neighbor 10.200.0.5 send-community extended
 exit-address-family
 address-family ipv4 vrf vpna
 neighbor 10.6.0.10 remote-as 65000
 neighbor 10.6.0.10 description VPNA***
 neighbor 10.6.0.10 version 4
 neighbor 10.6.0.10 activate
 neighbor 10.6.0.10 as-override
 neighbor 10.6.0.10 advertisement-interval 5
 neighbor 10.6.0.10 soft-reconfiguration inbound
 no auto-summary
 no synchronization
 network 10.6.0.8 mask 255.255.255.252
 exit-address-family
ip classless
control-plane
end
```

PE_NY2# show running-config

```
version 12.3
…
hostname PE_NY2
…
no aaa new-model
ip cef
ip vrf vpna
 rd 450:112
 route-target export 450:112
 route-target import 450:112
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
interface Loopback0
 ip address 10.200.0.3 255.255.255.255
interface FastEthernet0/0
 description TO ce_ny_vrfa-2
 ip vrf forwarding vpna
 ip address 10.6.0.13 255.255.255.252
 speed 100
 full-duplex
interface Serial0/0
 ip address 10.6.0.6 255.255.255.252
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.200.0.3
 log-adjacency-changes
```

```
 network 10.6.0.4 0.0.0.3 area 0
 network 10.200.0.3 0.0.0.0 area 0
router bgp 450
 no synchronization
 bgp router-id 10.200.0.3
 bgp log-neighbor-changes
 network 10.200.0.3 mask 255.255.255.255
 timers bgp 60 180
 neighbor 10.200.0.2 remote-as 450
 neighbor 10.200.0.2 description link-to_PE_NY1***
 neighbor 10.200.0.2 update-source Loopback0
 neighbor 10.200.0.4 remote-as 450
 neighbor 10.200.0.4 description link-to_PE_LON***
 neighbor 10.200.0.4 update-source Loopback0
 neighbor 10.200.0.5 remote-as 450
 neighbor 10.200.0.5 description link-to_PE_BUD***
 neighbor 10.200.0.5 update-source Loopback0
 no auto-summary
 address-family vpnv4
 neighbor 10.200.0.2 activate
 neighbor 10.200.0.2 send-community extended
 neighbor 10.200.0.4 activate
 neighbor 10.200.0.4 send-community extended
 neighbor 10.200.0.5 activate
 neighbor 10.200.0.5 send-community extended
 exit-address-family
 address-family ipv4 vrf vpna
 neighbor 10.6.0.14 remote-as 65000
 neighbor 10.6.0.14 description VPNA***
 neighbor 10.6.0.14 version 4
 neighbor 10.6.0.14 activate
 neighbor 10.6.0.14 as-override
 neighbor 10.6.0.14 advertisement-interval 5
 neighbor 10.6.0.14 soft-reconfiguration inbound
 no auto-summary
 no synchronization
 network 10.6.0.12 mask 255.255.255.252
 exit-address-family
ip classless
ip http server
control-plane
end
```

PE_BUD# show running-config

```
version 12.3
...
hostname PE_BUD
...
ip subnet-zero
ip cef
ip vrf vpna
 rd 450:112
```

```
 route-target export 450:112
 route-target import 450:112
ip vrf vpnb
 rd 450:111
 route-target export 450:111
 route-target import 450:111
mpls label protocol ldp
mpls traffic-eng tunnels
no ftp-server write-enable
class-map match-all cos-map-match-exp-2-st
 match mpls experimental topmost 2
class-map match-all cos-map-match-exp-6-nm
 match mpls experimental topmost 6
class-map match-all cos-map-match-exp-0-ec
 match mpls experimental topmost 0
class-map match-all cos-map-match-exp-7-bu
 match mpls experimental topmost 7
class-map match-all cos-map-match-exp-5-vo
 match mpls experimental topmost 5
class-map match-all cos-map-dscp-bu-1
 match ip dscp af21  af22  af23
class-map match-all cos-map-dscp-vo-1
 match ip dscp ef
class-map match-all cos-map-dscp-ec-1
 match ip dscp default
class-map match-all cos-map-dscp-st-1
 match ip dscp af41  af42  af43
class-map match-all cos-map-dscp-vs-1
 match ip dscp cs3  af31
class-map match-all cos-map-dscp-nm-1
 match ip dscp cs6
policy-map cos-po-in-set-dscp
 class cos-map-match-exp-5-vo
  set ip dscp ef
 class cos-map-match-exp-0-ec
  set ip dscp default
 class cos-map-match-exp-2-st
  set ip dscp af41
 class cos-map-match-exp-6-nm
  set ip dscp cs6
 class cos-map-match-exp-7-bu
  set ip dscp af21
policy-map cos-po-in-set-exp-dscp-1
 class cos-map-dscp-ec-1
  set mpls experimental imposition 0
 class cos-map-dscp-st-1
  set mpls experimental imposition 2
 class cos-map-dscp-bu-1
  set mpls experimental imposition 7
 class cos-map-dscp-vo-1
  set mpls experimental imposition 5
 class cos-map-dscp-vs-1
  set mpls experimental imposition 7
```

```
 class cos-map-dscp-nm-1
  set mpls experimental imposition 6
 class class-default
  set mpls experimental imposition 0
policy-map cos-po2-vpnb-fe-0.0.200
 class cos-map-dscp-bu-1
  bandwidth 500
 class cos-map-dscp-nm-1
  bandwidth 280
  queue-limit 50
 class class-default
  bandwidth 300
  queue-limit 300
interface Loopback0
 ip address 10.200.0.5 255.255.255.255
interface FastEthernet0/0
 description TO ce_deb_vrfab-1
 no ip address
 service-policy output cos-po2-vpnb-fe-0.0.200
 speed 100
 full-duplex
interface FastEthernet0/0.100
 description VRFA
 encapsulation dot1Q 100
 ip vrf forwarding vpna
 ip address 10.8.0.5 255.255.255.252
interface FastEthernet0/0.200
 description VRFB
 encapsulation dot1Q 200
 ip vrf forwarding vpnb
 ip address 10.8.0.9 255.255.255.252
 service-policy input cos-po-in-set-exp-dscp-1
interface Serial0/0
 ip address 10.8.0.2 255.255.255.252
 service-policy input cos-po-in-set-dscp
 mpls traffic-eng tunnels
 tag-switching ip
 clockrate 2000000
...
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.200.0.5
 log-adjacency-changes
 network 10.8.0.0 0.0.0.3 area 0
 network 10.200.0.5 0.0.0.0 area 0
router bgp 450
 no synchronization
 bgp router-id 10.200.0.5
 bgp log-neighbor-changes
 timers bgp 60 180
 neighbor 10.200.0.1 remote-as 450
 neighbor 10.200.0.1 description link-to_PE_FRA***
```

```
 neighbor 10.200.0.1 update-source Loopback0
 neighbor 10.200.0.2 remote-as 450
 neighbor 10.200.0.2 description link-to_PE_NY1***
 neighbor 10.200.0.2 update-source Loopback0
 neighbor 10.200.0.3 remote-as 450
 neighbor 10.200.0.3 description link-to_PE_NY2***
 neighbor 10.200.0.4 remote-as 450
 neighbor 10.200.0.4 description link-to_PE_LON***
 neighbor 10.200.0.4 update-source Loopback0
 no auto-summary
 address-family vpnv4
 neighbor 10.200.0.1 activate
 neighbor 10.200.0.1 send-community extended
 neighbor 10.200.0.2 activate
 neighbor 10.200.0.2 send-community extended
 neighbor 10.200.0.3 activate
 neighbor 10.200.0.3 send-community extended
 neighbor 10.200.0.4 activate
 neighbor 10.200.0.4 send-community extended
 exit-address-family
 address-family ipv4 vrf vpnb
 neighbor 10.8.0.10 remote-as 65000
 neighbor 10.8.0.10 description VPNB***
 neighbor 10.8.0.10 version 4
 neighbor 10.8.0.10 activate
 neighbor 10.8.0.10 as-override
 neighbor 10.8.0.10 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.8.0.8 mask 255.255.255.252
 exit-address-family
 address-family ipv4 vrf vpna
 neighbor 10.8.0.6 remote-as 65000
 neighbor 10.8.0.6 description VPNA***
 neighbor 10.8.0.6 version 4
 neighbor 10.8.0.6 activate
 neighbor 10.8.0.6 as-override
 neighbor 10.8.0.6 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.8.0.4 mask 255.255.255.252
 exit-address-family
ip classless
ip http server
...
```

```
ce-deb-vrfab-1# sh running-config
version 12.3
...
hostname ce_deb_vrfab-1
...
ip subnet-zero
```

```
ip cef
ip vrf vpna
 rd 450:1
ip vrf vpnb
 rd 450:2
class-map match-any cos-map-vpnb-fa-1.0-st
  match access-group name cos-map-vpnb-fa-1.0-st
class-map match-any cos-map-vpnb-fa-1.0-vo
  match access-group name cos-map-vpnb-fa-1.0-vo
class-map match-any cos-map-vpnb-fa-1.0-bu
  match access-group name cos-map-vpnb-fa-1.0-bu
class-map match-any cos-map-dscp-bu-1
  match ip dscp af21  af22  af23
class-map match-any cos-map-dscp-vo-1
  match ip dscp ef
class-map match-any cos-map-dscp-ec-1
  match ip dscp default  af11  af12  af13
class-map match-any cos-map-dscp-st-1
  match ip dscp af41  af42  af43
class-map match-any cos-map-dscp-vs-1
  match ip dscp cs3  af31
class-map match-any cos-map-dscp-nm-1
  match ip dscp cs6
policy-map cos-map-vpnb-fa-1.0
  class cos-map-vpnb-fa-1.0-vo
   set ip dscp ef
  class cos-map-vpnb-fa-1.0-st
   set ip dscp af41
  class cos-map-vpnb-fa-1.0-bu
   set ip dscp af21
  class class-default
   set ip dscp default
policy-map cos-child-vpnb-fa-0.1.200
  class cos-map-dscp-bu-1
   bandwidth 3530
   queue-limit 590
  class cos-map-dscp-nm-1
   bandwidth 277
   queue-limit 48
  class class-default
   bandwidth 1738
   queue-limit 290
policy-map cos-parent-vpnb-fa-0.1.200
  class class-default
   shape average 6144000
   service-policy cos-child-vpnb-fa-0.1.200
interface Loopback0
 ip address 10.250.0.1 255.255.255.255
interface FastEthernet0/0
 ip vrf forwarding vpna
 ip address 192.168.4.1 255.255.255.0
 duplex auto
 speed auto
```

```
...
interface FastEthernet0/1.100
 encapsulation dot1Q 100
 ip vrf forwarding vpna
 ip address 10.8.0.6 255.255.255.252
interface FastEthernet0/1.200
 encapsulation dot1Q 200
 ip vrf forwarding vpnb
 ip address 10.8.0.10 255.255.255.252
 service-policy output cos-parent-vpnb-fa-0.1.200
interface FastEthernet1/0
 ip vrf forwarding vpnb
 ip address 192.168.5.1 255.255.255.0
 duplex auto
 speed auto
 service-policy input cos-map-vpnb-fa-1.0
router bgp 65000
 bgp router-id 10.250.0.1
 bgp log-neighbor-changes
 timers bgp 60 180
 address-family ipv4
 no auto-summary
 no synchronization
 network 10.250.0.1 mask 255.255.255.255
 exit-address-family
 address-family ipv4 vrf vpnb
 neighbor 10.8.0.9 remote-as 450
 neighbor 10.8.0.9 description VPNB***
 neighbor 10.8.0.9 version 4
 neighbor 10.8.0.9 activate
 neighbor 10.8.0.9 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.8.0.8 mask 255.255.255.252
 network 192.168.5.0
 exit-address-family
 address-family ipv4 vrf vpna
 neighbor 10.8.0.5 remote-as 450
 neighbor 10.8.0.5 description VPNA***
 neighbor 10.8.0.5 version 4
 neighbor 10.8.0.5 activate
 neighbor 10.8.0.5 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.8.0.4 mask 255.255.255.252
 network 192.168.4.0
 exit-address-family
ip http server
ip classless
...
```

```
ce-man-vrfa-1# show running-config
version 12.3
...
hostname ce_man_vrfa-1
ip cef
interface Loopback0
 ip address 10.250.0.3 255.255.255.255
interface FastEthernet0/0
 ip address 192.168.2.1 255.255.255.0
 speed auto
 half-duplex
interface FastEthernet0/1
 ip address 10.7.0.6 255.255.255.252
 duplex auto
 speed auto
router bgp 65000
 bgp router-id 10.250.0.3
 bgp log-neighbor-changes
 timers bgp 60 180
 neighbor 10.7.0.5 remote-as 450
 address-family ipv4
 neighbor 10.7.0.5 activate
 neighbor 10.7.0.5 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.7.0.4 mask 255.255.255.252
 network 10.250.0.3 mask 255.255.255.255
 network 192.168.2.0
 exit-address-family
ip http server
ip classless
...
```

```
ce_mun_vrfb-1# show running-config
version 12.3
...
hostname ce_mun_vrfb-1
...
ip subnet-zero
ip cef
class-map match-any cos-map-vpnb-fa-0.0-st
  match access-group name cos-map-vpnb-fa-0.0-st
class-map match-any cos-map-vpnb-fa-0.0-vo
  match access-group name cos-map-vpnb-fa-0.0-vo
class-map match-any cos-map-vpnb-fa-0.0-bu
  match access-group name cos-map-vpnb-fa-0.0-bu
class-map match-any cos-map-dscp-bu-1
  match ip dscp af21  af22  af23
class-map match-any cos-map-dscp-vo-1
  match ip dscp ef
class-map match-any cos-map-dscp-ec-1
  match ip dscp default  af11  af12  af13
```

```
class-map match-any cos-map-dscp-st-1
  match ip dscp af41  af42  af43
class-map match-any cos-map-dscp-vs-1
  match ip dscp cs3  af31
class-map match-any cos-map-dscp-nm-1
  match ip dscp cs6
policy-map cos-map-vpnb-fa-0.0
  class cos-map-vpnb-fa-0.0-vo
   set ip dscp ef
  class cos-map-vpnb-fa-0.0-st
   set ip dscp af41
  class cos-map-vpnb-fa-0.0-bu
   set ip dscp af21
  class class-default
   set ip dscp default
policy-map cos-child-vpnb-fa-0.1
  class cos-map-dscp-bu-1
   bandwidth 3530
   queue-limit 590
  class cos-map-dscp-nm-1
   bandwidth 277
   queue-limit 48
  class class-default
   bandwidth 1738
   queue-limit 290
policy-map cos-parent-vpnb-fa-0.1
  class class-default
   shape average 6144000
   service-policy cos-child-vpnb-fa-0.1
interface Loopback0
 ip address 10.250.0.2 255.255.255.255
interface FastEthernet0/0
 ip address 192.168.3.1 255.255.255.0
 load-interval 30
 speed auto
 half-duplex
 service-policy input cos-map-vpnb-fa-0.0
interface FastEthernet0/1
 ip address 10.5.0.6 255.255.255.252
 duplex auto
 speed auto
 service-policy output cos-parent-vpnb-fa-0.1
router bgp 65000
 bgp router-id 10.250.0.2
 bgp log-neighbor-changes
 timers bgp 60 180
 neighbor 10.5.0.5 remote-as 450
 address-family ipv4
 neighbor 10.5.0.5 activate
 neighbor 10.5.0.5 advertisement-interval 5
 no auto-summary
 no synchronization
 network 10.250.0.2 mask 255.255.255.255
```

```
 network 192.168.3.0
 exit-address-family
ip http server
ip classless
ip access-list extended cos-map-vpnb-fa-0.0-bu
 remark *** Class Business ***
 permit icmp any any
 deny   ip any any
ip access-list extended cos-map-vpnb-fa-0.0-st
 remark *** Class Stream ***
 deny   ip any any
ip access-list extended cos-map-vpnb-fa-0.0-vo
 remark *** Class Voice ***
 deny   ip any any
end
```

ce-ny-vrfa-1# show running-config

```
version 12.3
…
hostname ce_ny_vrfa-1
no aaa new-model
ip cef
interface Loopback0
 ip address 10.250.0.4 255.255.255.255
interface FastEthernet0/0
 ip address 192.168.1.2 255.255.255.0
 duplex auto
 speed auto
 standby 1 ip 192.168.1.1
 standby 1 preempt delay minimum 60
 standby 1 track FastEthernet1/0 20
interface FastEthernet0/1
 ip address 10.6.0.17 255.255.255.252
 duplex auto
 speed auto
interface FastEthernet1/0
 ip address 10.6.0.10 255.255.255.252
 duplex auto
 speed auto
router bgp 65000
 bgp router-id 10.250.0.4
 bgp log-neighbor-changes
 timers bgp 60 180
 neighbor 10.6.0.9 remote-as 450
 neighbor 10.6.0.18 remote-as 65000
 address-family ipv4
 neighbor 10.6.0.9 activate
 neighbor 10.6.0.9 send-community both
 neighbor 10.6.0.9 advertisement-interval 5
 neighbor 10.6.0.9 route-map PRIMARY-PATH out
 neighbor 10.6.0.18 activate
 no auto-summary
```

```
 no synchronization
 network 10.6.0.8 mask 255.255.255.252
 network 10.6.0.16 mask 255.255.255.252
 network 10.250.0.4 mask 255.255.255.255
 network 192.168.1.0
 exit-address-family
ip http server
ip classless
route-map PRIMARY-PATH permit 10
 set metric 50
end
```

ce-ny-vrfa-2# show running-config

```
version 12.3
…
hostname ce_ny_vrfa-2
no aaa new-model
ip cef
interface Loopback0
 ip address 10.250.0.5 255.255.255.255
interface FastEthernet0/0
 ip address 192.168.1.3 255.255.255.0
 duplex auto
 speed auto
 standby 1 ip 192.168.1.1
 standby 1 priority 90
 standby 1 preempt
interface FastEthernet0/1
 ip address 10.6.0.18 255.255.255.252
 duplex auto
 speed auto
interface FastEthernet1/0
 ip address 10.6.0.14 255.255.255.252
 duplex auto
 speed auto
router bgp 65000
 bgp router-id 10.250.0.5
 bgp log-neighbor-changes
 timers bgp 60 180
 neighbor 10.6.0.13 remote-as 450
 neighbor 10.6.0.13 description linkto_PE_NY2
 neighbor 10.6.0.17 remote-as 65000
 neighbor 10.6.0.17 description linto_CE_NY1
 address-family ipv4
 neighbor 10.6.0.13 activate
 neighbor 10.6.0.13 send-community both
 neighbor 10.6.0.13 advertisement-interval 5
 neighbor 10.6.0.13 route-map BACKUP-PATH out
 neighbor 10.6.0.17 activate
 no auto-summary
 no synchronization
 network 10.6.0.12 mask 255.255.255.252
```

```
 network 10.6.0.16 mask 255.255.255.252
 network 10.250.0.5 mask 255.255.255.255
 network 192.168.1.0
 exit-address-family
ip http server
ip classless
route-map BACKUP-PATH permit 10
 set metric 100
end
```

Kapcsolatok tesztelése:

```
trace PC4 => PC7

VPCS 4 >trace 192.168.1.4
traceroute to 192.168.1.4, 64 hops max
 1   192.168.4.1   81.000 ms  63.000 ms  96.000 ms
 2   10.8.0.5   311.000 ms  208.000 ms  238.000 ms
 3   10.8.0.1   610.000 ms  622.000 ms  607.000 ms
 4   10.0.0.9   475.000 ms  413.000 ms  447.000 ms
 5   10.6.0.9   407.000 ms  410.000 ms  380.000 ms
 6   10.6.0.10   359.000 ms  320.000 ms  375.000 ms
 7   192.168.1.4   368.000 ms  639.000 ms  750.000 ms

trace PC4 => PC6

VPCS 4 >trace 192.168.2.2
traceroute to 192.168.2.2, 64 hops max
 1   192.168.4.1   65.000 ms  74.000 ms  142.000 ms
 2   10.8.0.5   382.000 ms  148.000 ms  247.000 ms
 3   10.8.0.1   472.000 ms  374.000 ms  522.000 ms
 4   10.0.0.5   552.000 ms  337.000 ms  320.000 ms
 5   10.7.0.5   614.000 ms  424.000 ms  279.000 ms
 6   10.7.0.6   550.000 ms  578.000 ms  588.000 ms
 7   192.168.2.2   796.000 ms  617.000 ms  650.000 ms

trace PC6 => PC4
VPCS 6 >trace 192.168.4.2
traceroute to 192.168.4.2, 64 hops max
 1   192.168.2.1   81.000 ms  31.000 ms  126.000 ms
 2   10.7.0.5   167.000 ms  215.000 ms  244.000 ms
 3   10.7.0.1   803.000 ms  520.000 ms  313.000 ms
 4   10.0.0.6   650.000 ms  336.000 ms  497.000 ms
 5   10.8.0.5   373.000 ms  251.000 ms  398.000 ms
 6   10.8.0.6   817.000 ms  682.000 ms  505.000 ms
 7   192.168.4.2   392.000 ms  536.000 ms  474.000 ms
```

```
trace PC6 => PC7

VPCS 6 >trace 192.168.1.4
traceroute to 192.168.1.4, 64 hops max
 1   192.168.2.1   63.000 ms  88.000 ms  94.000 ms
 2   10.7.0.5   363.000 ms  247.000 ms  146.000 ms
 3   10.7.0.1   559.000 ms  379.000 ms  388.000 ms
 4   10.0.0.1   369.000 ms  486.000 ms  489.000 ms
 5   10.6.0.9   507.000 ms  644.000 ms  357.000 ms
 6   10.6.0.10   546.000 ms  512.000 ms  499.000 ms
 7   192.168.1.4   477.000 ms  641.000 ms  542.000 ms

trace PC7 => PC6

VPCS 7 >trace 192.168.2.2
traceroute to 192.168.2.2, 64 hops max
 1   192.168.1.2   119.000 ms  52.000 ms  64.000 ms
 2   10.6.0.9   177.000 ms  156.000 ms  190.000 ms
 3   10.6.0.1   416.000 ms  782.000 ms  512.000 ms
 4   10.0.0.2   830.000 ms  390.000 ms  390.000 ms
 5   10.7.0.5   303.000 ms  243.000 ms  302.000 ms
 6   10.7.0.6   500.000 ms  314.000 ms  681.000 ms
 7   192.168.2.2   433.000 ms  463.000 ms  769.000 ms

trace PC7 => PC4

VPCS 7 >trace 192.168.2.2
traceroute to 192.168.2.2, 64 hops max
 1   192.168.1.2   119.000 ms  52.000 ms  64.000 ms
 2   10.6.0.9   177.000 ms  156.000 ms  190.000 ms
 3   10.6.0.1   416.000 ms  782.000 ms  512.000 ms
 4   10.0.0.2   830.000 ms  390.000 ms  390.000 ms
 5   10.7.0.5   303.000 ms  243.000 ms  302.000 ms
 6   10.7.0.6   500.000 ms  314.000 ms  681.000 ms
 7   192.168.2.2   433.000 ms  463.000 ms  769.000 ms
```